



ВОЙНА В НЕБЕ

1917

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ РЕДАКТОРА МИССИЙ “RISE OF FLIGHT“

ROF Editor user manual v1.0

СОДЕРЖАНИЕ

1. Введение.....	5
2. Описание логики редактора.....	6
<u>2.1. MCU.....</u>	<u>6</u>
2.1.1. MCU – Mission Control Unit.....	6
2.1.2. Объект (Entity).....	6
2.1.3. Команда.....	6
2.1.4. Триггер.....	6
2.1.5. Транслятор.....	6
<u>2.2. Связи.....</u>	<u>6</u>
2.2.1. Object Link.....	6
2.2.2. Target Link.....	7
<u>2.3. Сообщения и их виды.....</u>	<u>7</u>
3. Базовые правила создания миссий.....	8
4. Интерфейс и элементы управления редактора.....	9
<u>4.1. Строка инструментов редактора.....</u>	<u>9</u>
<u>4.2 Основное меню редактора.....</u>	<u>12</u>
4.2.1. Раздел File.....	12
4.2.2. Раздел View.....	13
4.2.3. Раздел Search and Select.....	14
4.2.4. Раздел Find missions objects.....	14
4.2.5. Раздел Find location objects.....	16
4.2.6. Раздел Draw.....	17
4.2.7. Раздел Tools.....	18
4.2.8. Раздел Location Database.....	20
<u>4.3. Камеры редактора и навигация в редакторе.....</u>	<u>20</u>
4.3.1. Режим 3D.....	20
4.3.2. Режим 2D.....	21

4.3.3. Камеры (Cameras)	21
4.4. Ниспадающее меню по правой кнопке мыши.....	22
4.5. Нижнее сервисное окно редактора.....	24
5. Миссия.....	25
5.1. Базовые свойства и настройки миссии.....	25
5.2. Файлы миссий, файлы локализации миссий, их формат.....	27
5.2.1. *.Mission.....	27
5.2.2. *.msnbin.....	29
5.2.3. [mission_name].eng.....	29
5.2.4. [mission_name].list.....	29
5.2.5. Работа с локализационными файлами миссии.....	29
5.2.6. ID локализации.....	30
6. Работа с объектами.....	31
6.1. Выделение объектов.....	31
6.2. Перемещение и ориентация объектов.....	32
6.3. Копирование, вставка, удаление.....	33
6.4. Группирование объектов.....	33
6.5. Объединение движущихся объектов в звено или группу.....	35
6.6. Сохранение в файл, импорт и вставка сохраненных групп.....	35
6.6.1 Сохранение в файл.....	35
6.6.2 Импорт объектов в миссию.....	36
7. Объекты редактора.....	37
7.1. Раздел Objects Library редактора.....	37
7. 2. Basic Properties для объектов редактора.....	38
7.2.1 Planes/Vehicles/Artillery/Aerostats.....	38
7.2.2. Trains.....	39
7.2.3. Firing Point.....	40
7.2.4. Buildings/Blocks/Airfields/Battlefield/Bridges.....	41
7.2.5. Helpers.....	43

7.2.6. Locations.....	46
<u>7.3. Advanced Properties для объектов редактора.....</u>	<u>54</u>
7.3.1. Planes/Vehicles/Artillery/Aerostats.....	54
<u>7.4. Mission Command Units (MCUs) редактора.....</u>	<u>58</u>
7.4.1. Basic Properties для MCUs.....	58
<u>7.5. Advanced Properties для MCUs.....</u>	<u>59</u>
<u>7.6. MCU Command.....</u>	<u>59</u>
7.6.1. Command:Attack	59
7.6.2. Command:AttackArea	62
7.6.3. Command:Behavior.....	65
7.6.4. Command:Cover.....	68
7.6.5. Command:Damage.....	71
7.6.6. Command:Effect.....	73
7.6.7. Command:Force Complete.....	75
7.6.8. Command:Formation.....	77
7.6.9. Command:Land	80
7.6.10. Command:Move.....	82
7.6.11. Command:PlayAnimation.....	82
7.6.12. Command:PlaySound	84
7.6.13. Command:Take off.....	86
<u>7.7. MCU Translator.....</u>	<u>88</u>
7.7.1. Translator:Animation Operator.....	88
7.7.2. Translator:Camera Operator.....	91
7.7.3. Translator:Icon	95
7.7.4. Translator:Media	102
7.7.5. Translator:Mission Objective.....	106
7.7.6. Translator:Mission begin.....	108
7.7.7. Translator:Mission End.....	110
7.7.8. Translator:Subtitle.....	111

<u>7.8. MCU Trigger</u>	115
7.8.1. Trigger:Activate.....	115
7.8.2. Trigger:Camera point.....	117
7.8.3. Trigger:Check zone.....	119
7.8.4. Trigger:Counter.....	124
7.8.5. Trigger:Deactivate.....	126
7.8.6. Trigger:Proximity.....	128
7.8.7. Trigger:Spawner.....	132
7.8.8. Trigger:Timer.....	134
7.8.9. Схема Random Timer.....	135
7.8.10. Trigger:Waypoint.....	137
8. Создание кооперативной миссии	141
9. Часто используемые блоки миссий, примеры и приемы построения	142
<u>9.1. Взлет, полет по маршруту и посадка</u>	142
<u>9.2.Построение (Formation) и перестроение в поле</u>	143
<u>9.3. Команда Attack на маршруте</u>	144
<u>9.4. Старт в воздухе</u>	144
<u>9.5. Колонна автомобилей на дороге</u>	145
<u>9.6.Движение поезда</u>	146
<u>9.7. Создание анимационной сцены на примере хелпера (фигура в воздухе)</u>	147
9.7.1. Объект анимации виден сразу после запуска миссии.....	147
9.7.2. Объект анимации виден не сразу после запуска миссии.....	149
<u>9.8. Создание анимационной сцены на примере сцены из тренировочной кампании</u>	149
<u>9.9. Экран (или заставка) при старте миссии</u>	151
<u>9.10. Базовая миссия – все города, аэродромы и мосты на карте</u>	152
<u>9.11. Файлы, используемые в Media и Animation Operator</u>	152
9.11.1. Media файлы в сборке игры.....	152
9.11.2. *.chr файлы для Animation Operator.....	152
10. Общая информация	154

1. Введение.

Редактор ROF это:

- универсальная триггерная система выполнения и распределения команд;
- возможность активации и деактивации по определенным событиям объектов в миссии, что значительно увеличивает производительность при наличии большого количества объектов в миссии;
- возможность обратной совместимости однопользовательских и мультиплеерных миссий (миссии работают на одной и той же логике и создаются при помощи универсальных инструментов редактора);
- возможность установки приоритетов выполнения команд искусственным интеллектом, что в свою очередь определяет возможность нелинейности сюжета миссии. Эта особенность дает возможность искусственному интеллекту действовать в зависимости от сложившейся вокруг него обстановки, что в свою очередь создает иллюзию более естественных (человечных) действий со стороны искусственного интеллекта. Особенно, в случае мультиплеерной миссии. Например, в случае отсутствия угрозы со стороны игрока или других объектов миссии, искусственный интеллект продолжит выполнение задания. Если такая угроза есть – действия AI можно либо жестко определить при помощи дизайна миссии, либо AI будет действовать самостоятельно, и после исчезновения угрозы – продолжить выполнение задания. Все это очень положительно влияет на игровой процесс и атмосферность игрового процесса;
- визуализация игрового процесса при помощи интерфейсных хелперов (иконки объектов, иконки маршрута и пр.). Разнообразие типов иконок, возможность визуализации миссии в брифинге и дебрифинге карты. Возможность добавления описания действий в миссии во всплывающих подсказках на карте миссии;
- встроенная в редактор система камер, которая позволяет создавать сценарные ролики на движке игры. Собственно, сами камеры являются частью миссии и могут действовать как часть игрового процесса без его прерывания;
- возможность вставки текстовых, графических и видео вставок в миссию, как часть игрового процесса. Все эти возможности позволяют упростить игровой процесс, а так же создать антураж и использовать сюжетную линию в миссии;
- широкие возможности собственно редактирования миссий. Возможность группового редактирования свойств объектов, перемещение, вращение, копирование (как отдельных объектов, так и всей миссии). Возможность группирования объектов в отдельные блоки миссии с последующим их использованием как конструктора (например, стандартные группы самолетов или автомашин, стандартные виды наземных построек). Возможность импорта миссий – это дает возможность объединения усилий нескольких дизайнеров миссий в одну, законченную миссию и ускорение производства сложных миссий (один делает художественную вставку-сюжет, второй расставляет антураж миссии, третий – создает функциональную часть миссии и т.д.).

Все вышеперечисленные свойства редактора позволят вам реализовать широчайшие творческие задумки в плане миссий, кампаний и игрового процесса.

2. Описание логики редактора.

Для создания динамически изменяющихся миссий Полный Редактор Миссий (далее ПРМ) использует скриптовый язык, основанный на взаимодействии объектов в среде симулятора и некоторых логических элементов (MCU) в скрипте миссии – команд, триггеров и трансляторов.

2.1. MCU

2.1.1. MCU – Mission Control Unit – это скрипты, которые задают различные действия/свойства объектам миссии и самой миссии в зависимости от типа MCU.

2.1.2. Объект (Entity)– некоторая игровая сущность, которая(ые) в миссии может(гут) взаимодействовать между собой и игроком – самолеты, постройки, техника. Объектам может быть присвоена какая-то модель поведения или последовательность действий в миссии. Для того, чтобы объект мог так взаимодействовать и быть редактируемым, его обязательно нужно «оживить» с помощью кнопки Create Linked Entity, которая находится в меню объекта Basic Properties (см.п.7.2.)

Важно знать, что с точки зрения правильного понимания работы скриптового движка, объект становится собственно объектом только в запущенной в игре миссии, в редакторе же мы устанавливаем транслятор, через который, впоследствии, скрипт задает модель поведения объекта в движке игры.

2.1.3. Команда – это MCU, задающая объекту его линию поведения (команда типа «атакуй врагов в окрестности») или последовательность совершаемых им действий (команда типа «двигайся в точку»). Другими словами – команда «говорит» объекту, что ему нужно делать и в каком порядке. В списке MCUs команды имеют префикс «Command:».

2.1.4. Триггер – это MCU, который активирует команду или транслятор при выполнении некоторых условий. Триггер может быть как зависимым от объекта (например, активировать команду/событие при вхождении объекта в некоторую зону), так и нет (триггер активирует команду/событие по истечении какого-то времени от предыдущей команды/события). В списке MCUs команды имеют префикс «Trigger:».

2.1.5. Транслятор – сущность скрипта, ответственная за связь скрипта и объекта. То есть транслятор, это скрипт, который средствами редактора привязывается к объекту и действует на этот объект в миссии по заданным условиям.

2.2. Связи

Для написания корректного скрипта миссии необходимо понимать логику связей логических элементов с объектами и между собой. Существуют 2 типа связей – объектные и целевые.

2.2.1. Object Link (горячие клавиши Shift+O) Объектная связь (далее ОС – в редакторе обозначается линией со стрелкой зеленого цвета) - устанавливается между командой, триггером, транслятором и объектом (от MCU на объект т.е. нужно выделить данный MCU => Shift+O=>объект), на который она влияет. Объектная связь (через транслятор, привязанный к объекту), говорит, какой объект должен выполнить данную MCU.

Например, объектная связь от команды «идти в точку» к объекту «самолет» говорит, что в миссии именно этот самолет должен проследовать в указанную точку. В случае, если связь

устанавливается от триггера, она может означать так же, что объект является участником условий для срабатывания данного триггера.

2.2.2. Target Link (горячие клавиши Shift+T) Целевая связь (далее ЦС - в редакторе обозначается линией со стрелкой красного цвета) – устанавливается между MCU, задавая последовательность их выполнения.

То есть, нужно выделить MCU, из которого исходит целевая связь и указать на тот MCU, который будет следующим в логической цепочке, проще говоря – «дернуть» одним MCU другой MCU

Важно помнить, что для правильной последовательности выполнения скрипта связь должна быть проставлена от предыдущего MCU к следующему, и никак иначе. На один и тот же MCU может быть завязано множество других MCU, в различных последовательностях.

Примечание: не все MCU могут выдавать целевые связи, для этого смотрите пункт 7.5 с подробным описанием свойств MCU.

2.3. Сообщения и их виды.

Message – (в редакторе обозначается линией со стрелкой синего цвета) – это еще один вид связи, задающей последовательность срабатывания MCU.

Однако, в отличие от целевой связи, причиной срабатывания связи является не срабатывание «родительского» MCU, а изменение определенного состояния объекта (через транслятор объекта – его сущность) в результате поступления определенной команды или изменения свойств объекта (например - его смерть).

Другими словами через сообщение объект говорит скрипту «я сделал это». Интерфейсно сообщение устанавливается следующим образом – выделяем объект (обязательно с entity), нажмем правой кнопкой мыши (ПКМ) на его значке, в контекстном меню выбираем пункт Select Object Menu, после чего выбираем необходимое нам сообщение (сообщения имеют постфикс Message).

Существует несколько типов Message:

- Add On Land Message – сообщение о факте выполнения команды посадки;
- Add On Take off Message – сообщение о факте выполнения команды взлета;
- Add On Move Message – сообщение о факте движения;
- Add On Killed Message – сообщение о факте «смерти» объекта;
- Add On Damage Message – сообщение о фактически полученных повреждениях;
- Add On Spawned Message – сообщение о факте «рождения» объекта;
- Add On Target Attacked Message – сообщение о факте выполнения команды атаки цели;
- Add On Area Attacked Message – сообщение о факте выполнения команды атаки зоны;

При назначении Message использует двойную связь – source и target. В большинстве случаев, достаточно использовать target (чтобы пропустить source – нажмем Esc в режиме назначения сообщения – курсор поменяет свой тип, и укажите target), но иногда требуется использовать и

source – это когда вам нужно использовать сообщение по факту выполнения определенной команды.

Например, в миссии присутствует две команды на взлет для одного самолета, вы указываете source – первую команду взлета и target - первую маршрутную точку полета самолета, соответственно, когда самолету нужно произвести второй взлет, то как source вам нужно указать уже другую команду. То есть в данном примере, объект выдаст сообщение после выполнения определенной цепочки действий. Таким образом, можно поступать и с другими выполненными командами.

3. Базовые правила создания миссий.

При создании миссий важно помнить следующие базовые правила:

- необходим и достаточен как минимум один транслятор Mission Begin, но их количество не ограничено, например, вы можете импортировать самостоятельные функциональные блоки миссии, при этом сам блок будет иметь свой Mission Begin;
- каждая команда должна быть привязана к объекту, который выполняет эту команду, т.е. команда должна «знать» для кого она предназначена;
- команда или триггер исполняются (срабатывают) только при условии их запуска целевой связью или сообщением от другого объекта;
- необходимо избегать одновременной подачи команд на один объект и детерминировать последовательность поступления команд с помощью триггеров Timer (1-2 секунды);
- деактивированный объект не получает команд, храня последние команды, полученные перед деактивацией;
- при активации объекта нужна задержка перед подачей ему новых команд в 1-2 секунды, т.к. процесс активации происходит более чем за один шаг моделирования.
- при использовании MCU Spawner надо помнить что нельзя «рождать» игрока (самолет управляемый игроком), нельзя «рождать» группы;
- не ставить более 20 объектов одинакового типа (все, что находятся в списках Bridges, Buildings, Airfields, Battlefield окна Library) «в кадре» (примерно в радиусе 5 км.);
- важно учитывать приоритет выданной объекту команды;
- не ставить более 35 объектов разного типа «в кадре»;
- AI будет атаковать объект только противоположной (вражеской) коалиции, важно помянуть об этом при создании миссии;
- важно соблюдать ограничения для MCU редактора, которые написаны в разделе X данного руководства.

4. Интерфейс и элементы управления редактора.

4.1. Строка инструментов редактора.



1) – «вырезать» объект – удалить объект с карты, и поместить его в буфер обмена.

2) – копировать объект – поместить копию объекта в буфер обмена.

3) – Measure Mode – линейка, позволяет измерить расстояние между точками на карте, возможно задание любой ломаной линии по точкам, с динамическим измерением дистанции и отображением дистанции между точками отрезков, где:

D – длина последнего отрезка,

L – общая длина измеренного расстояния,

Hdg – курсовой угол (направление) измерения.

Горячие клавиши для использования линейки – Shift+M

Убрать последний отрезок и далее, все предыдущие отрезки – последовательные нажатия клавиши `backspace`. Выход из режима – однократное нажатие клавиши `Esc`.

4) – Toggle Camera – переключает вид камеры в редакторе с изометрического 3D на ортогональный 2D, и наоборот. Горячая клавиша для переключения из 2D в 3D – F9.

5) – Focus on Selection - фокусирует камеру на выделенном объекте. Объект размещается в центре монитора.

6) – Set on Ground – устанавливает объект (или выделенную группу объектов) на поверхность.

7) – Zoom to Cursor – если кнопка включена то приближение камеры идет в точку, где находится курсор, если отключена – по центру экрана.

8) – Draw Grid – включает/отключает сетку карты.

9) – Draw Icons – включает/отключает прорисовку иконок установленных на карту объектов и MCU редактора.

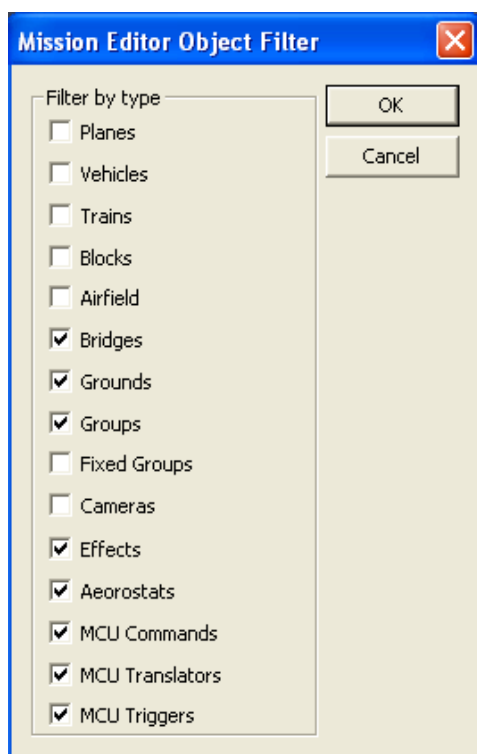
10) – Draw Icon Text – включает/отключает вывод текста с названием и ID сущностей установленных на карту.

11) - Draw Icons in Group – включает/выключает отображение иконок сущностей в группах (по умолчанию отключено).

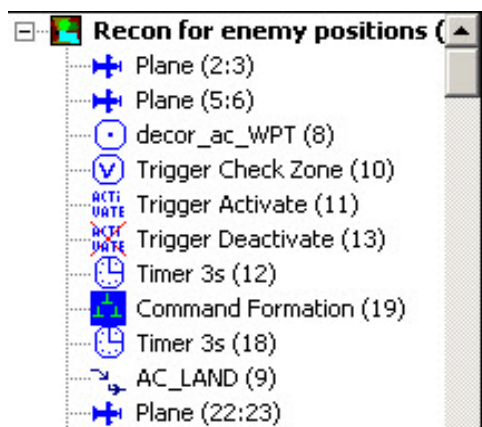
12) – включает/выключает компас, где синяя линия – направление на север, причем в 3D режиме направление дублируется цифрами – куда смотрит камера (курсовой угол), зеленая линия – направление ветра у поверхности земли.

13) – Object Filter – открывает окно Mission Editor Object Filter, которое позволяет отключить прорисовку иконок на карте выбранных категорий объектов и MCU. Так же, этот фильтр скрывает

иконки объектов, чтобы они не мешали при редактировании, если они не нужны в данный момент (грубо говоря – объект нельзя выделить), например можно скрыть иконки мостов и блоков (или групп наземных построек) т.к. при построении миссии они будут занимать рабочее пространство редактора и просто мешать:

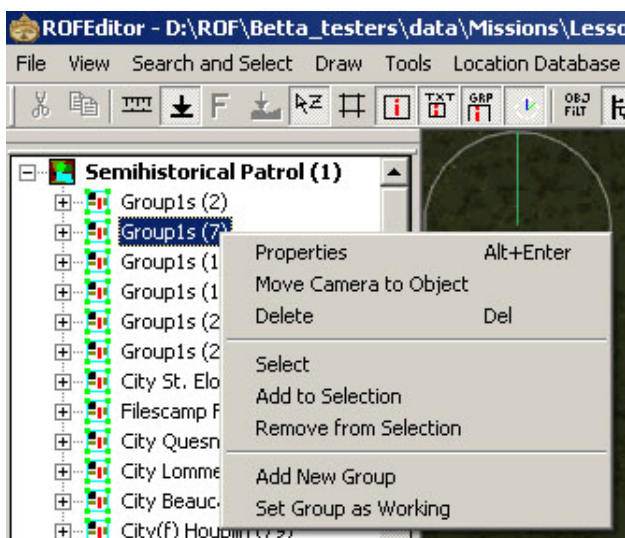


14) – Mission Tree – включает/выключает окно Mission Tree, представляющее собой список всех установленных на карту объектов и MCU.



В скобках первой цифрой показан ID объекта, второй ID entity объекта, если это MCU, то это ID MCU.

Также, помимо отображения списка сущностей, данное окно позволяет быстро переместиться к нужной сущности двойным нажатием по ее имени в списке. Если в этом меню выделить объект и нажать ПКМ, то появится всплывающее меню:



Properties – свойства выбранного объекта (MCU);

Move Camera to Object – перемещение именно к выбранному объекту (MCU);

Delete – удаление выбранного объекта (MCU);

Select – выделение выбранного объекта (MCU);

Add to Selection – добавление данного объекта (MCU) в список уже выделенных объектов (MCU);

Remove from Selection – удаление данного объекта (MCU) из списка ранее выделенных объектов (MCU);

Add New Group – не используется;

Set Group as Working - возможность редактировать группу объектов (см. раздел 6 «Работа с объектами»).

15) – Objects Library – открывает/закрывает окно Objects Library.

16) – Property Dialog – открывает/закрывает окно Basic Properties.

17) – MCU OBJ – включает/выключает отображение объектных связей.

18) – MCU TAR – включает/выключает отображение целевых связей.

19) – MCU MES – включает/отключает отображение связей типа «сообщение».

20) – MCU SEL – включает/выключает отображение связей у выбранных сущностей.

21) – MCU UNSE – включает/выключает отображение связей у невыделенных сущностей.

22) – базовая автопроверка графа миссии.

23) – включает/выключает автопроверку при каждой save/load миссий

24) – Unit Tests

25) – включает маркировку (подсветку дорог) красной маской

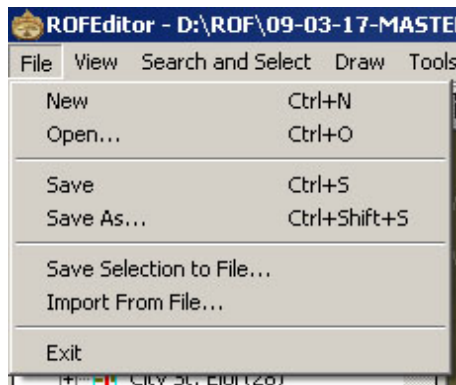
26) – не используется

27) – не используется.

28) – информация о редакторе.

4.2 Основное меню редактора

4.2.1. Раздел File:



- New – создать новый файл миссии;

- Open... – открыть файл миссии;

- Save – сохранить файл миссии;

- Save As... – сохранить файл миссии под новым именем...;

- Save Selection to File... – сохранить выбранные объекты (сущности) в отдельном файле для последующей вставки или импорта в миссии.

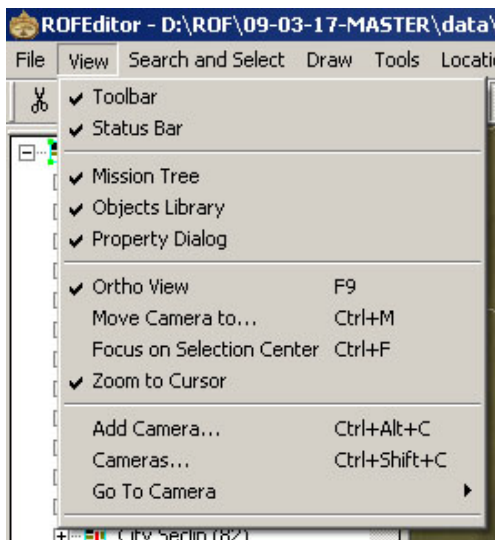
При этом объекты сохраняются в группу с расширением *.Group. Как работать с группами – см. п.6.

По умолчанию *.Group файлы сохраняются в директорию \Data\Missions;

- Import From File... – импортировать *.Group файлы в миссию;

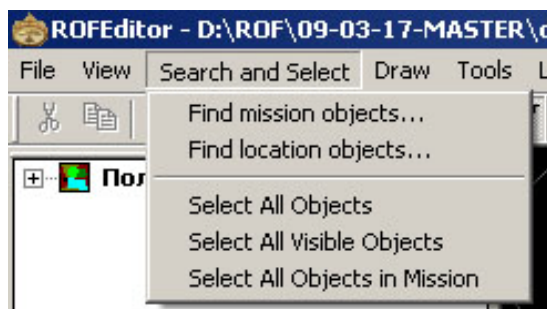
- Exit – выход из редактора миссий.

4.2.2. Раздел View:



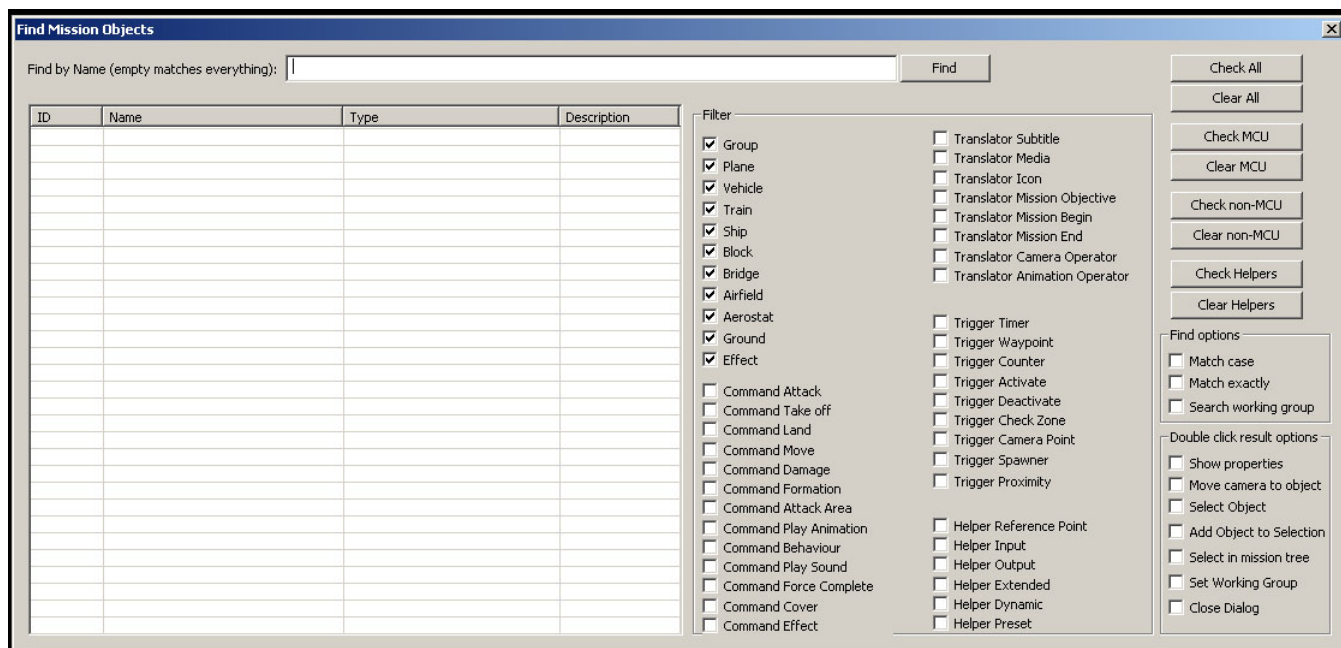
- Toolbar – отображение линейки инструментов редактора;
- Status Bar – отображение панели сервисной информации рабочей области редактора (см.п. 4.5);
- Mission Tree – отображение окна «дерево» миссии;
- Objects Library – отображение окна сущностей редактора (объектов, команд, трансляторов и т.д.);
- Property Dialog – отображение окна свойств миссии и объектов;
- Ortho View – плоский режим просмотра 2D;
- Move Camera to... – перемещение камеры в заданные координаты;
- Focus on Selection Center – центрирование взгляда (фокусировка) на выделенном объекте или MCU;
- Zoom to Cursor – если кнопка включена то приближение камеры идет в точку, где находится курсор, если отключена – по центру экрана.
- Add Camera... – добавить камеру;
- Cameras... – редактирование настроек камер в списке предустановленных камер;
- Go to Camera – выбор и перемещение в камеру, выставленную с помощью функции Add Camera (см.п.4.3).

4.2.3. Раздел Search and Select:



- Find missions objects... – поиск и выбор объектов в миссии;
- Find location objects... – поиск и выбор локаций;
- Select All Objects – выбор всех сущностей миссии и объектов игрового мира, а так же выбор всех объектов в рабочей группе;
- Select All Visible Objects – выбор всех видимых объектов в миссии, прорисованных с помощью Mission Editor Object Filter (см. Выше), а так же выбор всех объектов в рабочей группе;
- Select All Objects in Mission – выбор всех объектов миссии.

4.2.4. Раздел Find missions objects:



- поле Find by Name (empty matches everything) – поле поиска по имени сущности или объекта;
- кнопка Find – найти сущность или объект миссии по заданному имени;
- окно ID, Name, Type, Description - окно найденных сущностей или объектов, отображающее идентификатор в миссии, имя, тип и описание, соответственно;

Поле Filter – настройка фильтра поиска и выбора по типу объекта.

Кнопка Check All – выбор всех чекбоксов фильтра;

Кнопка Clear All – очистить все чекбоксы фильтра;

Кнопка Check MCU – выбор всех чекбоксов фильтра MCU (команды, трансляторы, триггеры);

Кнопка Clear MCU – очистить все чекбоксы фильтра MCU;

Кнопка Check non-MCU – выбор всех чекбоксов фильтра для объектов;

Кнопка Clear non-MCU – очистить все чекбоксы фильтра для объектов;

Кнопка Check Helpers – выбор всех чекбоксов фильтра «помощников»;

Кнопка Clear Helpers – очистить все чекбоксы фильтра «помощников».

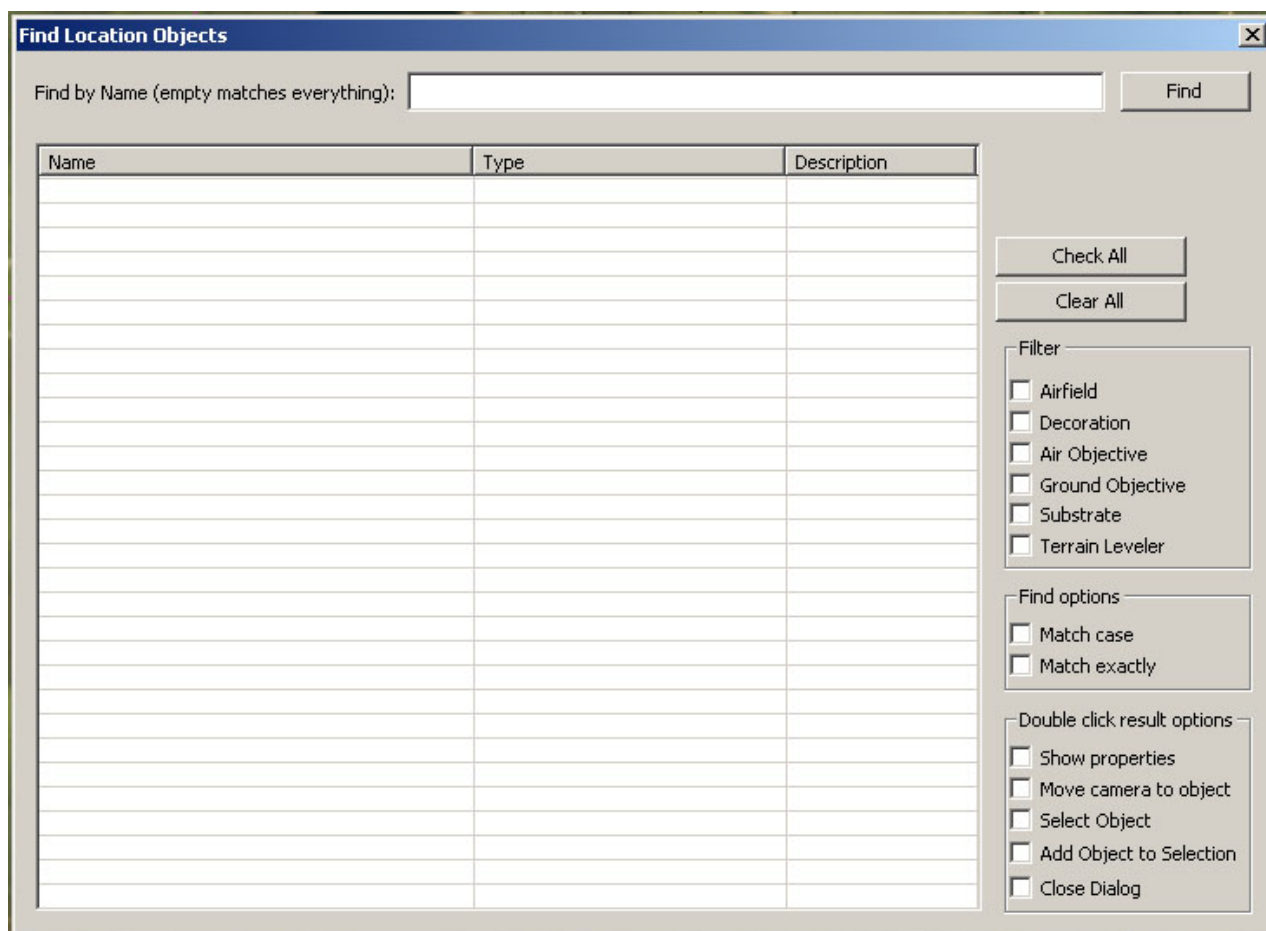
Поле Find options – настройки фильтра поиска:

- чекбокс Match case – поиск с учетом регистра;
- чекбокс Match exactly – поиск по полному совпадению имени;
- чекбокс Search working group – поиск только в рабочей группе;

Поле Double click result options – настройка действий по двойному нажатию на найденный объект в списке найденных объектов:

- чекбокс Show properties – отображение окна Properties найденного объекта по двойному нажатию;
- чекбокс Move camera to object – перемещение камеры к найденному объекту по двойному нажатию;
- чекбокс Select Object – выбор (селектирование) найденного объекта по двойному нажатию;
- чекбокс Add Object to Selection – добавление объекта в список выбранных по двойному нажатию;
- чекбокс Select in mission tree – выбор (селектирование) найденного объекта в окне «дерева» миссии;
- чекбокс Set Working Group – установление режима редактирования для найденного объекта по двойному нажатию (если он в группе);
- чекбокс Close Dialog – закрывает окно Find Mission Object после двойного нажатия по имени (в списке) найденного объекта.

4.2.5. Раздел Find location objects...



Этот режим позволяет найти и выделить локацию установленную на карте, которая появится на карте после загрузки БД локаций - Default.ldf, Substrate.ldf и TerrainLeveler.ldf. Прежде чем приступить к поиску, необходимо подгрузить эти файлы в редактор.

Поле Find by Name (empty matches everything) – поле ввода полного (не полного) имени сущности.

Кнопка Find – найти локацию по установленному имени.

Окно Name, Type, Description - окно найденной локации, отображающее ее имя, тип и описание, соответственно.

Кнопка Check All – выбор всех чекбоксов фильтра;

Кнопка Clear All – очистить все чекбоксы фильтра;

Поле Filter – фильтр поиска локаций в редакторе.

Чекбокс Airfield – поиск локаций аэродромов;

Чекбокс Decoration – поиск локаций декораций игрового мира;

Чекбокс Air Objective – поиск локаций типа Air Objective;

Чекбокс Ground Objective – поиск локаций типа Ground Objective;

Чекбокс Substrate – поиск локаций типа Substrate;

Чекбокс Terrain Leveler – поиск зон (локаций) выравнивания местности;

Поле Find options – дополнительный фильтр поиска и выбора локаций в редакторе.

Чекбокс Match case – поиск с учетом регистра;

Чекбокс Match exactly – поиск по полному имени;

Поле Double click result options – настройка действия по даблклику в окне ID, Name, Type, Description.

Чекбокс Show properties – отображение окна Properties найденной локации;

Чекбокс Move camera to object – перемещение камеры к найденной локации;

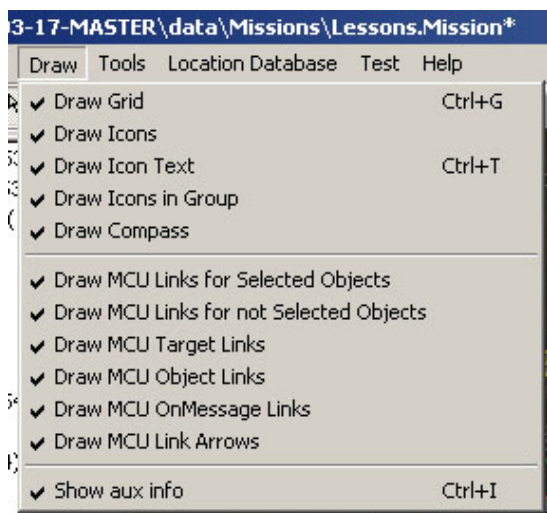
Чекбокс Select Object – выбор (селектирование) найденной локации;

Чекбокс Add Object to Selection – добавление локации в список выбранных;

Чекбокс Close Dialog – закрывает окно Find Location Object...

Данный режим поиска непосредственно пересекается с такими объектами редактора как Helpers и Locations, подробнее о них читай в разделе 7.2.5 и 7.2.6.

4.2.6. Раздел Draw – меню прорисовки в главном окне редактора различных объектов и MCU в рабочем поле редактора.



Draw Grid – прорисовка сетки координат в 2D режиме;

Draw Icons – прорисовка иконок объектов и MCU редактора;

Draw Icon Text – отображение имен иконок объектов и MCU редактора;

Draw Icons in Group – прорисовка иконок групп;

Draw Compass – прорисовка компаса игрового мира;

Draw MCU Links for Selected Objects – прорисовка связей для выделенных объектов;

Draw MCU Links for not Selected Objects – прорисовка связей для не выделенных объектов;

Draw MCU Target Links – прорисовка целевых связей;

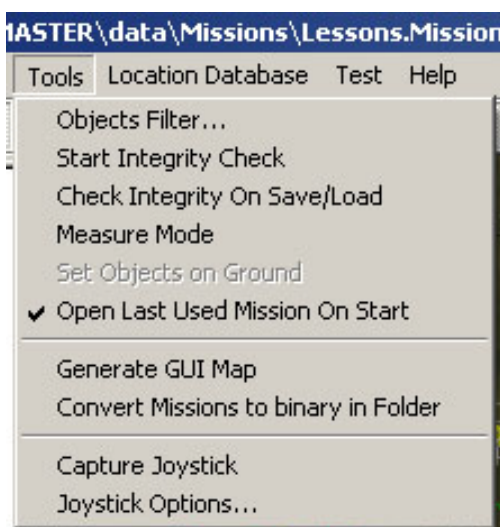
Draw MCU Object Links – прорисовка объектных связей;

Draw MCU OnMessage Links – прорисовка связей типа сообщения;

Draw MCU Link Arrows – прорисовка стрелок логического направления связей;

Show aux info – прорисовка информации о частоте обновления кадров и количестве FPS.

4.2.7. Раздел Tools – меню инструментов редактора.



Object Filter – открывает окно Mission Editor Object Filter, которое позволяет отключить прорисовку на карте выбранных категорий объектов (см. выше).

Start Integrity Check – запускает программу проверки назначенных связей между сущностями миссии и выводит результаты в главном окне редактора;

Check Integrity On Save/Load – запускает программу проверки назначенных связей между сущностями миссии и выводит результаты в главном окне редактора при каждом сохранении или загрузке миссии;

Measure Mode - линейка, позволяет измерить расстояние между точками на карте, возможно задание любой кривой по точкам, с динамическим измерением дистанции и отображением дистанции между отрезками. Убрать последний отрезок и далее остальные отрезки – последовательные нажатия клавиши backspace.

Выход из режима – однократное нажатие клавиши Esc;

Open Last Used Mission On Start – при запуске редактора загружается последняя открытая в редакторе миссия;

Generate GUI Map – для внутреннего использования разработчика;

Convert Missions to binary in Folder – конвертация файла миссии в бинарный формат – при серьезных изменениях в версии редактора миссий часто меняется и бинарный формат миссий (см.п. 5.2.2.). Данная опция позволяет автоматически пересохранять старые миссии в новом формате.

Для этого выберите директорию с миссиями, которые вы хотели бы пересохранить, и редактор автоматически пересохранит все миссии, находящиеся в данной директории и поддиректории;

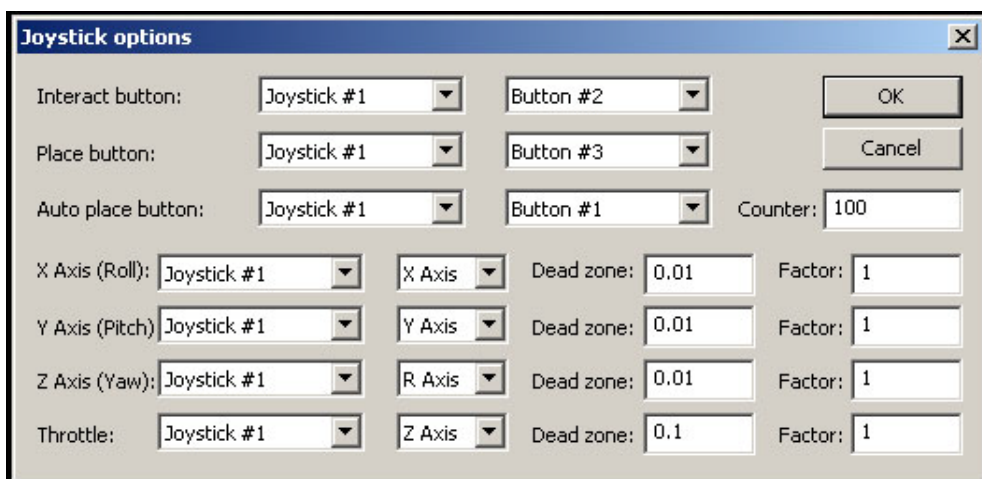
Capture Joystick – управление полетом камеры в редакторе при помощи джойстика



При помощи этого режима осуществляется «перехват» джойстика редактором миссий; два режима – простой полет в мире игры и полет транслятора Camera Operator. Во втором случае, если поставить на карту транслятор Camera Operator – ПКМ – Selected Object Menu – Look from this Object вы сможете в 3D режиме пролететь по желаемой траектории, причем редактор позволяет расставить все промежуточные контрольные точки камеры (Trigger Came Point) в автоматическом или ручном режиме.

Так же, при помощи данного режима, можно расставлять Waypoints для самолетов и техники, для этого вам нужно выделить объекто, для которого вы будете выставлять точки (или последний Waypoint данного объекта).

Joystick Options – панель настройки управления джойстиком для полетов в простом или Camera Operator режимах.



Interaction button – выбор кнопки джойстика для начала движения;

Place button – кнопка ручной расстановки промежуточных контрольных точек полета камеры;

Auto place button - кнопка автоматической расстановки промежуточных контрольных точек полета камеры, расставляет точки во время «полета» автоматически;

Counter – время в миллисекундах между промежуточными контрольными точками (только для режима Auto place button);

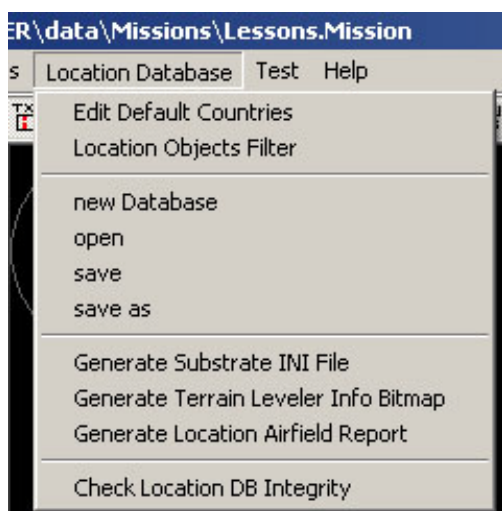
Factor – настройки чувствительности джойстика;

X, Y, Z Axis – настройки осей джойстика;

Throttle – скорость перемещения в полете;

Dead zone – «мертвые» зоны в отклике джойстика.

4.2.8. Раздел Location Database – меню для работы с локациями для генератора миссий.



Edit Default Countries – редактирование состава коалиций Opposite и Order (см.п.5.1.);

Locations Object Filter – фильтр локаций;

New Database – создать новую базу данных;

Open - открыть;

Save as – сохранить как;

Generate Substrate INI file – сгенерировать файл текстур-«подложек»;

Generate Terrain Leveler Info Bitmap – сгенерировать файл расположения зон выравнивания для инструмента выравнивания местности под объектами (в разрешении .bmp);

Generate Location Airfield Report – создает файл-рапорт (data\LocationDatabase\ Plane-Squad-Date_report.csv) по информации находящейся в базе данных аэродромов (даты нахождения эскадрий на аэродромах и даты их перебазирований);

Check Location DB Integrity – проверка и отображение ошибок в базе данных локаций, файл сохраняется в data\LocationDatabase\IntegrityCheck.txt.

4.3. Камеры редактора и навигация в редакторе

4.3.1. Режим 3D:

- горячая клавиша для переключения между 2D и 3D – F9 или через кнопку в инструментах (п.4.1).
- в изометрическом режиме – ЛКМ + движения мышью позволяют осуществлять управление ориентацией камеры;
- ПКМ + движения мышью позволяют перемещать камеру в пространстве в горизонтальной плоскости, дублируется кнопками W, S, A, D – вперед, назад, влево, вправо соответственно, клавиша SHIFT ускоряет движение;
- колесико мыши отвечает за перемещение камеры по вертикальной оси. Направление поворота камеры можно определить при помощи компаса (кружок в левом верхнем углу экрана, зеленая линия соответствует направлению на север, синяя – направлению ветра).

4.3.2. Режим 2D:

В плоском режиме – аналогично изометрическому режиму, за исключением возможности изменить ориентацию камеры. Приближение/удаление (зуммирование) производится с помощью колесика «мышь».

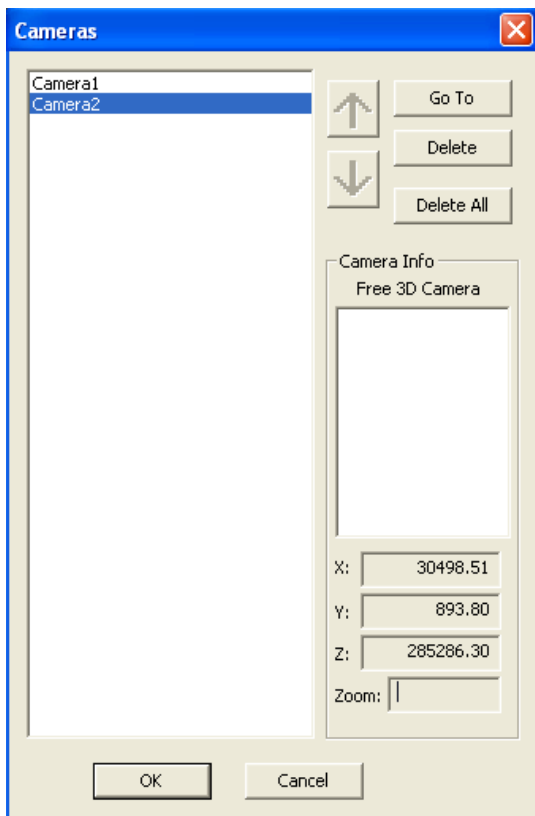
Для быстрого переключения между участками карты можно использовать предустановленные камеры. Для этого кликаем ПКМ в нужной карте и в контекстном меню выбираем Add Camera... (дублируется сочетанием клавиш Ctrl+Alt+C), в появившемся окне написать имя камеры и нажать ОК. Для того, чтобы перейти к нужной камере, надо в контекстном меню по ПКМ выбрать пункт Go to Camera и в выпавшем списке выбрать нужную камеру.

Важно: (!) Камера при переключении будет работать в том режиме, в котором она была установлена – т.е. в случае если камера была установлена в изометрическом режиме, она будет переключать редактор в изометрический режим, каждый раз, как мы переходим к ней, тоже верно и для плоского режима(!).

4.3.3. Камеры (Cameras)

Список предустановленных камер может быть просмотрен и отредактирован в окне Cameras (Ctrl+Shift+c или ПКМ =>Cameras).

После добавления камеры в список камер редактора, можно при помощи горячих клавиш Ctrl+1, Ctrl+2...0 перемещаться на позицию камеры, при этом порядковый номер (очередность вызова камеры по горячей клавише) можно редактировать перемещая камеры в списке при помощи стрелок вверх и вниз в меню Cameras.



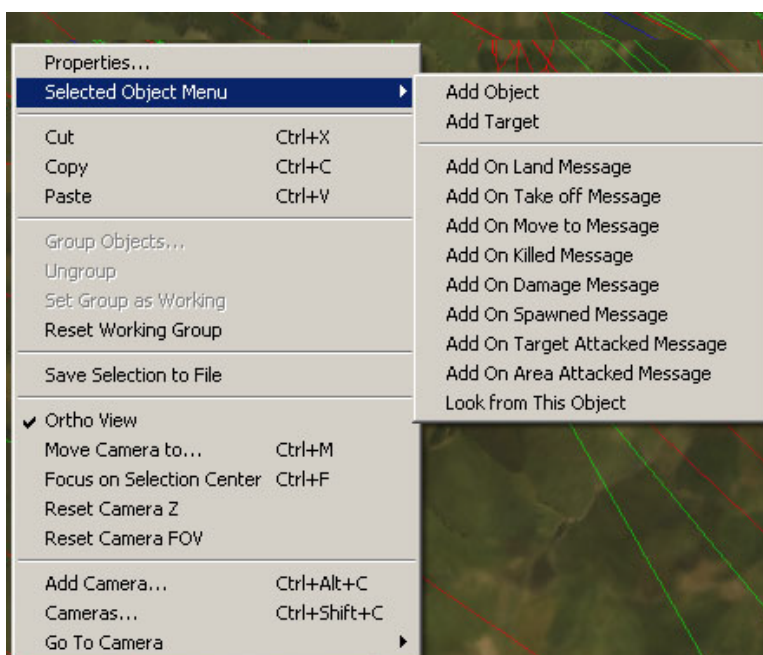
В данном окне:

Кнопка Go To позволяет переключиться на камеру, выбранную в поле списка в левой части окна.

Кнопки Delete и Delete All удаляют выбранную камеру и все камеры соответственно.

Группа Camera Info отображает тип камеры (Free 3D для камер, установленных в изометрическом режиме, Ortho для камер, установленных в плоском режиме), заметку по камере (может быть добавлена/редактирована вручную) и координаты камеры (поля X:, Y:, Z:). Поле Zoom: показывает величину приближения камеры к поверхности (только для Ortho камер).

4.4. Ниспадающее меню по правой кнопке (ПКМ) мыши.



Если нажать ПКМ в любом месте карты, появляется ниспадающее меню, в котором:

Properties... - открывает окно редактируемого набора свойств (см. выше);

Selected Object Menu – меню действий над выбранным объектом:

Add Object – назначение объектной связи;

Add Target – назначение целевой связи;

Add On Land Message – назначение связи в виде сообщения о факте посадки;

Add On Take off Message - назначение связи в виде сообщения о факте взлета;

Add On Move Message - назначение связи в виде сообщения о факте движения;

Add On Killed Message - назначение связи в виде сообщения о факте «смерти»;

Add On Damage Message - назначение связи в виде сообщения о фактически полученных повреждениях;

Add On Spawned Message - назначение связи в виде сообщения о факте «рождения»;

Add On Target Attacked Message - назначение связи в виде сообщения о факте атаки цели;

Add On Area Attacked Message - назначение связи в виде сообщения о факте атаки зоны;

Look from This Object – переход в 3D режим и вид с выбранной сущности.

Cut – вырезать.

Copy – скопировать.

Paste – вставить.

Group Objects... – группировка выделенных объектов в одну группу.

Ungroup – разгруппировка выделенных сущностей.

Set Group as Working – возможность редактировать группу.

Reset Working Group – выход из режима редактирования группы.

Save Selection to File – сохранение выделенных сущностей в отдельный файл.

Ortho View – плоский режим.

Move Camera to... – перемещение камеры в заданные координаты.



Где X и Z координаты карты, в которую переместится камера, после нажатия ОК

Focus on Selection Center – центрирование взгляда (фокусировка) на выделенной сущности.

Reset Camera Z – сброс наклона камеры по горизонтали в нормальное положение.

Reset Camera FOV – сброс угла зрения камеры в дефолтное.

Add Camera – добавить камеру.

Cameras... – редактирование настроек камер в списке предустановленных камер.

Go to Camera – выбор и перемещение в камеру, выставленную с помощью функции Add Camera.

4.5. Нижнее сервисное окно редактора.

Mouse(m):	1855.35	80815.08	Grid(m):	5000	Camera (X Z Zoom):	47273.45	86757.64	2.758	
	1	2		3		4	5	6	7

Располагается в нижней части окна редактора:

1) – координата курсора мыши в редакторе по оси X;

2) – координата курсора мыши в редакторе по оси Z;

3) – Grid(m) – размер ячейки (квадрата) сетки в редакторе при текущем увеличении;

4) - Camera (X Z Zoom): - координаты и положения камеры от первого лица в редакторе:

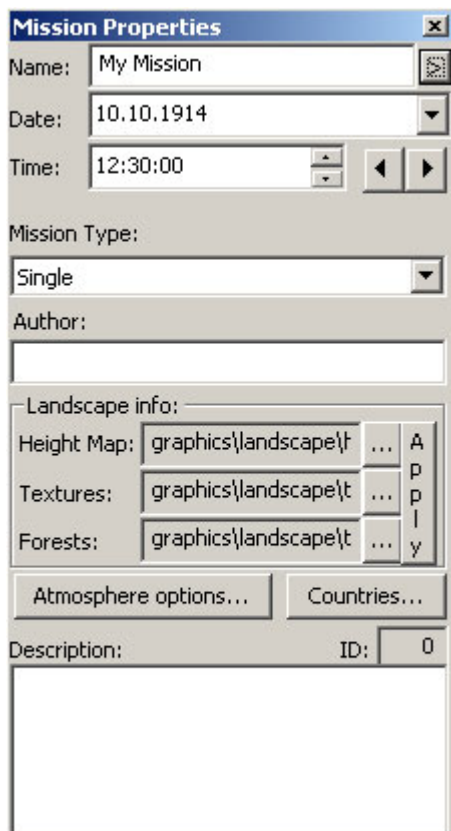
- 5 – координата камеры по X;

- 6 – координата камеры по Z;

- 7 – координата камеры по Y (высота), в 2D режиме в этом окне указывается значение Zoom карты.

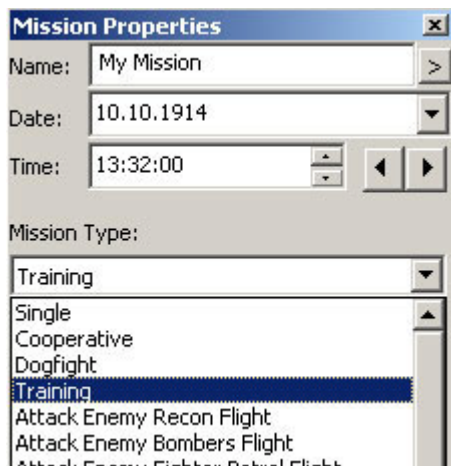
5. Миссия.

5.1. Базовые свойства и настройки миссии.



При создании миссии, в левом нижнем окне редактора находится окно для настройки свойств миссии:

- Name – имя миссии, причем именно это имя будет отображаться в списке миссий в интерфейсе игры, если имя пустое миссия будет отображаться по имени файла миссии;
- Date – установка даты в миссии, можно выставить дату в окне миссии или можно выставить дату через календарь, который появится при нажатии на свойства даты (справа в окне установки даты);
- Time – установка стартового времени миссии;
- Mission Type – выставление типа миссии;



При создании миссий используйте только первые четыре типа миссий (Single, Cooperative, Dogfight и Training), остальные типы миссий в списке используются генератором миссий.

При создании миссии, нужно корректно выставлять тип миссии, потому что, если например, установить тип миссии как Cooperative и поместить эту миссию в каталог для одиночных миссий (data\missions) то эта миссия не будет видна в списке миссий в интерфейсе игры.

- Author – поле для указания автора миссии;

- Landscape Info – раздел для подключения другого типа карты;

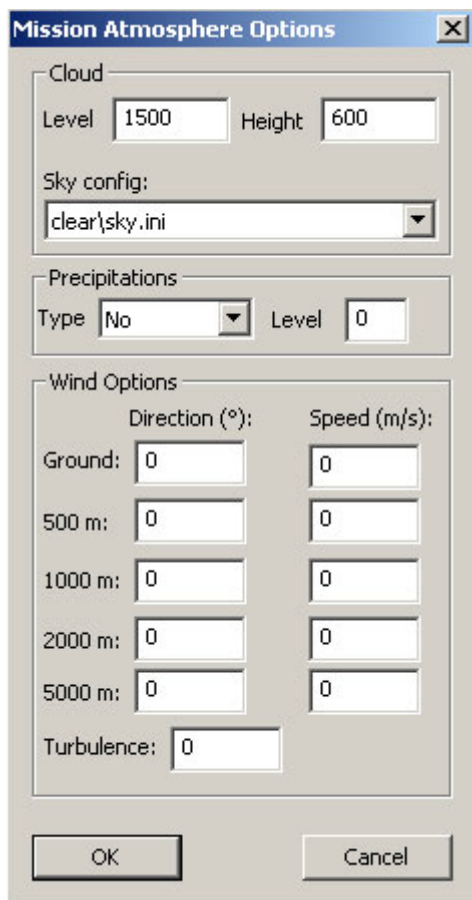
При подключении другого типа карты, все файлы описывающие новую карту должны находится внутри каталога data\graphics\[terrain_name] в нашем случае карта находится внутри папки Landscape и описывается при помощи трех файлов:

Height Map – (height.ini) файл в котором описываются свойства поверхности карты, дороги, водные поверхности, расположение лесных массивов, а так же описывается набор текстур для покрытия поверхности карты.

Textures – (textures.ini) файл в котором описывается набор текстур для составляющих элементов карты, таких как леса, вода, маски поверхностей, текстуры подложек под города и пр.

Forests – (trees\summer.wds) файл, в котором задается вид деревьев в соответствии с сезоном, в данном случае это летний вид деревьев.

Atmosphere Options – настройки погоды в миссии;



Cloud – подраздел настроек облаков и типа погоды:

- level – высота нижней кромки облаков над поверхностью земли;

- Height – толщина слоя облачности;

Sky config – выбор типа погоды, clear – безоблачно, average – редкая облачность, heavy – облачно, precipitation – сплошная облачность, thunderstorm – сплошная облачность и туман.

Precipitations – раздел настроек осадков и их интенсивности:

- Type – выбор типа осадков, дождь/снег;

- Level – выбор интенсивности осадков, от 0 до 10-ти.

Wind Options – раздел для настройки ветра в миссии:

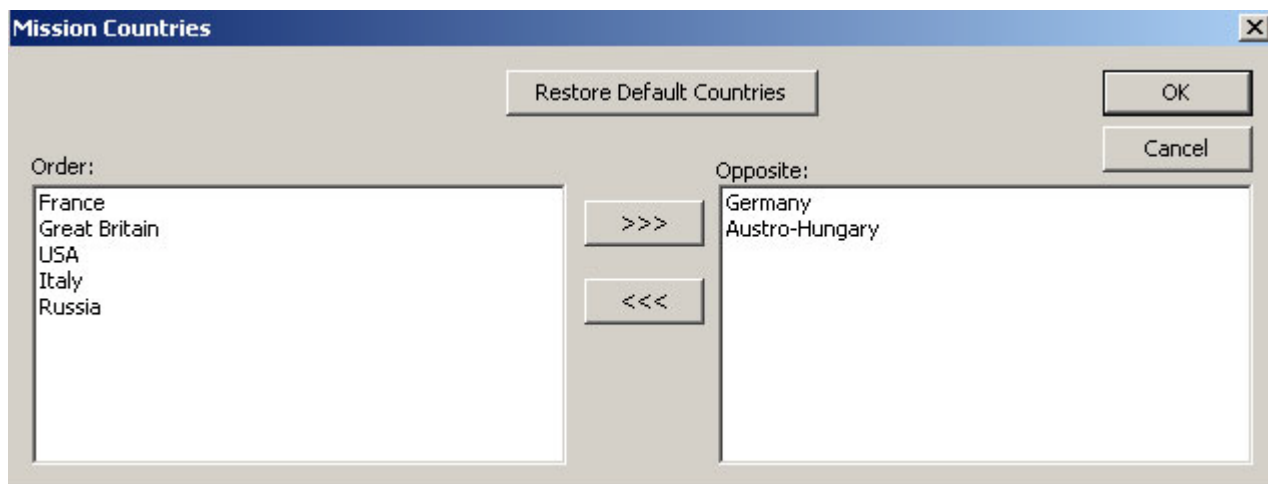
- Direction – направление ветра (куда дует);

- Speed (m/s) – скорость ветра;

- Ground/500m/1000m/2000m/5000m – слои ветра по высотам, соответственно справа настройка направления и силы ветра в слоях;

- Turbulence – сила турбулентности.

Кнопка Countries – настройка коалиций в миссии:



При помощи кнопок [>>>] и [<<<] можно добавлять вышеперечисленные страны в коалиции Order или Opposite.

Restore Default Countries – сбросит настройки коалиций по умолчанию.

Окно Description – в это окно можно ввести описание миссии, которое будет отображаться в интерфейсе игры при выборе миссии в списке миссий, а так же как текст в описании миссии при загрузке миссии.

5.2. Файлы миссии, файлы локализации миссии, их формат.

При создании миссий редактор создает несколько файлов.

5.2.1. *.Mission – файл миссии в текстовом формате, фактически – это исходник миссии, поэтому старайтесь всегда сохранять его если вы планируете использовать миссию в будущем.

Содержание текстового формата миссии на примере пустой миссии (файл миссии *.Mission можно открыть в любом текстовом редакторе):

```
# Mission File Version = 1.0;
```

Options

```
{  
  
  LCName = 0; //индекс строки локализации имени миссии  
  
  LCDesc = 1; // индекс строки локализации описания миссии  
  
  LCAuthor = 2; // индекс строки локализации автора миссии  
  
  PlayerConfig = ""; //скрипт самолета доступный для игрока в миссии  
  
  Time = 13:32:0; //начальное время в миссии  
  
  Date = 10.10.1914; //дата в миссии  
  
  HMap = "graphics\landscape\height.ini"; //файл описывающий карту миссии;  
  
  Textures = "graphics\landscape\textures.ini"; //файл описывающий карту миссии;  
  
  Forests = "graphics\landscape\trees\summer.wds"; //файл описывающий карту миссии;  
  
  MissionType = 3; // тип миссии, в данном случае 3=Training (0=Single, 1=Cooperative, 2=Dogfight)  
  
  CloudLevel = 1500; //высота облачности  
  
  CloudHeight = 600; //толщина слоя облачности  
  
  PrecLevel = 0; //интенсивность осадков  
  
  PrecType = 0; //тип осадков  
  
  CloudConfig = "clear\sky.ini"; //тип погоды  
  
  Turbulence = 0; //сила турбулентности  
  
  WindLayers //настройки ветра по слоям (высота : направление : скорость)  
  {  
  
    0 : 0 : 0;  
  
    500 : 0 : 0;  
  
    1000 : 0 : 0;  
  
    2000 : 0 : 0;  
  
    5000 : 0 : 0;
```

```

}

OppositeCountries //индексы стран относящихся к Opposite коалиции

{

    501; //Германия

    502; //Австро-Венгрия

}

OrderCountries //индексы стран относящихся к Order коалиции

{

    101; //Франция

    102; //Англия

    103; //США

    104; //Италия

    105; //Россия

}

}

# end of file

```

5.2.2. *.msnbin – файл миссии в бинарном формате. Функционирование данного типа файла зависит от формата миссий (читай – если выходит обновленная версия проекта – этот файл будет не работоспособен). В случае если, редактор выдает сообщение об ошибке при чтении миссии, то, если у вас остался исходник миссии – удалите *.msnbin файл и пересохраните миссию, редактор автоматически создаст новую версию бинарного файла.

Так же, бинарная миссия грузится значительно быстрее, чем текстовая.

Загрузка миссии произойдет как при наличии обоих типов файлов миссии (причем *.msnbin имеет приоритет чтения над *.Mission) так и при наличии каждого типа файлов по отдельности.

5.2.3. [mission_name].eng (или *.rus, *.ger, *.fra, *.spa) – локализационные файлы миссии в которых хранятся все текстовые данные, которые использует миссия. Соответственно, если вы создали миссию на русском языке и хотите создать, например, английскую версию локализации, то просто создайте копию данного файла и поменяйте его расширение с *.rus на *.eng.

5.2.4. [mission_name].list – файл необходимый для корректной работы кооперативной миссии (см.п.8).

5.2.5. Работа с локализационными файлами миссии.

При работе с локализацией будьте внимательны!

Работайте (изменяйте или пересохраняйте в редакторе) миссию в том языке (settings.exe – язык) в котором вы создали миссию, иначе, если вы откроете и пересохраните миссию в редакторе, переключенном на другой язык – могут возникнуть проблемы с ID строк локализации.

Открыть файл локализации можно в простом блокноте, т.к. они в текстовом формате

Формат файлов локализации должен быть UCS-2 Little Endian.

5.2.6. ID (индекс) локализации.

При создании миссии, все тексты миссии сохраняются в локализационные файлы миссии, в которых блокам текста присваиваются ID (фактически внутри файла это выглядит как цифра с двоеточием – как индекс строки).

Соответственно, если вы создаете несколько видов языков локализации миссии, следите за тем, чтобы ID ВСЕХ локализационных файлов совпадали и имели абсолютное соответствие ID оригинальному файлу локализации. Так же, при пересохранении миссии – проверяйте ID, т.к. при добавлении новых объектов они могут измениться (хотя и не должны, но лучше это постоянно контролировать чтобы избежать проблем).

Пример файла локализации ([mission_name].rus) :

0:Внимание! Приближаются истребители противника! Перехватите их!

1:Отличная работа! Противник повержен!

2:Истребители противника

3:

4:Старт в воздухе

5:Будьте готовы к действию - следуйте по маршрутным точкам!

6:

7:Основная задача провалена

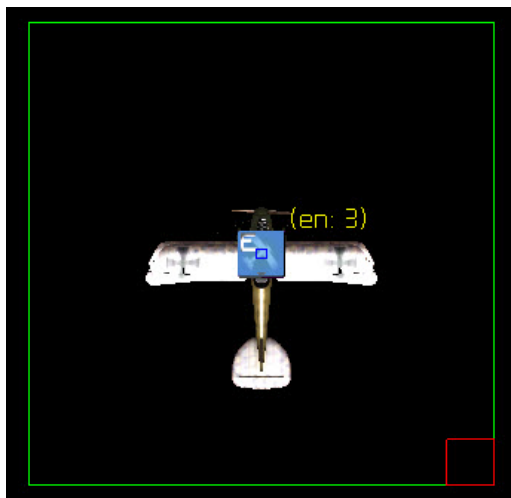
8:Наши разведчики должны добыть очень важные сведения о войсках противника, не допустите к ним истребители противника!

6. Работа с объектами.

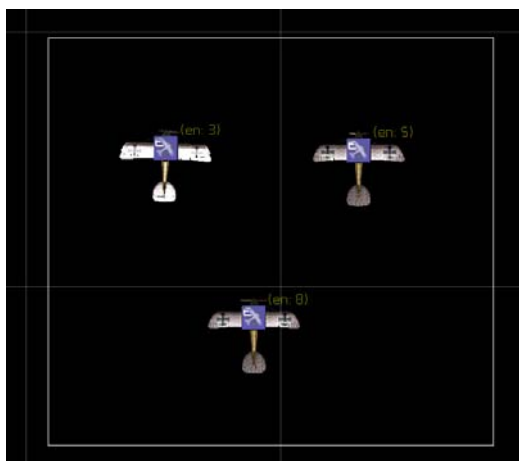
6.1. Выделение объектов.

Выделите объект левой кнопкой мыши, при этом объект выделится зеленой рамкой (1).

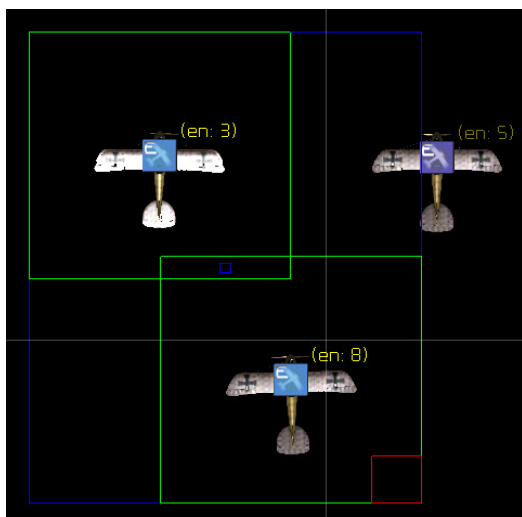
1)



2)



3)



Можно выделить сразу несколько объектов рамкой (2), или можно при зажатом Ctrl добавлять объекты в рамку выделения, при этом каждый объект будет иметь свою рамку зеленого цвета, а вся группа выделенных объектов, будет выделена синей рамкой с центром группы, обозначенным маленьким синим квадратом (3).

6.2. Перемещение и ориентация объектов.

Перемещение и ориентация объектов в редакторе может быть произведено как через окно Basic Properties (см.п.7.2), так и при помощи мыши и клавиш.

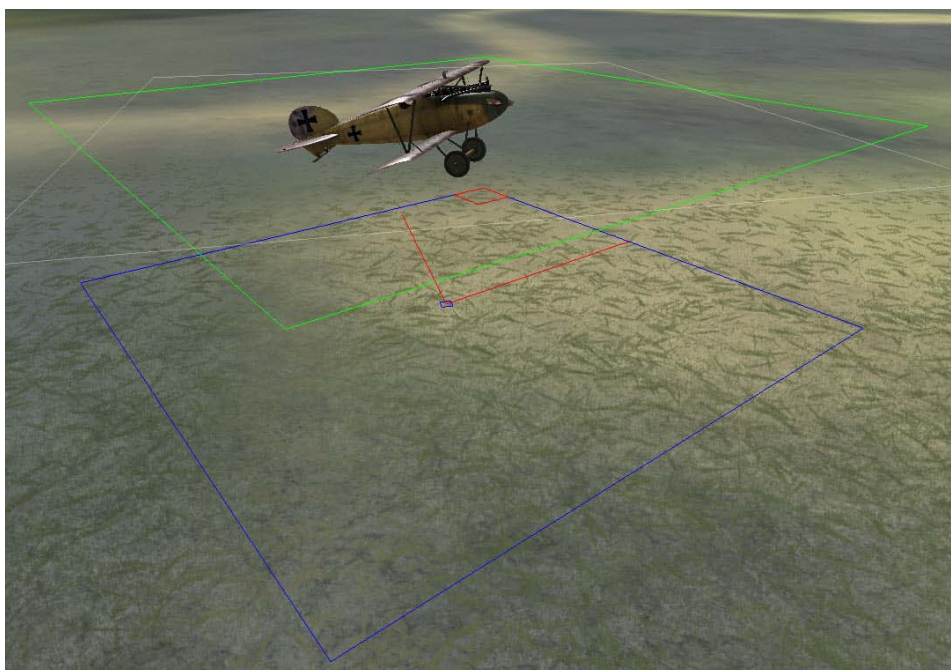
2D режим.

Для перемещения объекта необходимо его выбрать и, с зажатой ЛКМ, переместить его в нужное место. Для поворота объекта вокруг своей оси – потянуть за красный квадрат в углу рамки вокруг выбранного объекта (курсор при этом поменяется на букву R), при этом две красные линии от центра будут визуальнo показывать угол поворота, а серая рамка – прошлое положение объекта.

Для перемещения объекта при помощи клавиатуры используются «стрелки» на клавиатуре. Выделите объект, зажатая клавиша Shift увеличивает скорость перемещения. Зажатая кнопка Ctrl уменьшает скорость перемещения. Кнопки «<» и «>» вращают объект вокруг своей оси.

Кнопки PgUp и PgDown регулируют высоту (YPos).

3D режим.



При выделении объекта в 3D режиме зеленая рамка показывает фактическое положение объекта, а синяя рамка показывает проекцию положения на поверхность.

Для перемещения объекта необходимо его выбрать и, с зажатыми Shift+ЛКМ, переместить его в нужное место.

Для поворота объекта вокруг своей оси, выделите его и с зажатыми Ctrl+ЛКМ подвигайте мышкой (курсор при этом поменяется на букву R).

В 3D режиме так же работают горячие клавиши для перемещения объекта (как и в 2D). То есть это стрелки для перемещения объекта по X и Z, клавиши PgUp (вверх) или PgDown (вниз) для перемещения объекта по Y, клавиши “<” и “>” для вращения объекта, при этом, замедление перемещения +Ctrl, а ускорение перемещения объекта +Shift.

6.3. Копирование, вставка, удаление.

Любой выделенный объект (или группа любых выделенных объектов) в редакторе могут быть скопированы в буфер и вставлены в любом месте на карте.

Горячие клавиши Ctrl+C – копировать, Ctrl+V – вставить. Эти же действия можно произвести с помощью ниспадающего меню по ПКМ. В этом случае объекты разместятся в точке, где находится курсор мыши.

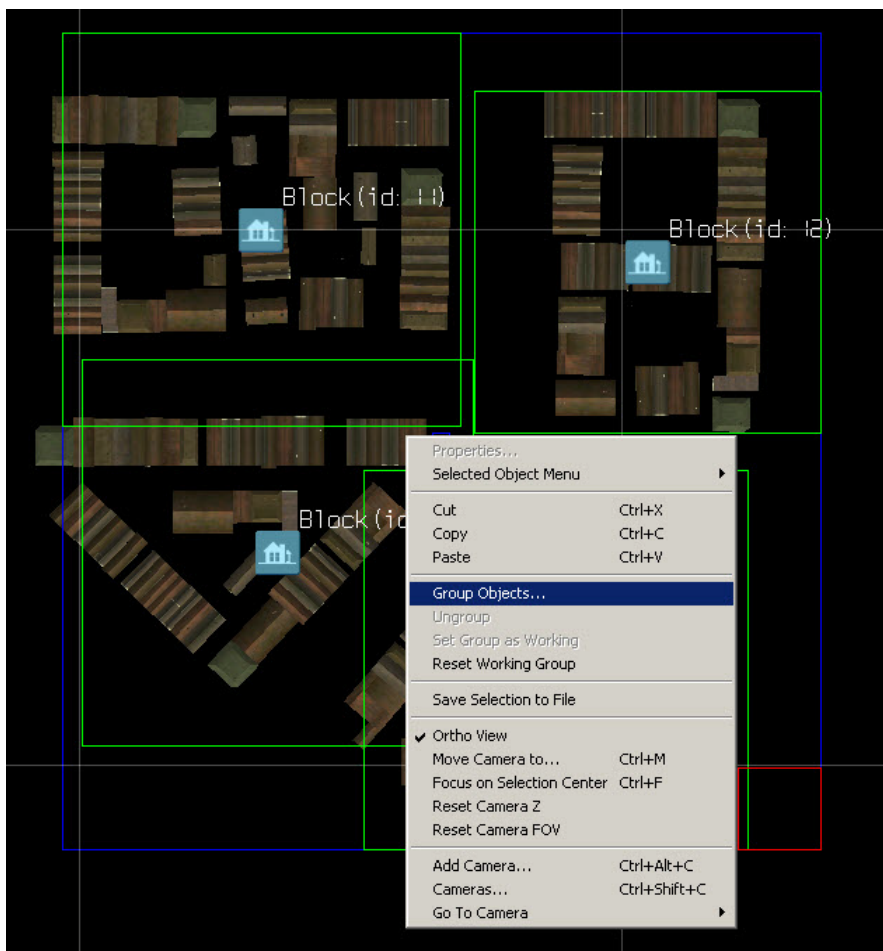
Так же, можно повторно выставить один и тот же тип объекта на карту, если при выборе объекта из библиотеки объектов и его установке на карту держать Ctrl зажатым

6.4. Группирование объектов.

Группирование объектов помогает создавать из большого количества объектов миссии создавать своего рода блоки внутри миссии, для более удобной работы с ними.

Группирование объектов можно производить как в 2D, так и в 3D режимах.

Для того, чтобы сгруппировать объекты на карте, выделите эти объекты, ПКМ => Group Objects, при этом, все объекты объединятся под иконку Group, и последующее их перемещение по карте можно будет производить через неё, т.е. фактически «сшивает» объекты. Это очень удобно, например, при создании большого города из большого количества мелких блоков, или при создании какого-либо замкнутого функционального блока миссии.





Можно «разгрузить» пространство карты, отключив через фильтр объектов (раздел 4.1) блоки, при этом группа будет идентифицироваться под иконкой Group.

Так же можно редактировать уже созданные группы.

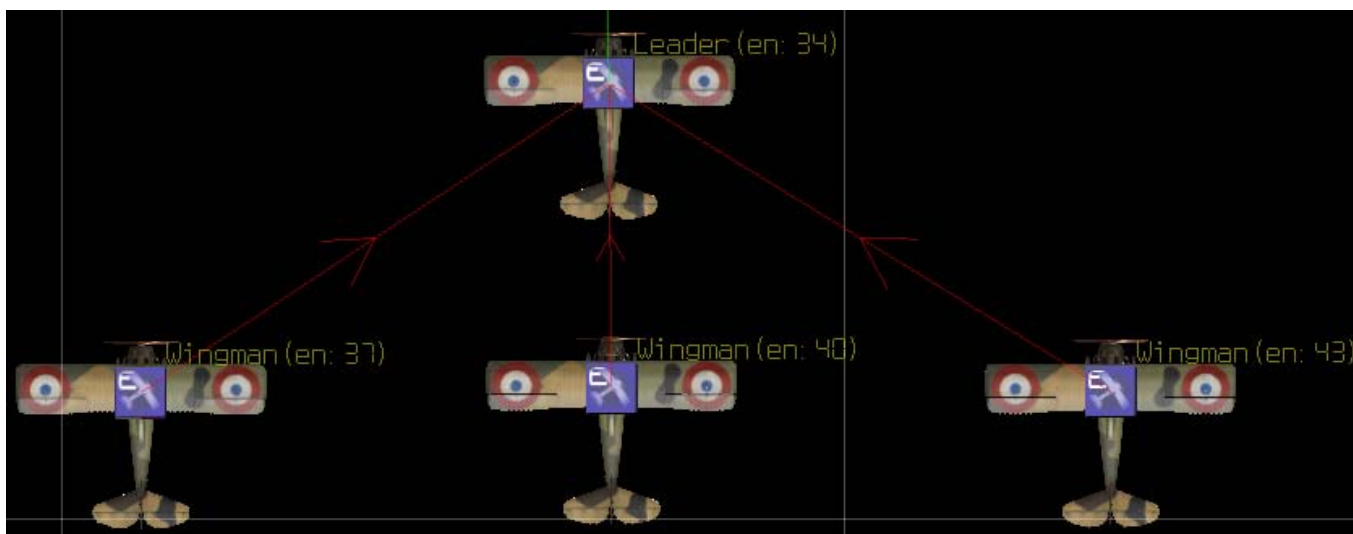
Для этого выделите группу, ПКМ=> Set Group as Working (если вы собираетесь добавить/или подвинуть блоки внутри группы – не забудьте включить соответствующий фильтр в фильтре объектов, иначе вы не сможете двигать объекты).

После установки Set Group as Working, группа выделится желтой рамкой, в этом режиме вы можете передвигать, удалять и добавлять объекты в группе. Для этого просто установите объект в рабочее поле редактора, он автоматически добавится в рабочую группу.

Обязательно!: - не забывайте закрывать режим редактирования группы (т.к. данный режим предполагает работу только с этой группой в текущий момент времени), для этого ПКМ=>Reset Working Group.

6.5. Объединение движущихся объектов в звено или группу.

В данном случае под группой понимается, что это несколько объектов (в приведенном примере самолетов), которые подчинены одному лидеру и выполняет все команды, которые в свою очередь, получает лидер.



Для объединения любых движущихся объектов в звено или группу, нужно «прикрепить» все подчиненные объекты группы к лидеру группы целевой связью.

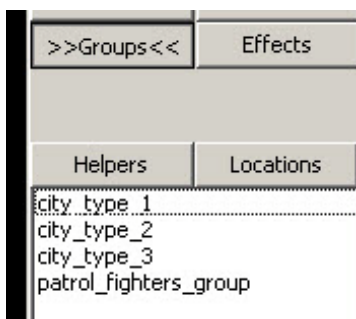
При этом, лидер будет транслировать все команды поступающие из миссии своим подчиненным, т.е. дальнейшая работа миссии с группой идет через лидера группы и все непосредственные команды отдаются лидеру группы (существуют некоторые примечания, о них будет сказано далее).

Таким же образом производится объединение всех остальных объектов таких как автомобили, танки и артиллерия, за исключением поездов и аэростатов(см. п. 7.2.1.).

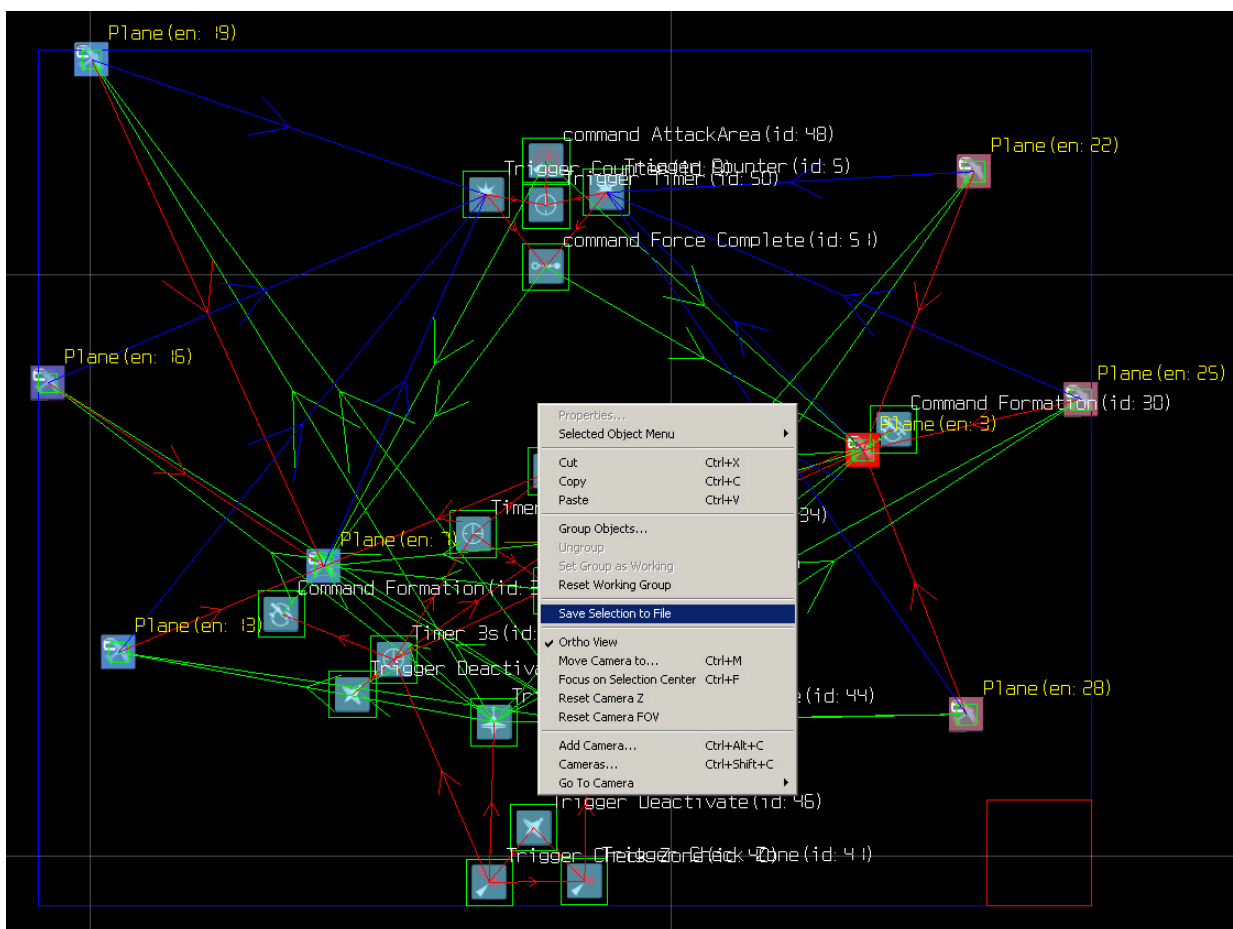
Существует возможность настройки положения ведомых в группе (см. п. 7.2.1.) а так же выдачи команды на тип построения в звене или группе (см. п. 7.7.8.).

6.6. Сохранение в файл, импорт и вставка сохраненных групп.

6.6.1 Сохранение в файл



Существует возможность сохранения в файл любых частей миссии, сохраненные файлы по умолчанию находятся и сохраняются в директорию data/missions и при этом видны в списке раздела Groups в правой части экрана редактора (Objects Library).



Для сохранения объектов в файл, ПКМ=> Save Selection to File

При этом, вы можете вставлять сохраненные группы из раздела Groups в миссию как обычные объекты (функциональные части миссии, заготовки, стандартные группы и т.п.), а так же импортировать их.

6.6.2 Импорт объектов в миссию.

Так же можно импортировать сохраненный в *.Group файл объекты миссии в любую другую миссию.

В этом случае, при импорте (Import from File...п. 4.2.1.) такого блока в следующую миссию, все объекты и сущности импортируются именно в те координаты и места, где они были сохранены в файл.

Это очень удобно использовать в случае, если вы хотите наложить шаблон одной миссии на другую миссию, таким образом можно делать параллельную работу над одной миссией или импортировать в миссию заранее созданные шаблоны миссии.

7. Объекты редактора.

Каждый MCU или объект, поставленный на карту в редакторе, имеет набор редактируемых свойств.

Редактор позволяет производить групповое редактирование набора базовых свойств (далее Basic Properties) одинаковых объектов, т.е. можно выделить рамкой или по CTRL несколько одинаковых объектов и задать им всем одинаковые свойства.

Окно Basic Properties автоматически открывается при выборе объекта или MCU и по умолчанию находится в левом нижнем левом углу окна редактора.

7.1. Раздел Objects Library редактора

Planes	Buildings
Vehicles	Blocks
Trains	Airfields
Ships	Battlefield
Artillery	Firing-point
Aerostats	>>Bridges<<
MCUs	Grounds
Groups	Effects
Helpers	Locations

Planes: объекты - самолеты.

Vehicles: объекты редактора - наземные объекты, автомобили и танки.

Trains: объекты - поезда и вагоны.

Ships: объекты – корабли.

Artillery: объекты – полевая и гаубичная артиллерия, зенитная артиллерия, пулеметы и зенитные пулеметы.

Aerostats: объекты – наблюдательные аэростаты.

MCUs: объекты миссии – Mission Command Units, триггеры, команды, трансляторы.

Groups: объекты – сохраненные в *.Group файлы в папке data/missions (см. п. 6.6.).

Buildings: объекты – отдельно стоящие наземные постройки.

Blocks: объекты – постройки объединенные в кварталы, базовые блоки городов, деревни, церкви.

Airfields: объекты – аэродромы или блоки аэродромов.

Battlefield: объекты– элементы поля боя, колючая проволока, огневые позиции, палатки.

Firing point: объекты – огневые позиции, ДОТы, ДЗОТы.

Bridges: объекты – мосты, автомобильные (HW) и железнодорожные (RW) разной длины и типов.

Grounds: - объекты из раздела Grounds

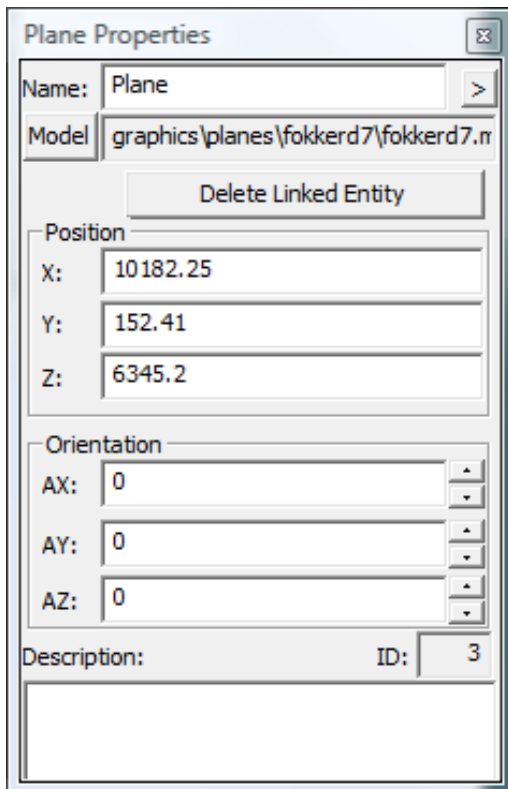
Effects: объекты – эмиттеры эффектов в миссии, звуки, дым, взрывы и прочие (см. п. 7.6.6.).

Helpers: объекты предназначенные для интерфейсной связи с генератором миссий (см.п.7.2.5.).

Locations: объекты предназначенные для расстановки типов локаций в базу данных, которую использует генератор миссий (см.п.7.2.6.).

7. 2. Basic Properties для объектов редактора.

7.2.1 Planes/Vehicles/Artillery/Aerostats



Кнопка «>» - открывает окно Additional Properties, его мы рассмотрим подробнее ниже.

Name – имя объекта, через данное поле можно задать название самолета, так же, в случае с сетевой миссией – это название слотов для выбора самолета.

Model – поле указывает путь, по которому находится принадлежащая объекту модель, кнопка Model– позволяет назначить другую модель (включая внутренний скрипт) при сохранении всех свойств объекта, причем, при выборе однотипных объектов, например выделить все типа Planes, или все типы Vehicle, то можно целиком отредактировать всю группу объектов, например, заменить в группе самолеты одного типа на самолеты другого типа.

Create Linked Entity/Delete Linked Entity – создает/удаляет сущность объекта в игровом мире.

Ограничение: Объект без сущности не может взаимодействовать с игровым миром, не может быть связан с MCU или другими объектами и даже не будет существовать визуально в миссии. Исключение составляют здания – они отображаются в игре, и объекты могут взаимодействовать с

ними даже в случае, когда сущность для них не создана, однако здания в этом случае не могут менять собственное состояние.

Поэтому, эту кнопку необходимо нажимать сразу же после установки объекта на карту если это объект, который участвует в миссии (самолет, танк и т.п.) выполняет команды миссии или выдает сообщения в миссию. Удаление игровой сущности никакой очевидной смысловой нагрузки не несет и может быть использована, разве что, для обрыва всех связей объекта.

Поля Position X:, Y:, Z: - указывают на координаты текущего местоположения объекта в игровом мире, где Y – высота относительно уровня моря.

Поля Orientation AX:, AY:, AZ: - указывают ориентацию объекта в игровом мире. Значения меняются по часовой стрелке от 0 до 360 градусов. Ось X соответствует оси крена (поворот объекта вдоль его продольной оси), ось Y оси курса (поворот объекта относительно его центра в горизонтальной плоскости), ось Z оси тангажа (изменение угла между продольной осью объекта и горизонтальной плоскостью, направление вращения через верх).

Поле Description – используется для занесения какого либо текстового описания или справки по объекту.

В окне Objects Library ЛКМ на кнопку Planes и в ниспадающем меню выбираем самолет. Далее, ЛКМ на карте в желаемом месте. Затем, чтобы «оживить» и настроить свойства этого объекта необходимо:

ЛКМ на иконку самолета → окно Plane Properties (в левом нижнем углу редактора) → ЛКМ на кнопку Create Linked Entity.

Свойства самолетов (объектов с возможностью быть управляемыми игроком) немного отличаются от свойств других объектов (артиллерии, танков/автомобилей и пр.), но в целом отличия заключаются лишь в отсутствии некоторых полей настройки, которые есть у самолетов.

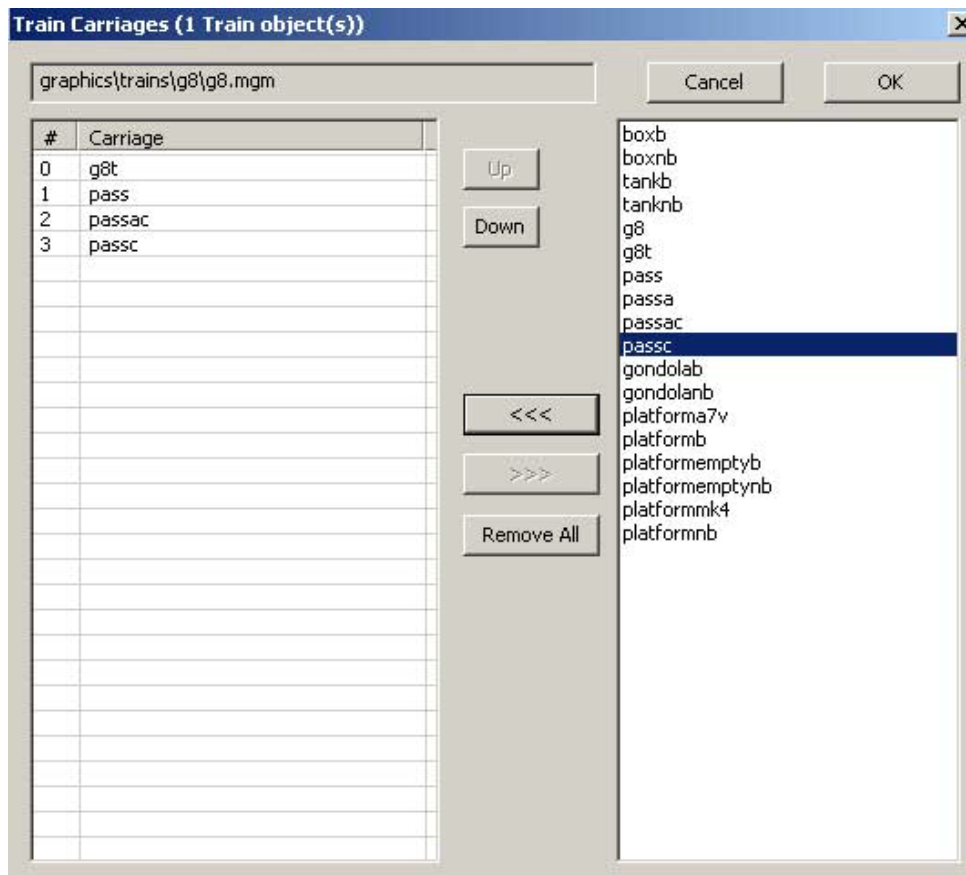
7.2.2. Trains

Отдельно нужно упомянуть объект поезд (Trains), т.к. у поезда есть некоторые особенности в расстановке вагонов.

Для того, чтобы создать поезд в миссии, вам нужно выставить на карту паровоз и создать ему сущность.

Для того, чтобы поезд появился в миссии с прицепленными к нему вагонами, в Basic Properties паровоза есть дополнительная кнопка настроек – Carriages.

Нажав на кнопку Carriages вы откроете диалог настройки опций состава поезда:



Вы сможете добавлять (<<<) или удалять (>>>) из состава поезда вагоны. Список прицепленных вагонов отображается в левом окне (Carriage). При этом кнопками Up и Down вы сможете изменять порядок вагонов в составе.

7.2.3. Firing Point

При созданной Entity фактически является самостоятельным наземным объектом, со всеми свойствами наземной техники – только не может передвигаться по полю боя.

В общих чертах, наземные объекты могут полноценно участвовать в миссии как любые другие объекты (самолеты/танки) и влиять на её исход своим состоянием.

The screenshot shows a 3D game environment with a bridge structure. Two windows are open in the foreground:

Bridge Properties

- Name: Bridge
- Model: graphics\bridges\bridge_rr70.mgm
- Create Linked Entity
- Position:
 - X: 48454.18
 - Y: 164.39
 - Z: 108300.28
- Orientation: 0
- Durability: 1000 ID: 27
- Damage:

#	Damage
0	1.000
1	1.000
2	1.000
3	1.000
- Add
- Delete All
- Description:

Edit Damage Level

- Structure ID: 1
- Damage Level: 1
- OK

7.2.5. Helpers

Предназначены для интерфейсной связи между редактором и генератором миссий. Более подробное описание практического применения хелперов читайте в документе «Руководство Пользователя для Редактора Шаблонов ROF»

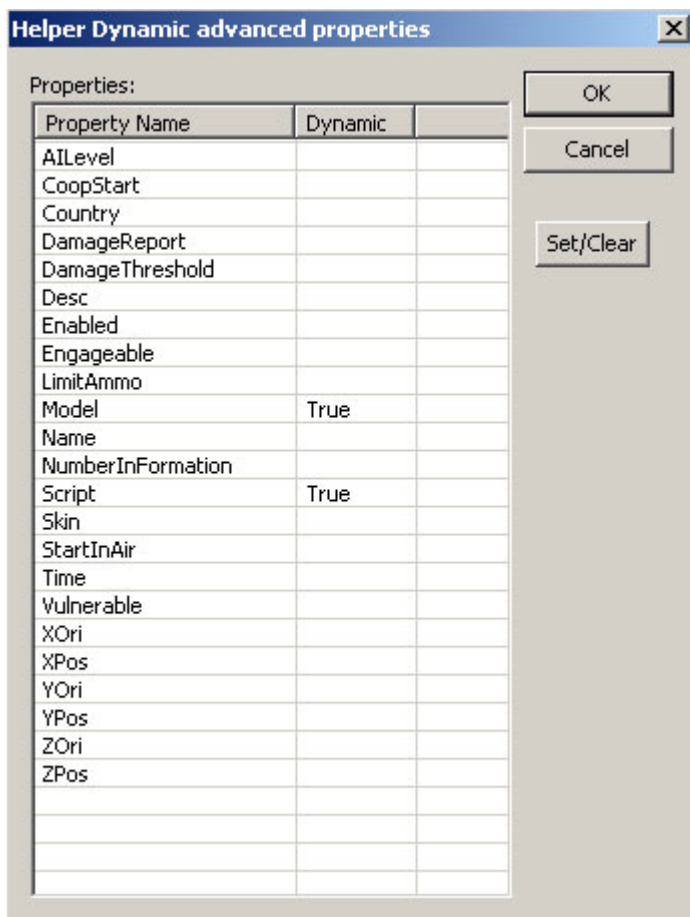
Helper: Dynamic



Задаёт динамические свойства объекту, используется при генерации миссий.

Целевой связью связывается с объектом или группой одинаковых объектов (ЦС с каждым).

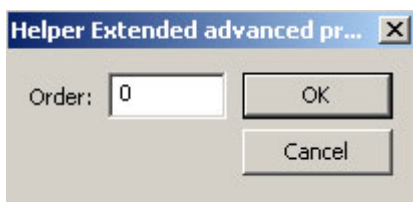
В Advanced Properties, доступен список задаваемых свойств генерируемому объекту “Property Name” и раздел “Dynamic” в котором выбирается набор используемых при генерации свойств, True=использовать и отображается в GUI редактора шаблонов.



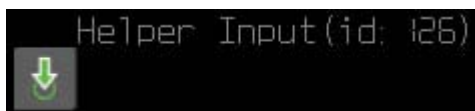
Helper: Extended



Обычно используется для определения/задачи количества объектов в группе. Задаёт для пресета ID объектов, которые добавляются/удаляются из пресете при генерации миссии.



Helper: Input



Обозначает все входящие в пресет связи.

Используется для интерфейсной связи редактора миссий и редактора шаблонов.

Helper: Output



Обозначает все входящие в пресет связи.

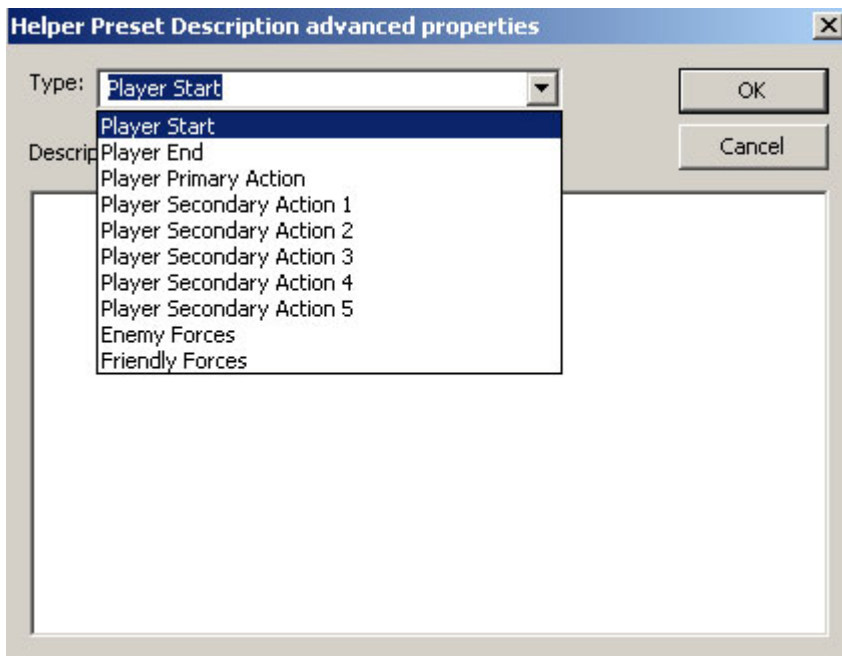
Используется для интерфейсной связи редактора миссий и редактора шаблонов.

Helper: Preset Description

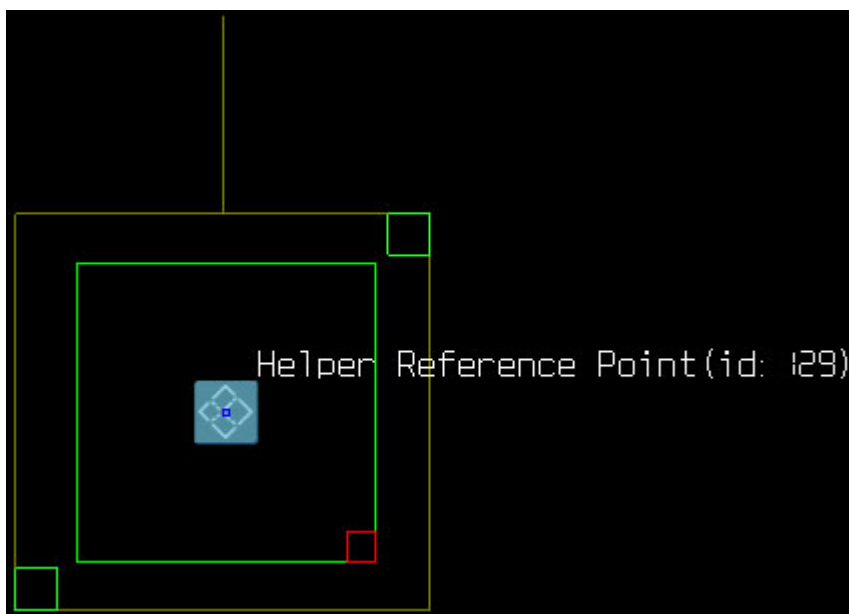


Задаёт текстовое описание пресета для генератора миссий (составная часть брифинга миссии).

В Advanced Properties задается тип пресета - "Type", а в поле "Description Text" вводится текст описывающий пресет.



Helper: Reference Point



Обозначает центр, ориентацию и размер пресета, используется при чтении правил установки пресета в локацию.

Зажав левую кнопку мыши и потянув за зеленые маркеры внешней рамки – можно регулировать размеры Reference Point, нажав на иконку и красный маркер внутренней рамки – перемещать и вращать Reference Point.

Размеры так же можно задать через “Advanced Properties” диалог.

Helper Reference Point advanced property...

Forward: 26.56

Left: 28.15

Right: 27.08

Backward: 26.29

OK Cancel

7.2.6. Locations

Предназначены для создания базы данных локаций, которая используется при генерации миссий.

У всех локаций есть базовый набор “Basic Properties”

Air Objective

Name: >

Position and Orientation

X:

Z:

AY:

Length: Width:

Order Countries: Edit Countries

Opposite Countries:

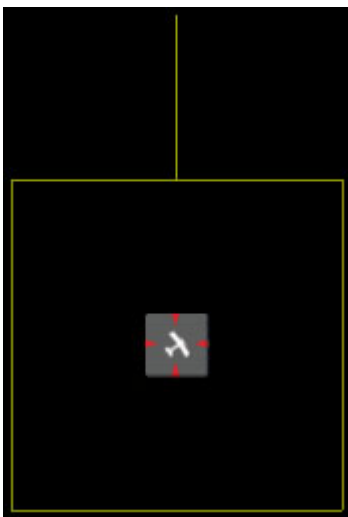
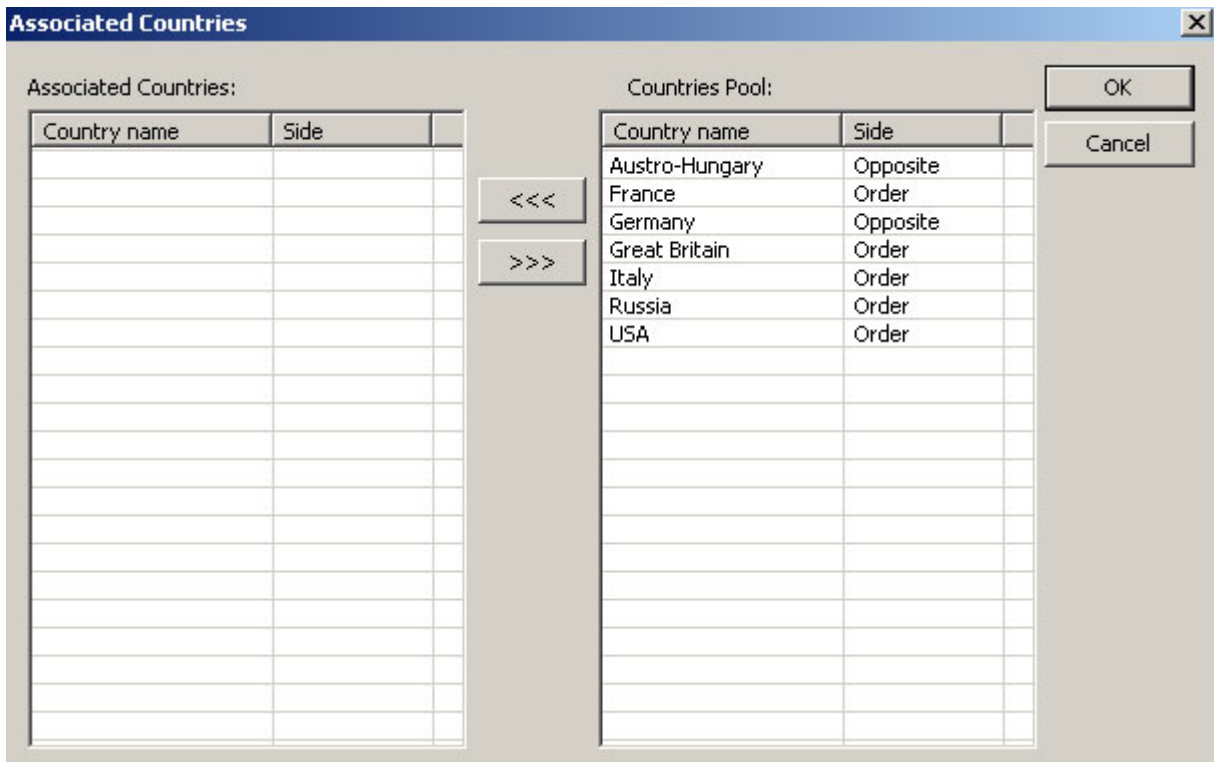
Description:

Name – имя локации.

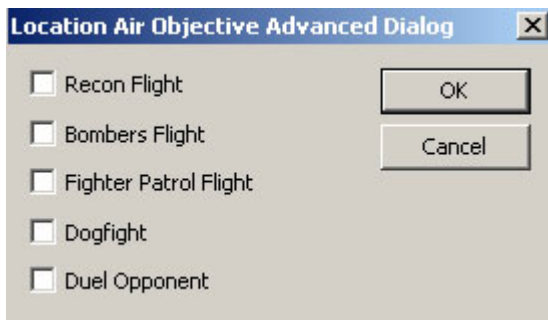
Поля для задачи положения локации (X;Y) и её ориентации (AY).

Length и Width – длина и ширина локации.

Положение и размеры локации используется при чтении правил установки пресета в локацию по Reference Point-у. Направление локации можно поменять нажав левой кнопкой мыши на красный маркер, а размер локации подвигав зеленый маркер.



В “Advanced Properties” задается тип “Air Objective” локации



Recon Flight – тип разведчики противника

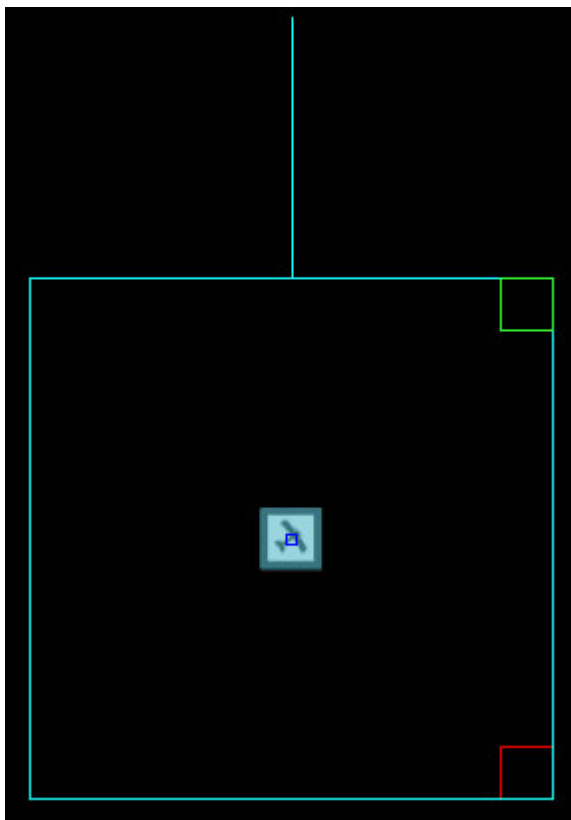
Bombers Flight – тип бомбардировщики противника

Fighter Patrol Flight – тип истребители противника

Dogfight – тип воздушный бой

Duel Opponent – тип дуэль

Airfield



Используется при создании базы данных аэродромов-эскадрилий.

В Advanced Properties задается набор свойств аэродрома, при этом создается база данных которая используется при генерации миссий.

Add Associated Squadron

Country Name	Squadron Name	Plane Types Number	Squad Start	Squad End	
France	Escadrille 167	1	22.08.1918	11.11.1918	
France	Escadrille 154	1	01.03.1918	11.11.1918	
France	Escadrille 152	1	01.07.1918	11.11.1918	
France	Escadrille 124	2	01.12.1917	11.11.1918	
France	Escadrille 103	1	01.04.1917	11.11.1918	
France	Escadrille 102	1	01.03.1918	11.11.1918	
France	Escadrille 99	1	01.01.1917	11.11.1918	
France	Escadrille 94	2	01.03.1918	11.11.1918	
France	Escadrille 88	1	01.03.1918	11.11.1918	
France	Escadrille 87	2	01.05.1918	11.11.1918	
France	Escadrille 84	1	01.06.1917	11.11.1918	
France	Escadrille 81	1	01.12.1917	11.11.1918	
France	Escadrille 77	2	01.02.1918	11.11.1918	
France	Escadrille 73	1	01.11.1917	11.11.1918	
France	Escadrille 68	2	01.03.1918	11.11.1918	
France	Escadrille 67	2	01.04.1918	11.11.1918	
France	Escadrille 65	1	01.02.1918	11.11.1918	
France	Escadrille 48	1	01.02.1917	11.11.1918	
France	Escadrille 38	1	01.10.1917	11.11.1918	
France	Escadrille 37	1	01.02.1918	11.11.1918	
France	Escadrille 31	1	01.09.1917	11.11.1918	
France	Escadrille 26	1	01.03.1918	11.11.1918	
France	Escadrille 23	2	01.08.1918	11.11.1918	
France	Escadrille 15	1	01.06.1917	11.11.1918	
France	Escadrille 12	1	01.02.1918	11.11.1918	
France	Escadrille 03	1	01.08.1917	11.11.1918	
Germany	Jasta 99	1	01.01.1917	11.11.1918	
Germany	Jasta 80	3	15.02.1918	01.07.1918	
Germany	Jasta 65	3	23.01.1918	01.09.1918	
Germany	Jasta 64	3	01.04.1918	01.08.1918	
Germany	Jasta 61	3	30.07.1918	11.11.1918	
Germany	Jasta 35	5	07.01.1917	01.01.1918	
Germany	Jasta 34	5	20.02.1917	01.12.1917	

Squadron Planes Details

Coming Date:

Leaving Date:

При этом выводится список доступных эскадрилий, с количеством типов самолетов на вооружении эскадрильи и датой появления и убытия эскадрильи с фронта.

Соответственно, в поле Coming Date – задается дата прибытия, в поле Leaving Date – дата убытия.

Через Squadron Planes Details, можно увидеть список самолетов эскадрильи по периодам

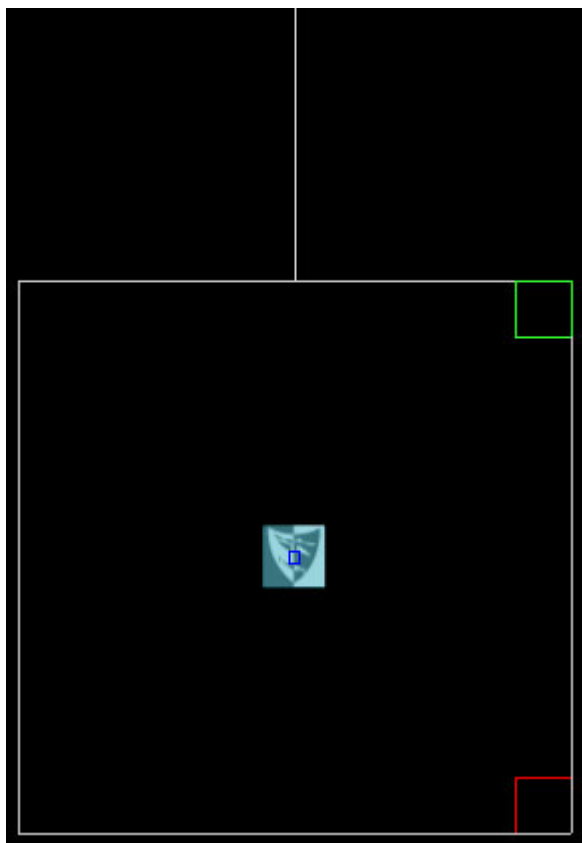
Squadron Plane Info

Squadron Name:

Squadron Country:

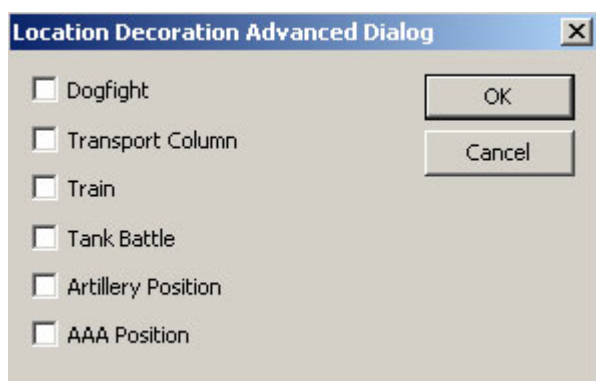
Plane Name	Start Date	End Date
SPAD XIII	01.07.1918	11.11.1918

Decoration



Используется при установке соответствующих типов пресетов в миссию, в данном случае – “Decoration”.

В “Advanced Properties” задается тип “Decoration” локации



Dogfight – тип воздушный бой

Transport Column – тип транспортная колонна

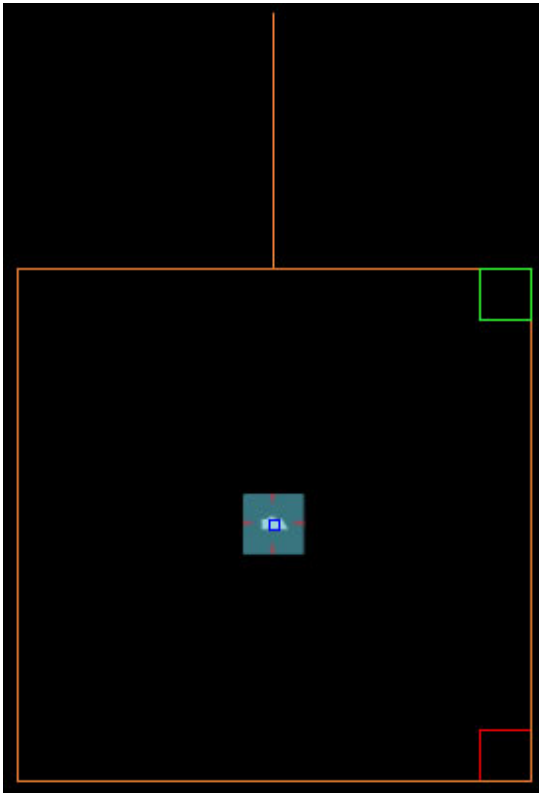
Train – тип поезд

Tank Battle – тип танковый бой

Artillery Position – тип артиллерийская позиция

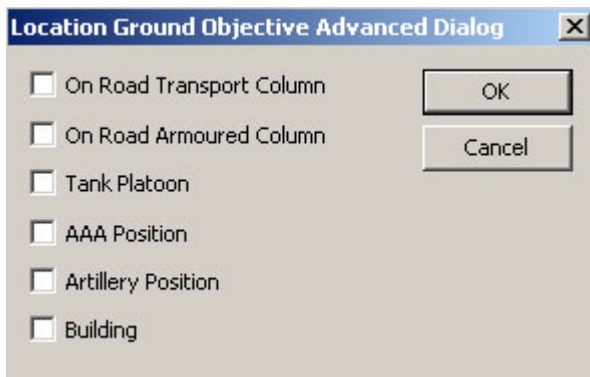
AAA Position – тип зенитная позиция

GroundObjective



Используется при установке соответствующих типов пресетов в миссию, в данном случае – “Ground Objective”.

В “Advanced Properties” задается тип “Ground Objective” локации



On Road Transport Column – тип транспортная колонна на дороге

On Road Armoured Column – тип колонна бронетехники на дороге

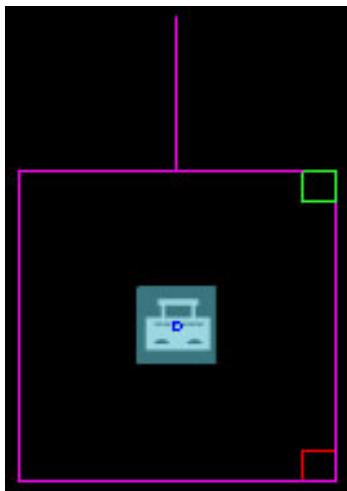
Tank Platoon – тип танковый взвод

AAA Position – тип зенитная позиция

Artillery Position – тип артиллерийская позиция

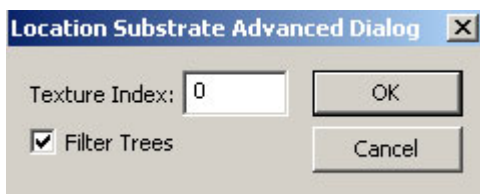
Building – тип наземные постройки

Substrate



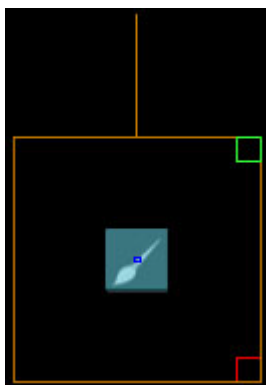
Локация для генерации текстур-подложек, при этом при генерации по команде “Generate Substrate INI file” (см.п.4.2.8) генерируется layers.ini файл, в котором сохраняются координаты текстур подложек на карте, а сами текстуры подложек устанавливаются (становятся видимыми в редакторе) на свои места.

В Advanced Properties



задается индекс текстуры, которая будет использоваться для подложки, а чекбокс “Filter Trees” вырезает/не вырезает деревья при генерации INI файла.

TerrainLeveler



Используется для генерации карты высот, указывает координаты для генерации области выравнивания.

После установки всех локаций, базу данных локаций нужно проверить на ошибки через команду “Check Location DB Integrity” (см.п.4.2.8.). Лог файл проверки сохраняется в файл IntegrityCheck.txt (data\LocationDatabase\).

7.3. Advanced Properties для объектов редактора.

7.3.1. Planes/Vehicles/Artillery/Aerostats

Окно Basic Properties → ЛКМ на кнопку “>” справа от поля Name.

В появляющемся окне (Plane) Advanced находятся параметры настройки выбранного объекта, где:

Plane Advanced (1 object(s))

Linked TR ID: 11 ☒ Enabled

AI: Normal

Country: France

in Formation: 0

☐ Cooperative Start ☐ Start in the Air

☒ Vulnerable ☒ Engageable ☒ Limit Ammo

☒ Damage: (Threshold)

Damage report: 50

☐ AI Return to Base Decision

Script: LuaScripts\WorldObjects\spad13.txt

Skin: <default>

Payload: 0 empty

Route Time (minutes): 60

On Messages Table:

Message Type	Command ID	Target

Target ID: -1

Name:

Description:

Поле Linked TR ID – ID Linked Entity автоматически присваивается редактором при создании Entity для данного объекта;

Поле Enabled – если стоит галочка, объект появится при загрузке миссии сразу, если нет – объект может появиться позже по ходу миссии при помощи Trigger:Activate (см.п. 7.8.1.), тогда, когда это будет нужно в соответствии со сценарием.

Поле AI – настройка уровня AI объекта (Low, Normal, High Ace, Player)

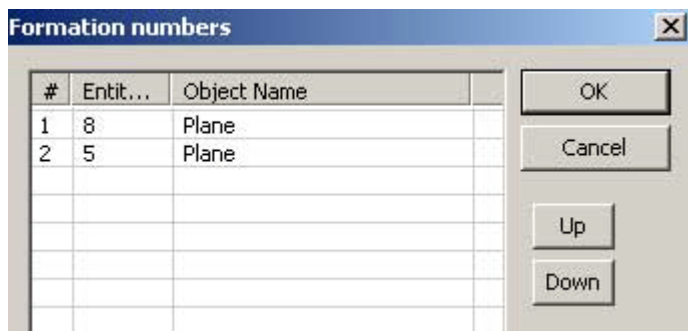
Для того, чтобы в миссии установить самолет игрока – (самолет который будет доступным в миссии для игрока) в поле AI нужно установить Player.

Существует способ динамического изменения опции AI (кроме Player) при помощи команды Behavior (см п. 7.6.3.).

Поле Country – выбор страны принадлежности объекта.

Поле # in Formation – место объекта в формации (группе).

Кнопка Change Formation – изменение порядка (положения внутри группы) ведомых в группе, работает только для ведущего (лидера) группы, выбрав строку, которая указывает порядковое местоположение ведомого в группе, можно перемещать его выше (Up) или ниже (Down) в списке.



Поле Cooperative Start – поставить галочку, если миссия сетевая, это позволяет сделать самолет доступным в лобби (список доступных для выбора самолетов) сетевой игры (см.п.8).

Поле Start in the Air – появление объекта в воздухе (только для самолетов), не забудьте выставить высоту самолета в Basic Properties (Y) для того, чтобы самолет стартовал на заданной высоте.

Поле Vulnerable – уязвимость объекта от повреждений или нет. Существует способ динамического изменения данной опции при помощи команды Behavior (см п. 7.6.3.).

Поле Engageable – будет объект атакован AI или нет, это означает, что при выставлении данной опции на объекте – он не будет атаковаться другими AI (самолетами, зенитками и п.р. объектами) в миссии. Существует способ динамического изменения данной опции при помощи команды Behavior (см п. 7.6.3.).

Поле Limit Ammo – бесконечный боезапас или нет. Существует способ динамического изменения данной опции при помощи команды Behavior (см п. 7.6.3.).

Поле Damage – Threshold или Step. Threshold – реакция на заданный остаток «жизни» у объекта в процентах (заданных в окне Damage report), от начального состояния. Step – пошаговая (циклическая) реакция по заданному отъему «жизни» (заданных в окне Damage report) в процентах.

Поле Damage report – количество процентов для условий Damage Report (Threshold/Step).

Поле Damage report используется для формирования условий срабатывания сообщения Damage Message. То есть, в случае Damage report (Threshold), например 25 - объект выдаст миссии сообщение при условии, когда его «жизнь» уменьшится или пересечет порог в 25% от исходного состояния, а в случае установки Damage report (Step) 25 – объект будет выдавать сообщение в

миссию, каждый раз, когда им будет получена «порция» повреждений в 25% от исходного состояния. Под исходным состоянием, нужно понимать значение в 100% (полная жизнь).

Поле AI Return to Base Decision – если установить эту опцию, то если в течении миссии AI получит критические повреждения, или у него кончится боезапас, то он в автоматическом режиме вернет самолет на ближайший дружественный аэродром.

Поле Script – путь размещения файла скрипта объекта.

Поле Skin – выбор схемы расцветки объекта.

Поле Payload – конфигурация вооружения самолета (боезапас, бомбы).

Поле Route Time (minutes) – расчет времени радиуса боевого действия для AI (пока не используется).

Поле On Messages Table – меню Сообщений которые заданы объекту.

Кнопка Add by Dialog (в случае лидера группы) или Set by dialog (для ведомого) – позволяют линковать ведомых к ведущему или линковать ведомого к ведущему через выбор в диалоговом окне, то есть сформировать группу при помощи диалогового окна.

Name – пока не используются.

Description – пока не используются.

Поле Target ID – в случае ведомого - идентификатор ведущего (его ID) у данного ведомого. У ведущего это значение всегда (-1).

Пример текстового формата самолета игрока в миссии с появлением на земле (Plane + MCU_TR_Entity):

Plane //тип объекта - самолет

```
{
  Name = "Plane"; //имя самолета, будет отображаться на иконке в мультиплеере, если самолет
управляется AI
  Index = 34; //ID самолета
  LinkTrId = 35; //ID Entity
  XPos = 2170.13; //координата самолета по оси X
  YPos = 116.53; // координата самолета по оси Y, в данном случае этот параметр будет равен
уровню поверхности на которой установлен самолет, если это автомобиль или артиллерия, то
данное значение ВСЕГДА будет равно уровню земли, на котором установлен объект;
  ZPos = 4263.06; //координата самолета по оси Z
  XOri = 0; //положение самолета (угол крена) по оси X,
  YOri = 0; //положение самолета (угол поворота ) по оси Y
  ZOri = 12.8; //положение самолета (угол тангажа) по оси Z
  Script = "LuaScripts\WorldObjects\se5a.txt"; //путь к скрипту самолета
  Model = "graphics\planes\se5a\se5a.mgm"; //путь к 3D самолета
  Country = 101; //индекс страны присвоенной самолету
  Desc = ""; //описание самолета
  Skin = ""; //путь к скину самолета
```

```

AILevel = 0; //уровень AI самолета, в данном случае 0=Player (1=Low, 2=Normal, 3=High, 4=Ace)
CoopStart = 0; //доступность самолета для кооперативной миссии, в данном случае самолет (слот
для занятия самолета игроком) будет не доступен
NumberInFormation = 0; //положение в группе при построении
Vulnerable = 1; //уязвимость (1=да, 0=нет)
Engageable = 1; //взаимодействие со стороны других AI (если =0, то другие AI в миссии не будут
атаковать данный самолет)
LimitAmmo = 1; //лимитированный боезапас (да=1, нет=0)
StartInAir = 0; //старт в воздухе да/нет, если поставить=1 то обязательно нужно указывать YPos
для самолета, иначе он появится на земле
Time = 60; //лимит на полетное время (для использования AI)
DamageReport = 50; //порог (или уровень) повреждений для использования On Damage Message
(если он есть)
DamageThreshold = 1; //отчет о повреждениях, в данном случае от полного значения (100%), если
=0, то отчет будет подаваться «порционно» (step)
}

```

```

MCU_TR_Entity //Entity самолета, если её не поставить (create linked entity) то самолет НЕ
появится в миссии
{
    Index = 35; //ID entity
    Name = "Plane entity"; //имя entity (фактически в миссии не используется)
    Desc = ""; // описание entity
    Targets = []; //объект на который ссылается Entity целевой связью
    Objects = []; //объект на который ссылается Entity объектной связью
    XPos = 2170.13; //совпадает с положением объекта которому присвоена, в данном случае
самолету
    YPos = 116.73; //совпадает с положением объекта которому присвоена, в данном случае самолету
    ZPos = 4263.06; //совпадает с положением объекта которому присвоена, в данном случае
самолету
    XOri = 0; //не используется
    YOri = 0; //не используется
    ZOri = 0; //не используется
    Enabled = 1; //начальное состояние объекта (самолета), если=1, то появится сразу после старта
миссии, если=0, то его нужно будет активировать
    MisObjID = 34; //ID объекта которому принадлежит Entity
    OnMessages //список сообщений которые будет выполнять самолет
    {
        OnMsg – одно из назначенных сообщений для самолета
        {
            Type = []; - тип назначенного сообщения
            CmdId = []; - ID команды (source) после выполнения которой следует дальнейшее действие
(target) (если есть)
            TarId = []; - ID команды (target) которую выполнит самолет
        }
    }
}

```

7.4. Mission Command Units (MCUs) редактора.

7.4.1. Basic Properties для MCUs.

MCU Basic Properties

Name: Trigger Waypoint >

Position

X: 11419.86

Y: 252.41

Z: 13645.73

Orientation

AX: 0

AY: 0

AZ: 0

Targets: Find target

id	Name	Description
13	Trigger Waypoint	

Objects: Find object

id	Name	Description
----	------	-------------

Description: ID: 12

У MCU, по сравнению с объектами, в Basic Properties отсутствуют элементы Model и Create Linked Entity/Delete Linked Entity, но добавлены списки Targets и Objects, отображающие список имен и ID объектов, с которыми MCU имеет целевые или объектные связи.

Кнопки Find target и Find object, позволяют указать MCU для установки целевой связи, и объект для объектной связи соответственно, то есть, фактически соответствуют функционалу горячих клавиш SHIFT+T или SHIFT+O.

Двойное нажатие на полях списков объектных и целевых связей позволяет добавить связи через прямое указание ID MCU. Так же через Двойное нажатие на связях в списках можно выполнить их редактирование.

Для некоторых типов MCU важна их установка в пространстве (CheckZone, AttackArea), об этих и других свойствах MCU будет разъяснено в следующем пункте руководства.

7.5. Advanced Properties для MCUs.

Рассмотрим подробно, что делает каждая команда, и какие настройки для них существуют.

Для начала рассмотрим настройку приоритета команд - Priority.

Priority (приоритет выполнения) Low/Medium/High - указывает на приоритетность выполнения команды для AI объекта. Чем выше приоритет, тем выше сосредоточенность AI на выполнение этой команды, вплоть до игнорирования окружающей обстановки.

Важно помнить для того, чтобы команда сработала (была принята к исполнению объектом), её нужно активировать, т.е. «дернуть» целевой связью или сообщением.

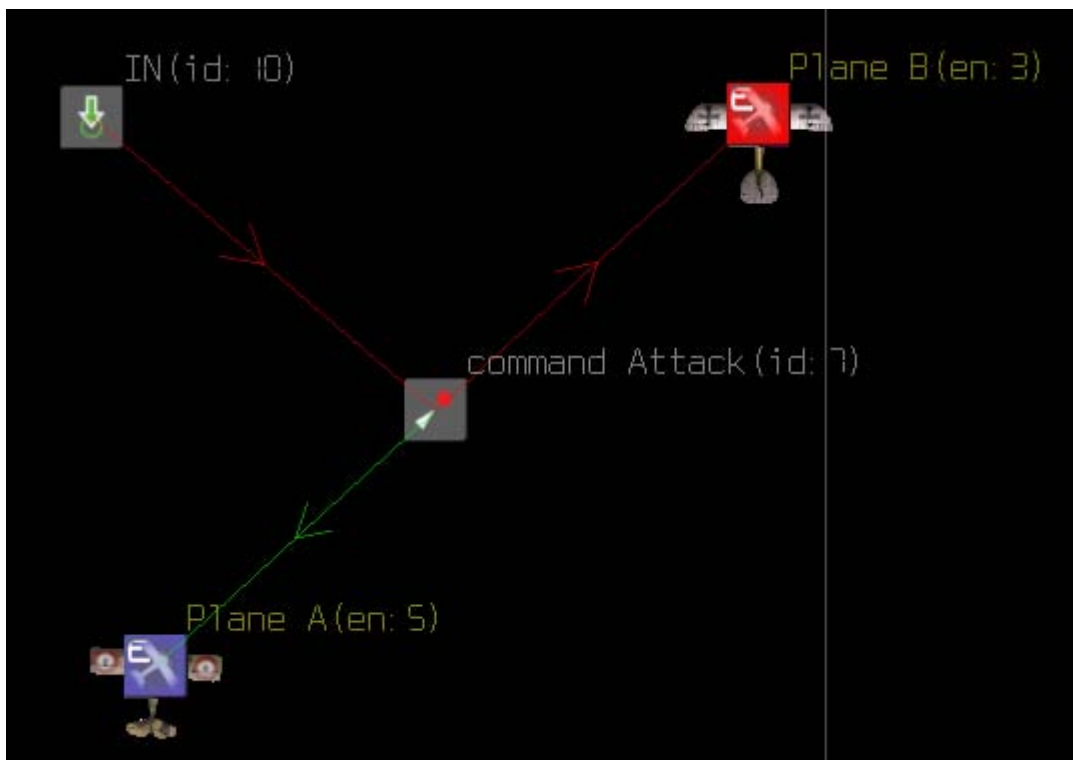
Момент срабатывания MCU должен быть определен логикой, заложенной создателем миссии, в соответствии со сценарием миссии.

7.6. MCU Command

7.6.1. Command:Attack



Эта команда говорит кому (ОС) и кого (ЦС) нужно атаковать. Особенностью данной команды является то, что ЦС от нее ставится не к команде или триггеру, а к атакуемому объекту. Объект, получивший эту команду, будет атаковать объект-цель до его разрушения.

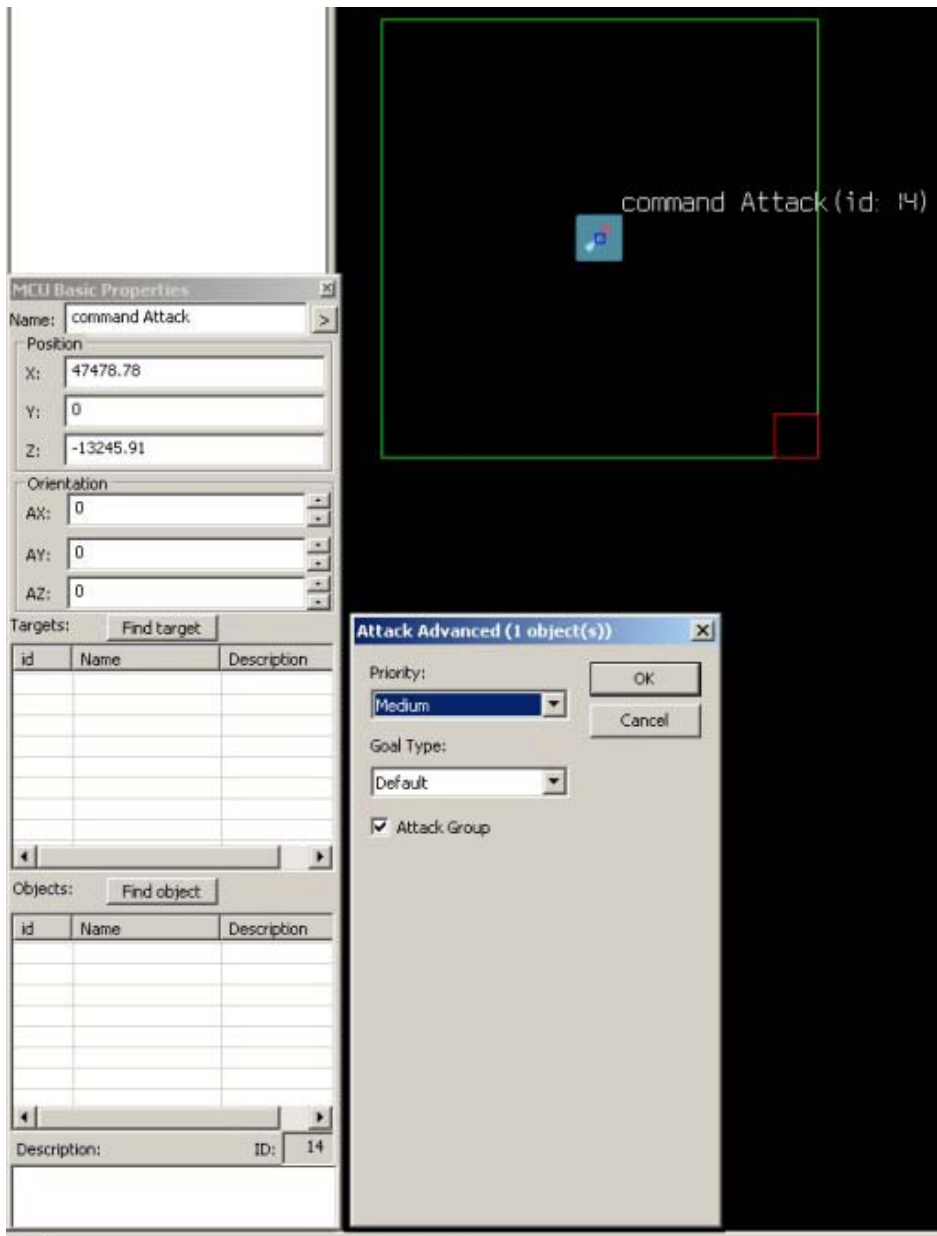


В примере на картинке – команда на атаку одного самолета другим самолетом, где:

- Plane A – атакующий, команда Attack объектной связью ссылается на этот самолет;
- Plane B – атакуемый самолет, на него целевой связью ссылается команда Attack;

- IN – любое исходящее из миссии событие, запускающее команду Attack.

Даблклик по иконке команды открывает меню Attack Advanced Properties.



Advanced Properties команды:

Priority – приоритет выполнения команды:

Поле Goal Type: Default, Primary, Secondary. Визуальное отображение иконки команды в игре, присвоенной какому-либо объекту. Сделано для удобства визуального распознавания объекта атаки и никак не влияет на результат миссии.

Default – иконка выглядит



Primary – иконка выглядит



Secondary – иконка выглядит



Чекбокс Attack Group – при включении этой опции атакующий будет атаковать всю группу, в которую входит объект на который ссылается ЦС, а не только объект, на который ему непосредственно указывает ЦС.

Приоритеты команды в зависимости от уровня AI:

При разном уровне AI получившего данную команду, атака будет выполнена если между целью и атакующим есть определенное расстояние (порог срабатывания команды по расстоянию) при выставленном приоритете Medium:

- AI Low – 4км
- AI Normal – 5км
- AI High – 6км
- AI Ace – 7км

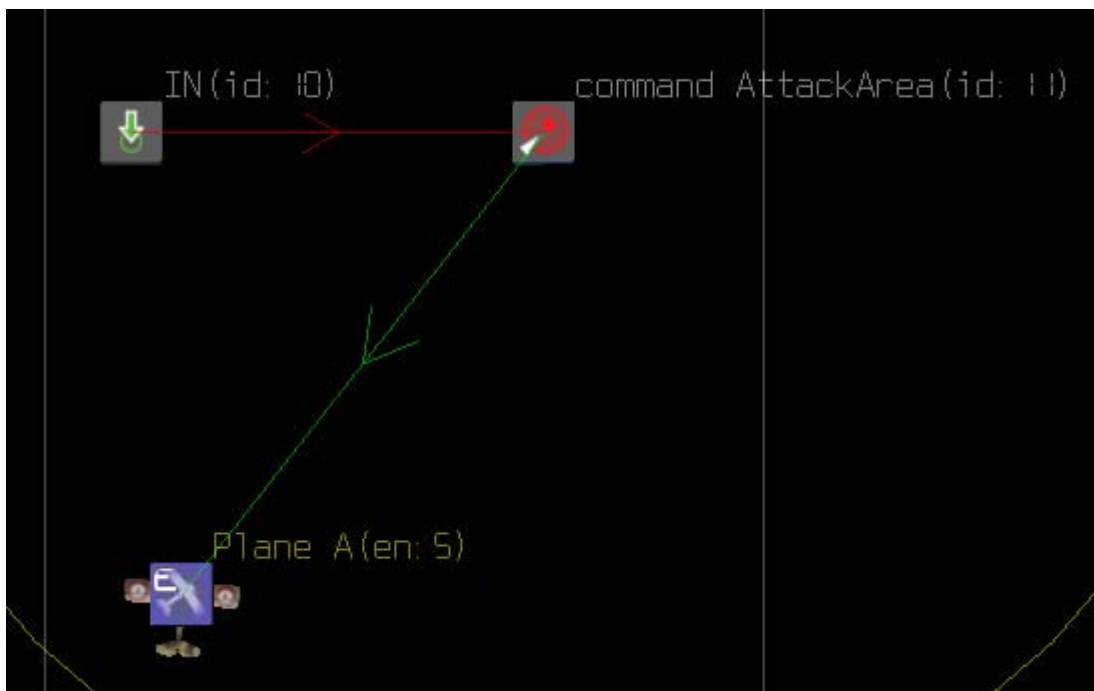
Текстовый формат команды:

```
MCU_CMD_AttackTarget //тип MCU
{
  Index = 5; //ID MCU
  Name = "command Attack"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2283.32; // координата по оси X
  YPos = 119.646; // координата по оси Y
  ZPos = 4280.93; // координата по оси Z
  XOri = 0;
  YOri = 0;
  ZOri = 0;
  AttackGroup = 1; //атаковать группу (если целевая связь ссылается на ведущего группы)
  GoalType = 0; //тип задачи, влияет на вид иконки в GUI
  Priority = 1; //значение приоритета выполнения команды
}
```

7.6.2. Command:AttackArea



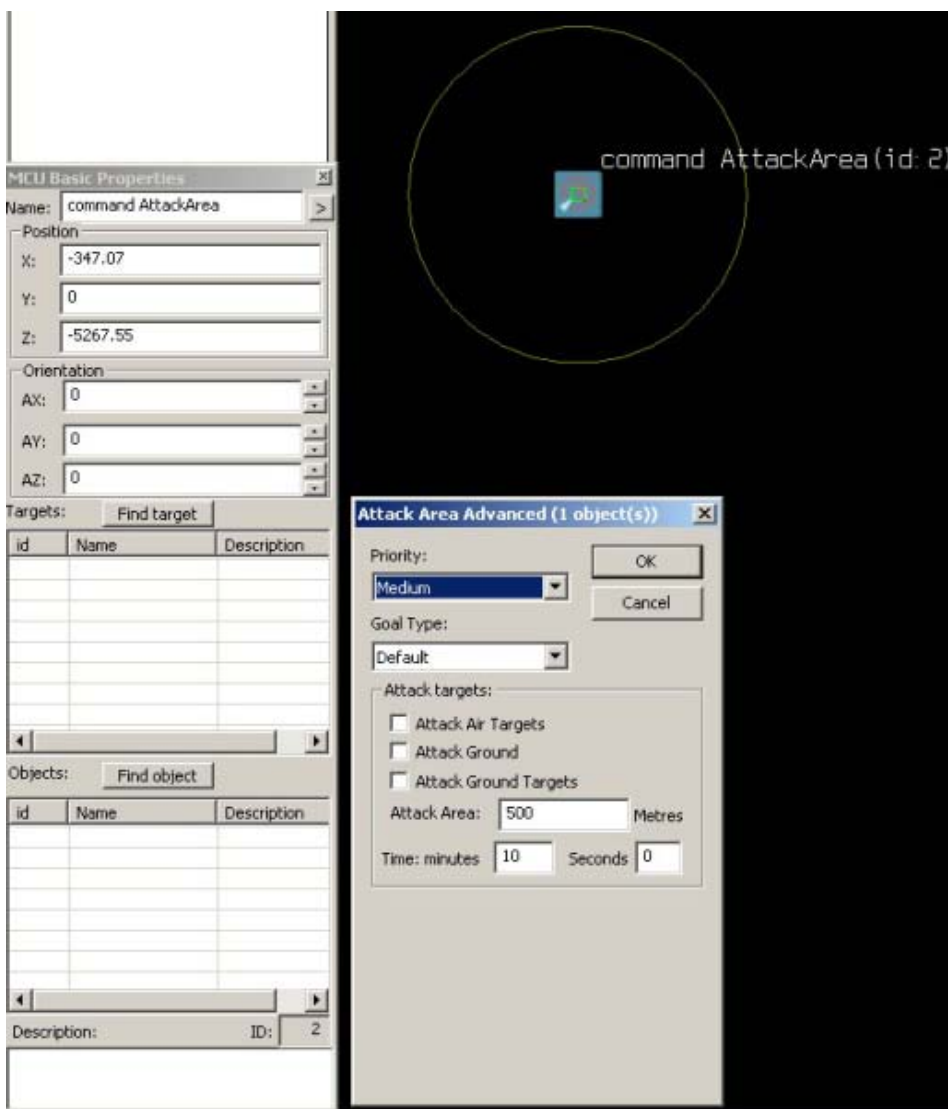
Эта команда указывает объекту область, в которой он должен атаковать вражеские объекты.



В примере на картинке – команда на атаку в зоне, где:

- Plane A – атакующий, команда AttackArea объектной связью ссылается на этот самолет;
- IN – любое исходящее из миссии событие, запускающее команду AttackArea.

Даблклик по иконке команды открывает меню Attack Area Advanced Properties.



Advanced Properties команды:

Priority – приоритет выполнения команды:

Поле Goal Type: Default, Primary, Secondary. Визуальное отображение иконки команды в игре, присвоенной какому-либо объекту. Сделано для удобства визуального распознавания зоны атаки и никак не влияет на результат миссии.

Default – иконка выглядит



Primary – иконка выглядит



Secondary – иконка выглядит



Чекбоксы Attack Air Targets, Attack Ground и Attack Ground Targets – при включении этих чекбоксов объект будет атаковать воздушные цели, область на земле и наземные цели, соответственно.

Существует тонкость с использованием опции Attack Ground – в зависимости от типа орудий объекта, эффект от команды может быть разным:

- пулеметы и зенитные орудия будут атаковать случайные точки в сфере с радиусом заданным в поле Attack Area, с центром в точке нахождения метки команды. Этот момент надо помнить, чтобы не вышло так, что объекты атакуют точки под землей;
- полевые пушки и гаубицы атакуют случайную точку на поверхности в области с радиусом указанным в поле Attack Area и центром в точке ортогональной проекции метки команды на поверхность;
- самолеты сбрасывают весь запас бомб в точку ортогональной проекции метки команды на поверхность, то есть, если есть задача сбросить бомбы с горизонтального полета на высоте, то нужно выставить данную команду на высоте сброса бомб с центром над целью.

Поле Attack Area – задает радиус атакуемой области в метрах.

Поля группы Time: minutes и Seconds – задают время действия команды в минутах и секундах, соответственно.

Текстовый формат команды:

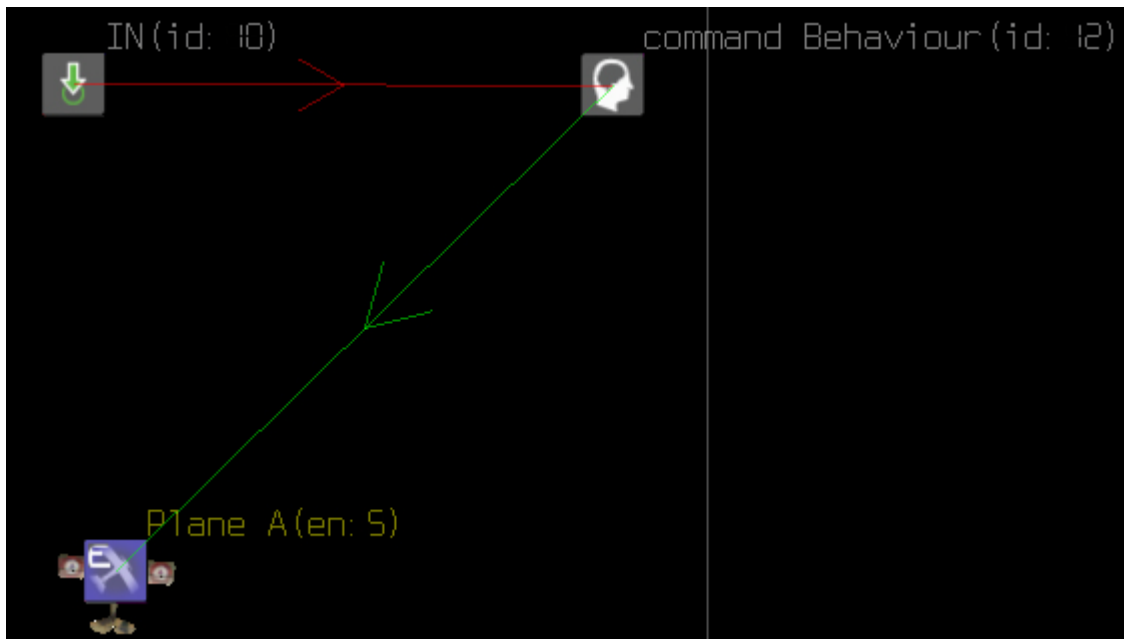
MCU_CMD_AttackArea

```
{
  Index = 4; //ID MCU
  Name = "command AttackArea"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2074.81; // координата по оси X
  YPos = 113.7; // координата по оси Y, это высота положения Attack Area, не забывайте выставлять
высоту т.к. AI учитывает её значение
  ZPos = 4018.81; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  AttackGround = 0; // тип Attack Area (Attack Ground да=1/нет=0)
  AttackAir = 0; // тип Attack Area (Attack Air да=1/нет=0)
  AttackGTargets = 0; //тип Attack Area (Attack Ground Targets да=1/нет=0)
  AttackArea = 1000; //радиус в метрах
  Time = 600; //время выполнения команды в секундах
  GoalType = 0; //тип задачи, влияет на вид иконки в GUI
  Priority = 1; //значение приоритета выполнения команды
}
```

7.6.3. Command:Behavior



Позволяет динамически изменять состояние свойств уязвимости, расходуемости боезапаса, возможности быть атакованным другими AI и изменение уровня AI.

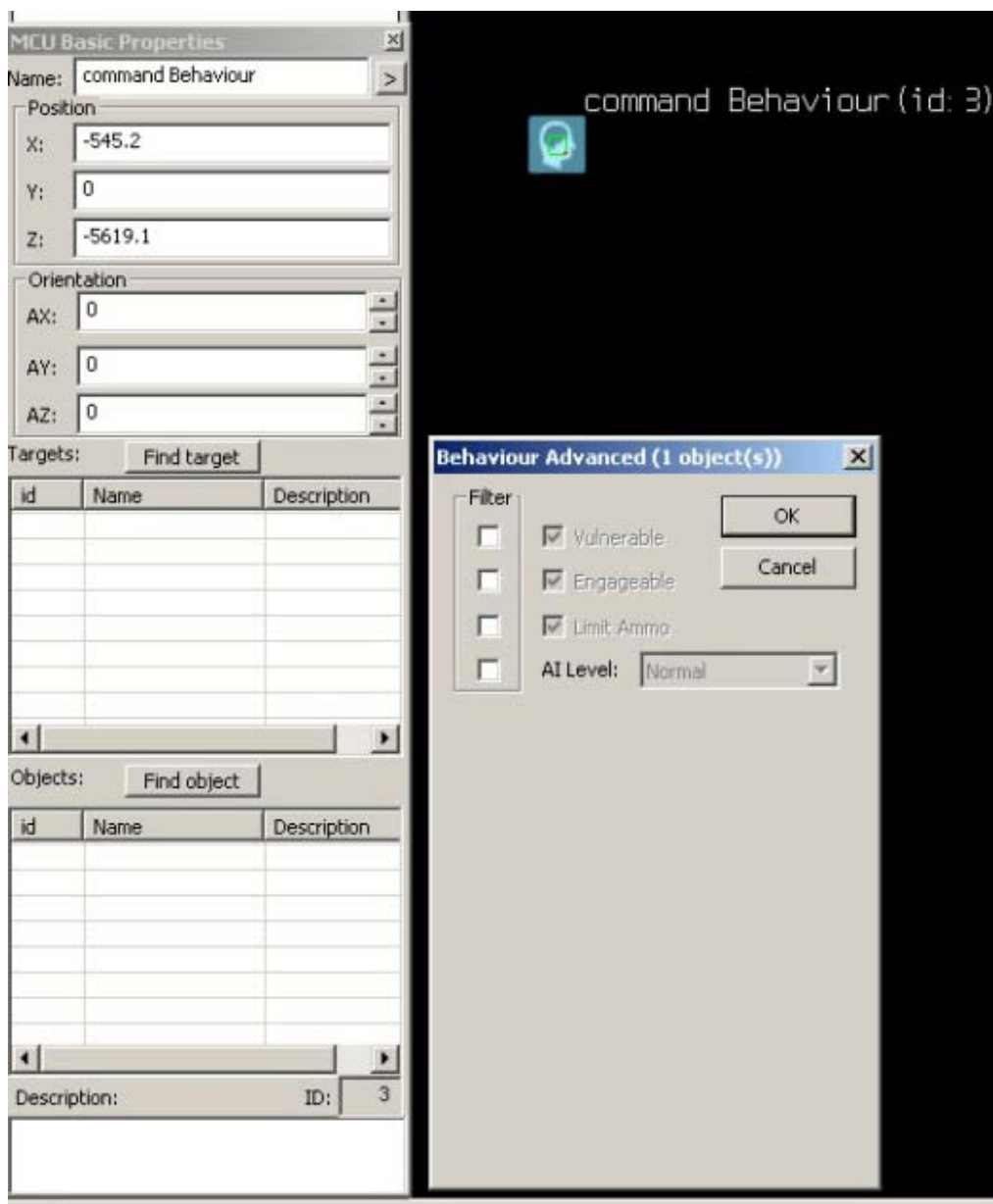


В приведенном примере:

- при активации команды Behavior исходящим событием из миссии (целевая связь от IN), Plane A изменит свои свойства в зависимости от установок внутри команды Behavior.

Команда может быть применена к любому объекту миссии (кроме зданий и мостов), в зависимости от наличия у него указанных динамически изменяемых свойств.

Линкуется к каждому объекту через ОС. Даблклик по иконке команды открывает меню Behavior Advanced Properties.



Чекбокс Filter – при включении этих чекбоксов свойства объекта будут настраиваемы.

Чекбокс Vulnerable – уязвимость объекта от повреждений или нет;

Чекбокс Engageable – будет объект атакован AI или нет;

Чекбокс Limit Ammo – бесконечный боезапас или нет;

Поле AI Level – настройка уровня мастерства объекта.

Player – нельзя назначить, НЕ используется;

Уровень AI можно задать любым самолетам (или объектам) которыми НЕ управляет игрок.

Low – низкий уровень мастерства;

Normal – средний уровень мастерства;

High – высокий уровень мастерства;

Ace – уровень мастерства ас.

Текстовый формат команды:

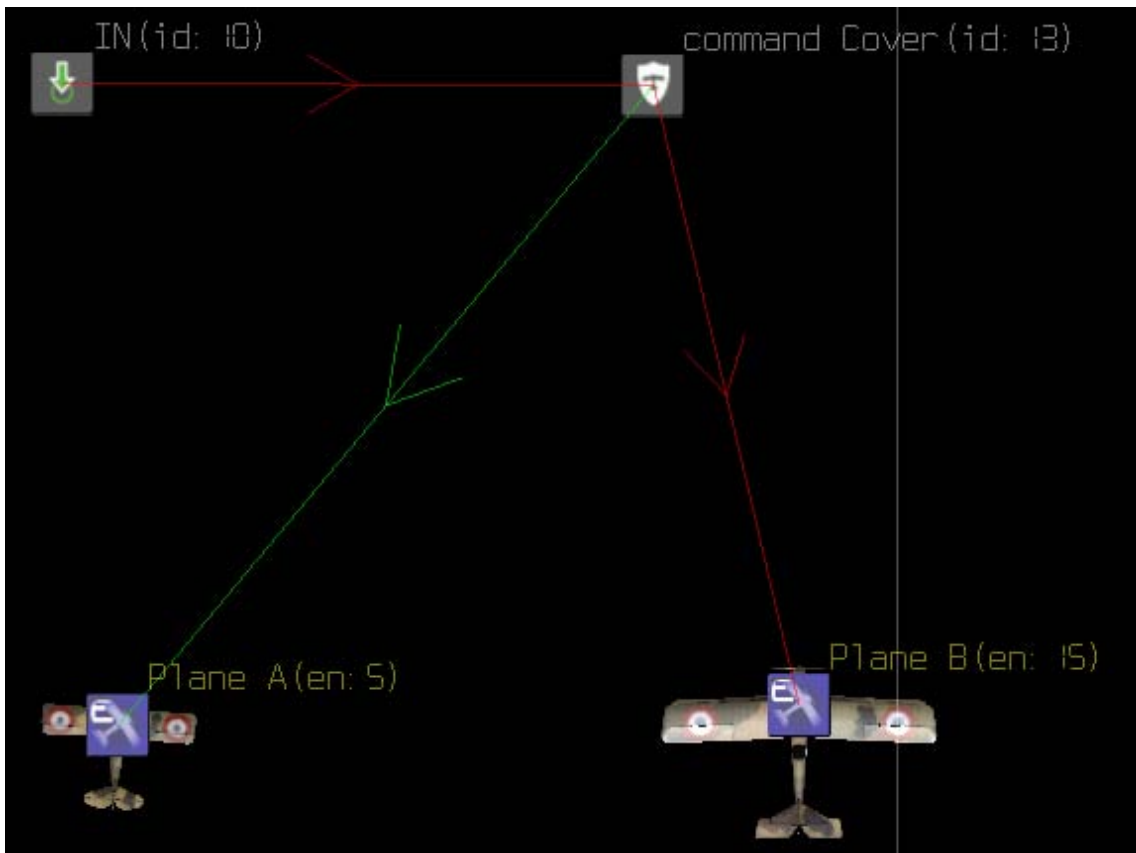
MCU_CMD_Behaviour //тип MCU

```
{
  Index = 6; //ID MCU
  Name = "command Behaviour"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2024.18; // координата по оси X
  YPos = 110.925; // координата по оси Y
  ZPos = 4164.77; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Filter = 0; //значение фильтра задаваемых свойств
  Vulnerable = 1; //установка объекту уязвимости
  Engageable = 1; //установка объекту воздействия со стороны других AI
  LimitAmmo = 1; //установка объекту лимита боезапаса
  AILevel = 2; //установка объекту уровня AI
}
```

7.6.4. Command:Cover

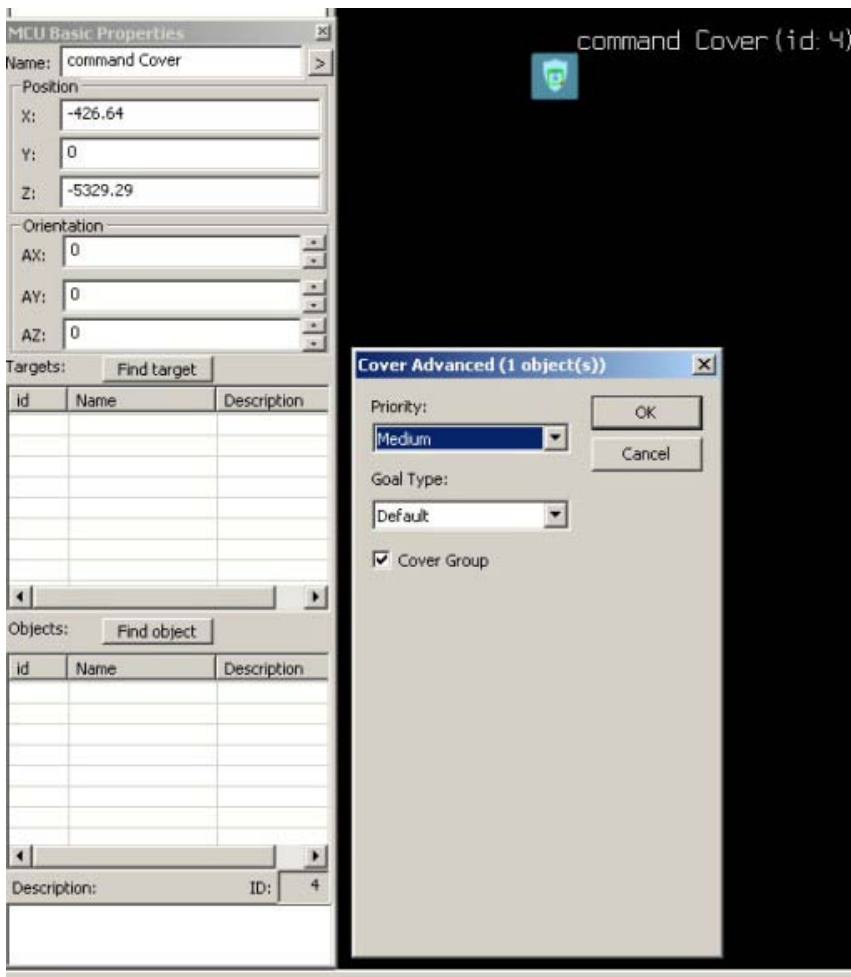


Команда на прикрытие (защиту) объектов или группы объектов, может использоваться ТОЛЬКО для самолетов.



На приведенном примере:

- Plane A получает команду Cover на прикрытие Plane B;
- команда Cover объектной связью ссылается на исполняющий её объект (Plane A), а целевой связью указывает на объект прикрытия;
- IN – любое исходящее из миссии событие, запускающее команду Cover.



Линкуется к лидеру группы защитников через ОС. Даблклик по иконке команды открывает меню Cover Advanced Properties.

Advanced Properties команды:

Priority – приоритет выполнения команды:

Поле Goal Type: Default, Primary, Secondary. Визуальное отображение иконки команды в игре, присвоенной какому-либо объекту. Сделано для удобства визуального распознавания объекта прикрытия.

Default – иконка выглядит



Primary – иконка выглядит



Secondary – иконка выглядит



Чекбокс Cover Group: при включении команда распространяется на прикрытие всей группы объектов. Если не включено, то команда распространяется только на объект, на который указывает целевая связь.

Текстовый формат команды:

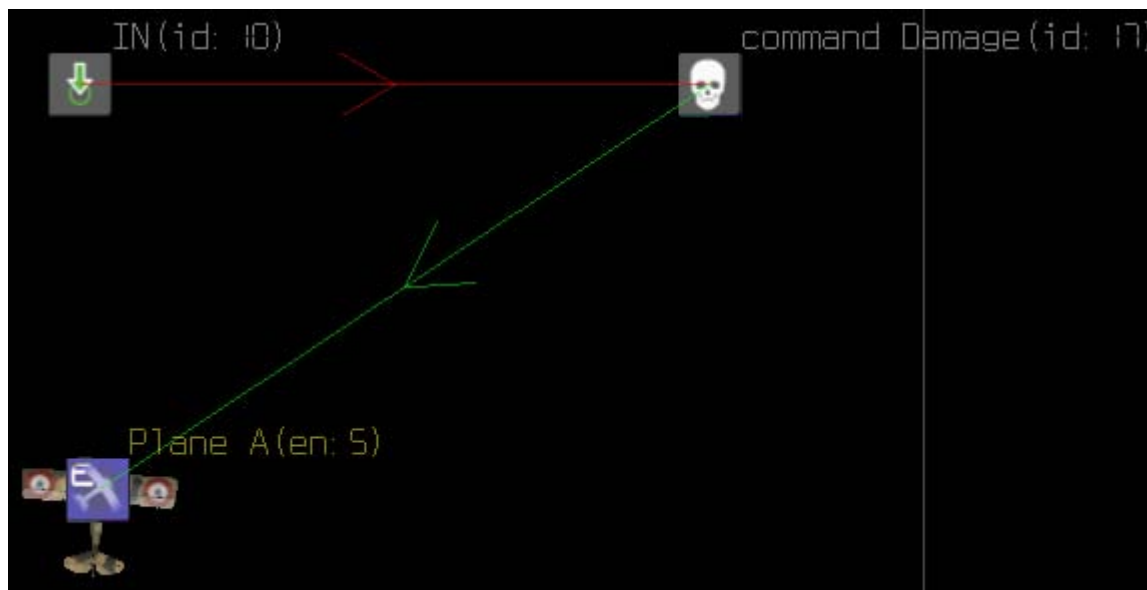
MCU_CMD_Cover //тип MCU

```
{
  Index = 7; //ID MCU
  Name = "command Cover"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2077.79; // координата по оси X
  YPos = 110.272; // координата по оси Y
  ZPos = 4355.4; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  CoverGroup = 1; //прикрытие группы да=1/нет=0
  GoalType = 0; //тип задачи, влияет на вид иконки в GUI
  Priority = 1; //значение приоритета команды
}
```

7.6.5. Command:Damage



Команда задает уровень повреждений наносимый объекту по команде из миссии.



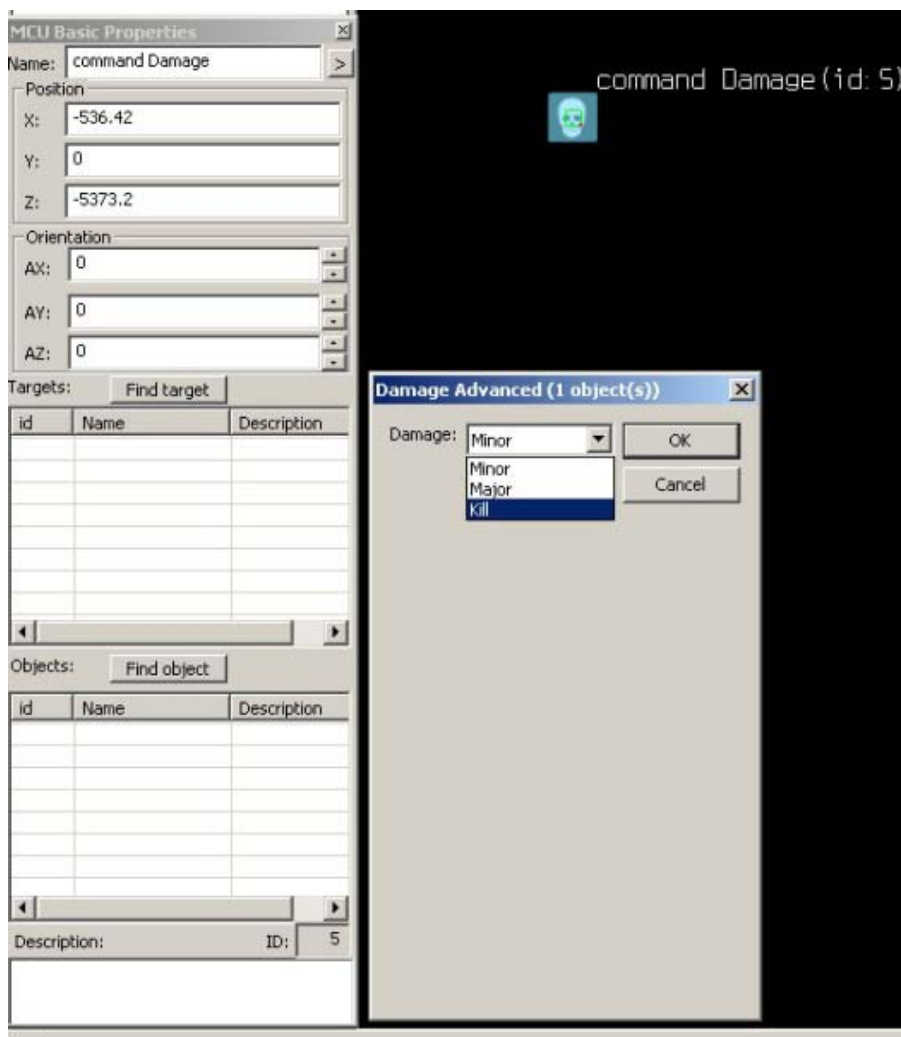
На приведенном примере:

- command Damage – через объектную связь прилинкована к Plane A;
- IN – любое исходящее из миссии событие, запускающее команду Damage, которая в свою очередь установит подчиненному объекту заданные свойства (Plane A).

Линкуется к объекту через ОС.

Если группа объектов, то к лидеру – применяется ко всем объектам в группе

Даблклик по иконке команды открывает меню Damage Advanced Properties.



Advanced Properties команды:

Поле Damage – настройка уровня повреждений объекта.

Minor – задает низкий уровень повреждений;

Major – задает высокий уровень повреждений;

Kill – задает уничтожение объекта.

Текстовый формат команды:

MCU_CMD_Damage //тип MCU

{

Index = 8; //ID MCU

Name = "command Damage"; //имя MCU (не отображается в GUI)

Desc = ""; //описание MCU (не отображается в GUI)

Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной команды исходящих целевых связей быть не может

Objects = []; //ID объектов, на которые объектной связью ссылается данная команда

XPos = 2107.58; // координата по оси X

```

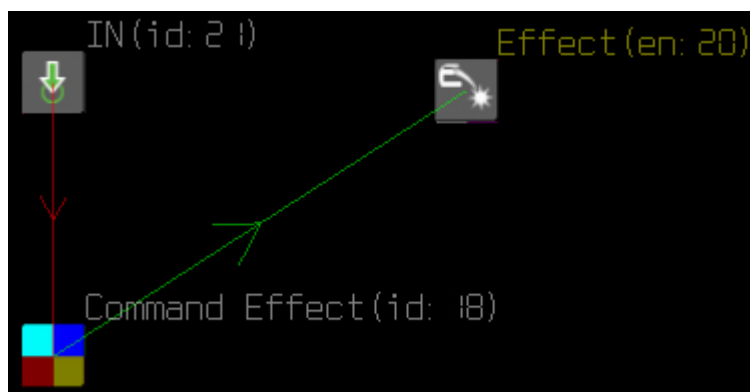
YPos = 112.755; // координата по оси Y
ZPos = 4272; // координата по оси Z
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)    Damage = 0; //
уровень повреждений задаваемый объекту 0=Minor, 1=Major, 2=Kill
}

```

7.6.6. Command:Effect



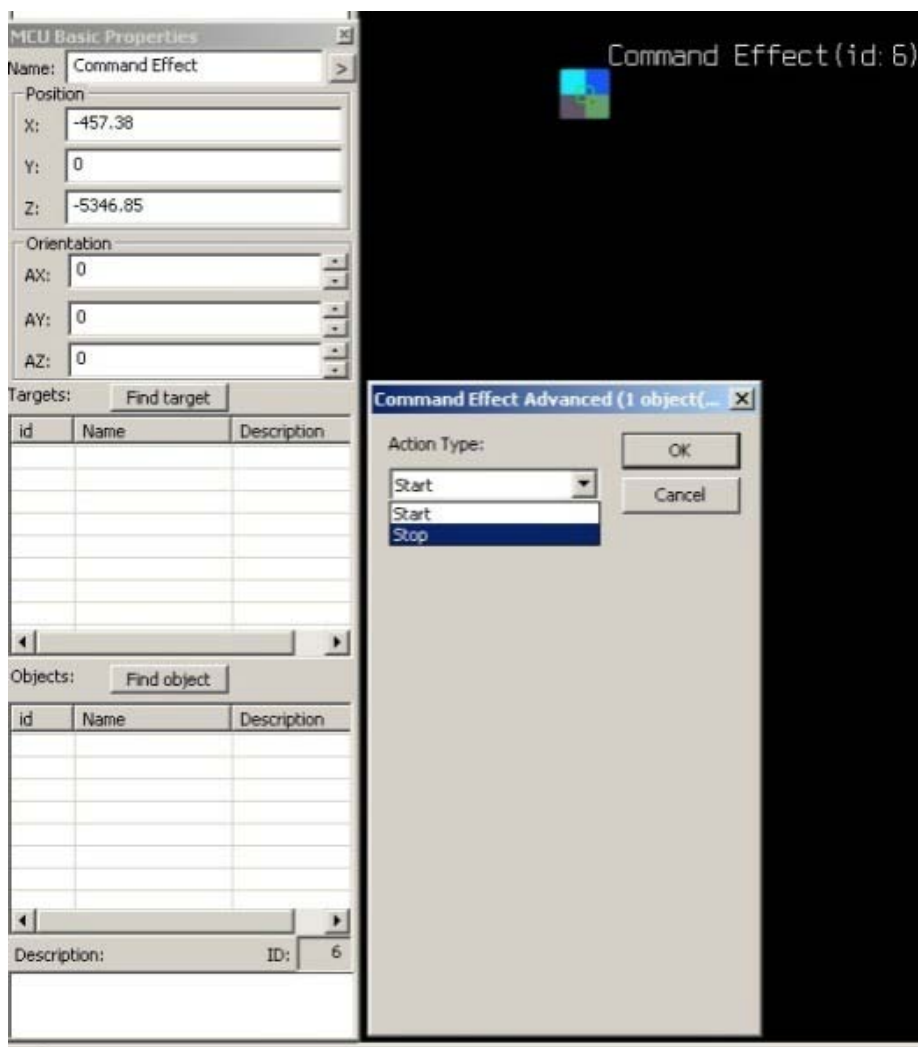
Команда позволяет запустить или остановить заранее выставленный на карту эффект.



На приведенном примере:

- IN любое исходящее из миссии событие, запускающее команду Effect;
- команда Effect объектной связью ссылается на объект Effect (см. далее) и, в зависимости от внутренних настроек, либо запускает его (start), либо останавливает (stop).

Даблклик по иконке команды открывает меню Effect Advanced Properties.



Advanced Properties команды:

Поле Action Type – выбор начала или окончания действия эффекта.

Start – начало действия эффекта;

Stop – окончание действия эффекта.

Примечание: данная команда работает только в связке с разделом Effects. В этом разделе находятся доступные в ПРМ эффекты, количество и типы которых могут варьироваться в разных версиях ПРМ.

Особенности использования:

Чтобы запустить какой-либо эффект необходимо поставить его на карту (Object Library=>Effects) и «оживить» (Create Linked Entity). Далее, от команды Effect устанавливается ОС на сам эффект. В этом случае команда Effect активируется от любого исходящего из миссии события.

Далее, произойдет или остановка проигрывания эффекта или начало его проигрывания, в зависимости от назначенного действия в поле Action Type.

Текстовый формат команды:

MCU_CMD_Effect //тип MCU

```
{
  Index = 9; //ID MCU
  Name = "Command Effect"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2116.51; // координата по оси X
  YPos = 114.073; // координата по оси Y, для самой команды положение не имеет значения, но
вот для эффекта (Effect), который запускает данная команда положение нужно выставить
обязательно т.к. эффект это визуальный (или звуковой) объект в миссии
  ZPos = 4218.38; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ActionType = 0; // старт=0 или стоп=1 (запуск или стоп) эффекта
}
```

7.6.7. Command:Force Complete



Прерывает последнюю команду, которую получил объект.



На приведенном примере, нам нужно прервать команду прикрытия и дать самолету следующую команду:

- Plane A (выполняет текущую команду Cover) – прикрывающий самолет Plane B;
- command activated – событие миссии, которое запустило команду на прикрытие (или любую другую текущую команду);
- IN – входящее событие из миссии, по которому нам нужно прервать текущую команду (в данном случае прикрытие самолета Plane B), при этом следующая команда для самолета Plane A дергается при помощи проставки из таймера Timer 1s (таймер с величиной в 1 секунду), во избежание коллизии событий (чтобы случайно не прервать следующую поступившую команду).

Даблклик по иконке команды открывает меню Force Complete Advanced Properties.

Advanced Properties команды:

Priority – приоритет выполнения команды

Особенности использования:

При использовании данной команды необходимо помнить, что в случае, если после этой команды объекту назначены какие-либо другие действия (команды), то между командой Force Complete и следующей командой обязательно должен быть 1 или 2 секундный триггер Timer, иначе все последующие команды могут не запуститься (см. пример).

Текстовый формат команды:

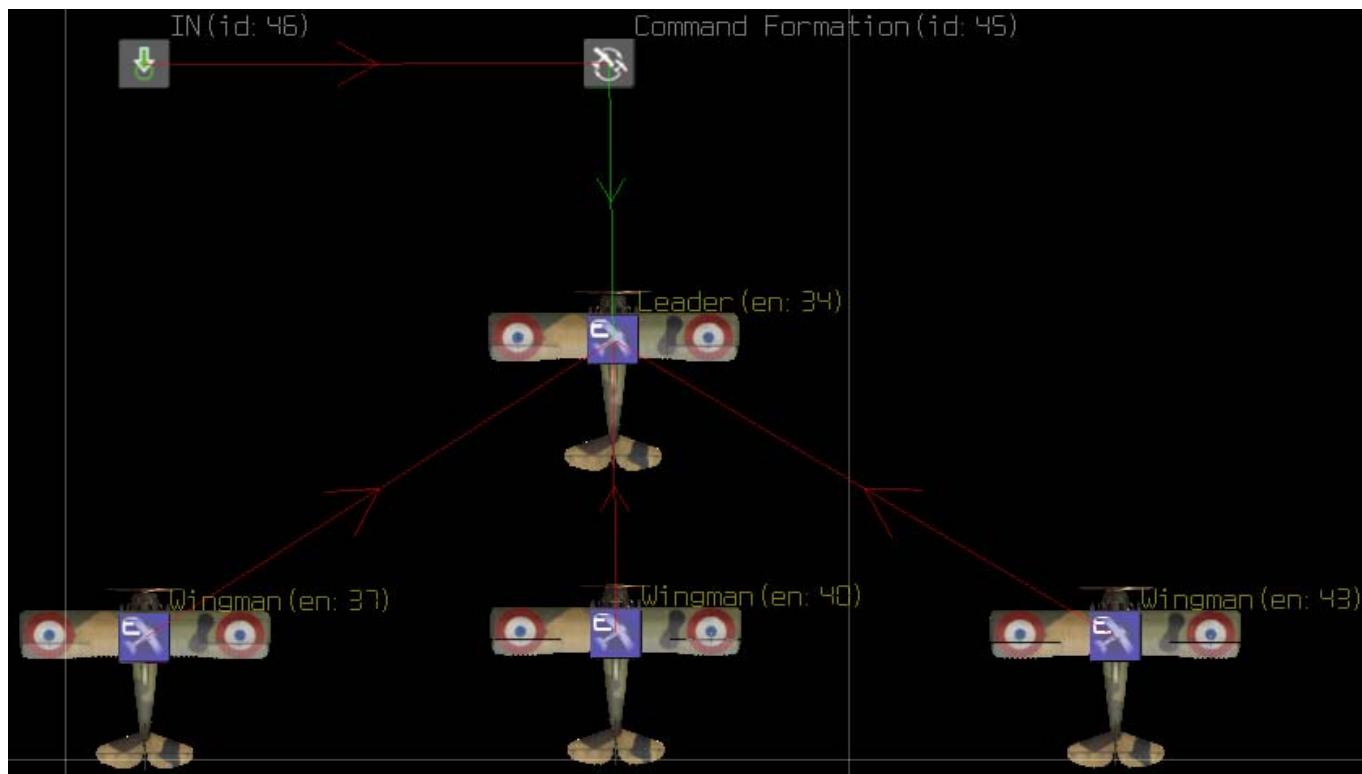
MCU_CMD_ForceComplete //тип MCU

```
{
  Index = 10; //ID MCU
  Name = "command Force Complete"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; // ID объектов, на которые целевой связью ссылается данная команда
  XPos = 2116.51; // координата по оси X
  YPos = 115.985; // координата по оси Y
  ZPos = 4039.66; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Priority = 2; //значение приоритета выполнения команды
}
```

7.6.8. Command:Formation



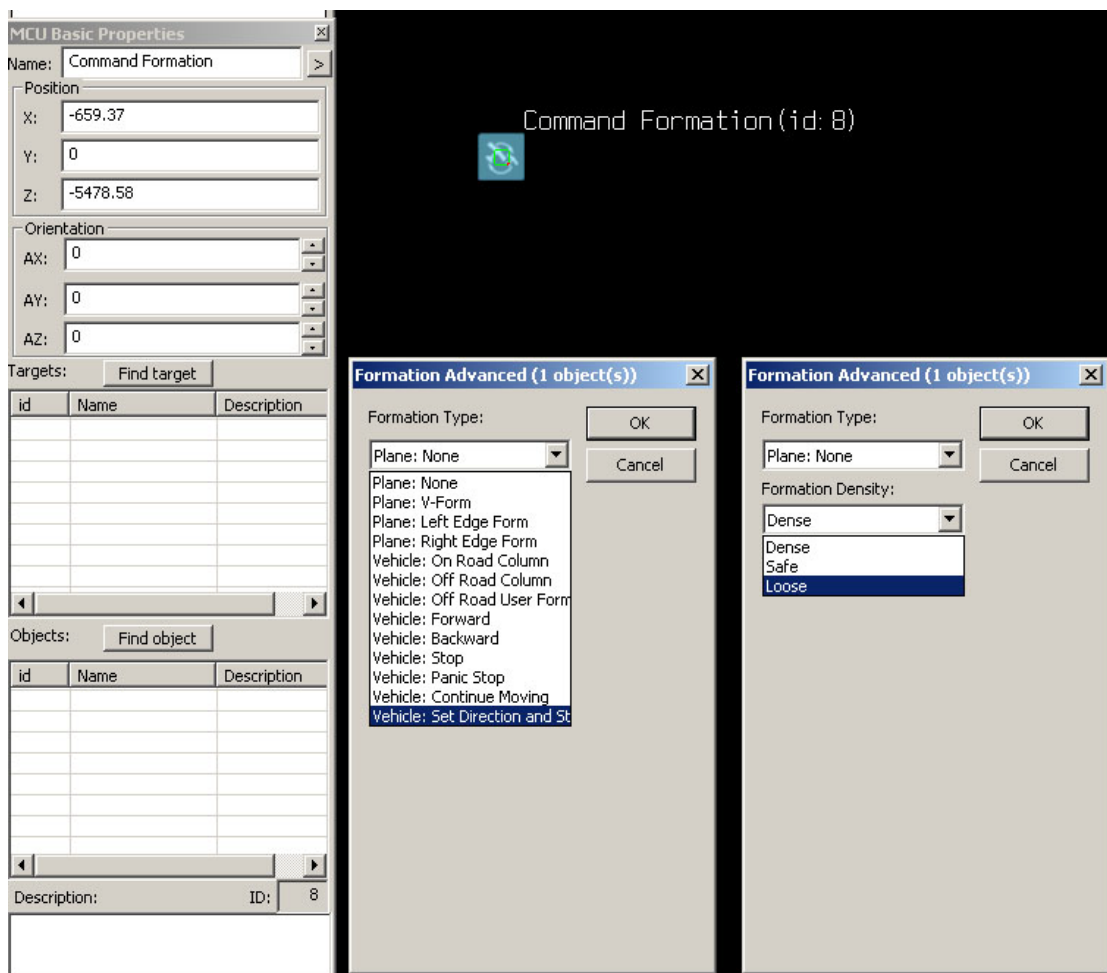
Эта команда задает строй (формацию), в котором будет двигаться группа объектов



В приведенном примере:

- Command Formation объектной связью прилинкована к ведущему группы;
- IN – любое исходящее из миссии событие, запускающее команду Formation, которая, в свою очередь, выдает ведущему заданное в команде построение объектов в группе.

Даблклик по иконке команды открывает меню Formation Advanced Properties.



Advanced Properties команды:

Список Formation Type – указывает тип построения.

Типы построений разделены на самолетные (имеют префикс Plane:) и для наземной техники (имеют префикс Vehicle).

Plane: None – самолеты: колонна.

Plane: V-Form – самолеты: V образное построение, с ведущим самолетом в вершине.

Plane: Left Edge Form – самолеты: левый пеленг.

Plane: Right Edge Form – самолеты: правый пеленг.

Vehicle: On Road Column – движение техники по дорожной сети, в колонне.

Vehicle: Off Road Column – движение техники в колонне, по прямой к точке, не учитывая рисунок дороги.

Vehicle: Off Road User Formation – движение техники в строю заданном пользователем в редакторе (тоесть так, как пользователь их расставил в редакторе), по прямой к точке.

Vehicle: Forward – движение передним ходом.

Vehicle: Backward - движение задним ходом (только для гусеничной техники).

Vehicle: Stop – остановка на маршруте.

Vehicle: Panic Stop – остановка с рассыпанием техники по местности.

Vehicle: Continue Moving – продолжение движения после остановки.

Vehicle: Set Direction and Stop – остановка с поворотом по направлению, для этого нужно задать команде YOri.

Список Formation Density – указывает дистанцию между объектами в построении от растянутой (Loose) до сжатой (Dense).

Текстовый формат команды:

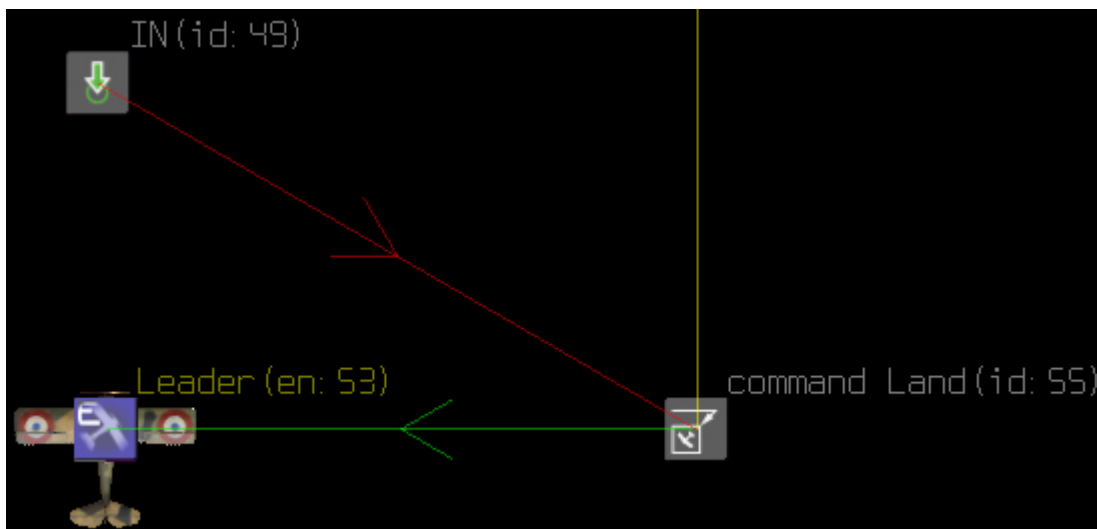
MCU_CMD_Formation //тип MCU

```
{
  Index = 11; //ID MCU
  Name = "Command Formation"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2036.09; // координата по оси X
  YPos = 111.578; // координата по оси Y
  ZPos = 4209.45; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве, используется только в случае Set Direction And Stop
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  FormationType = 0; //
тип построения 0=Plane: None, 1= Plane: V-Form, 2= Plane: Left Edge Form, 3= Plane: Right Edge
Form, 4= Vehicle: On Road Column, 5= Vehicle: Off Road Column, 6= Vehicle: Off Road User
Formation, 7= Vehicle: Forward, 8= Vehicle: Backward, 9= Vehicle: Stop, 10= Vehicle: Panic Stop, 11=
Vehicle: Continue Moving, 12= Vehicle: Set Direction and Stop
  FormationDensity = 0; //плотность построения 0=Dence, 1=Safe, 2=Loose
}
```


7.6.9. Command:Land



Команда на посадку объектам типа «самолет» в заданной точке (точке нахождения иконки команды).



В данном примере:

- IN - любое исходящее из миссии событие, запускающее командуLand.

Желтая линия указывает направление захода на посадку AI, поэтому будьте внимательны при установке команды.

Работает через объектную связь (ОС) с объектом (группой объектов → ОС на лидера группы).

При создании миссии, важно учитывать следующее:

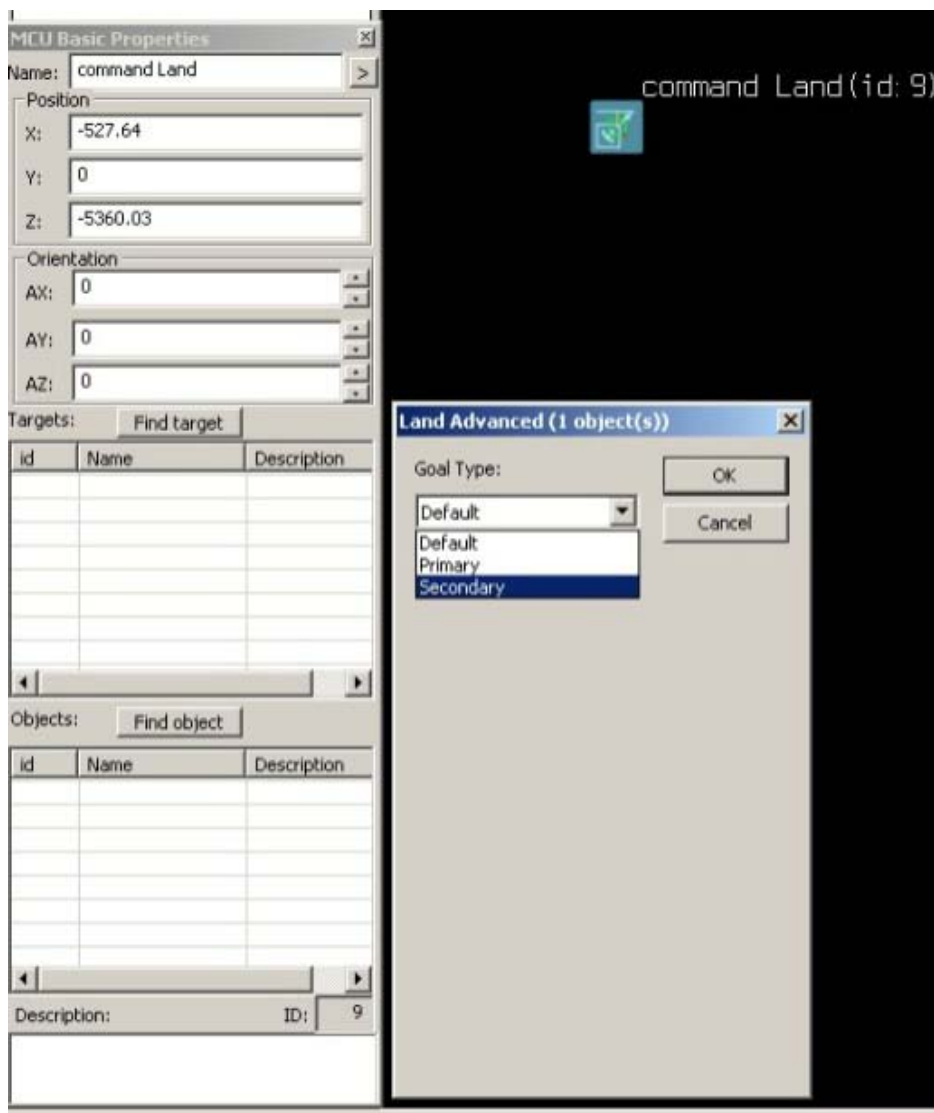
- выдавать данную команду следует только в непосредственной близости с аэродромом, потому что в случае, если AI получает команду на посадку – он становится беззащитным, и если дать данную команду далеко от аэродрома, то в случае атаки этого AI другими участниками миссии (противоположной коалиции), он не будет вести ответных действий (грубо говоря будет лететь как мишень по прямой).

Посадка самолета, может быть осуществлен в любой точке на карте, поверхность которой позволит AI произвести безопасную посадку (если площадка будет неровной или на глиссаде будут присутствовать препятствия – AI может разбиться).

В случае, если у аэродрома, на который установлена команда Land есть Entity, то AI будет осуществлять заход на посадку по заложенной в аэродром схеме.

Не критично, но - не забывайте опускать команду на уровень земли, потому как, если в миссии будет включено отображение маршрута полета, то команда посадки будет находится на уровне аэродрома.

Даблклик по иконке команды открывает меню Land Advanced Properties.



Advanced Properties команды:

Поле Goal Type: Default, Primary, Secondary. Визуальное отображение иконки команды в игре, присвоенной какому-либо объекту.

Сделано для удобства визуального распознавания зоны посадки и никак не влияет на результат миссии.

Default – иконка выглядит так:



Primary и Secondary иконки выглядят идентично.

Текстовый формат команды:

MCU_CMD_Land //тип MCU

```
{
  Index = 12; //ID MCU
  Name = "command Land"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
  Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
  XPos = 2036.09; // координата по оси X, нужно устанавливать данный параметр, т.к. AI
учитывает положение данной команды
  YPos = 111.578; // координата по оси Y нужно устанавливать данный параметр, т.к. AI
учитывает положение данной команды
  ZPos = 4283.91; // координата по оси Z нужно устанавливать данный параметр, т.к. AI учитывает
положение данной команды
  XOri = 0;
  YOri = 0; // направление захода на посадку – учитывается AI
  ZOri = 0;
  GoalType = 0; //тип задачи, влияет на вид иконки в GUI
}
```

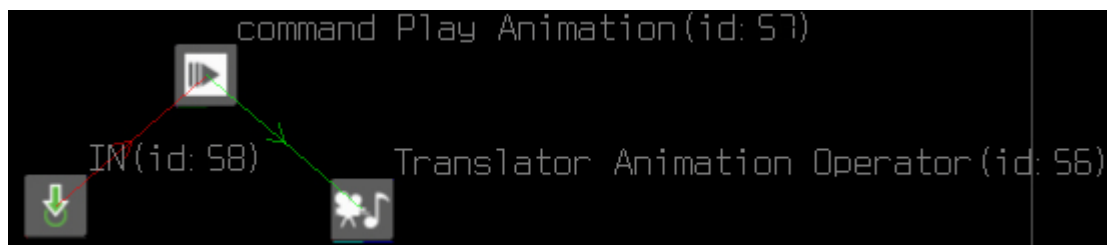
7.6.10. Command:Move

Данная команда НЕ используется.

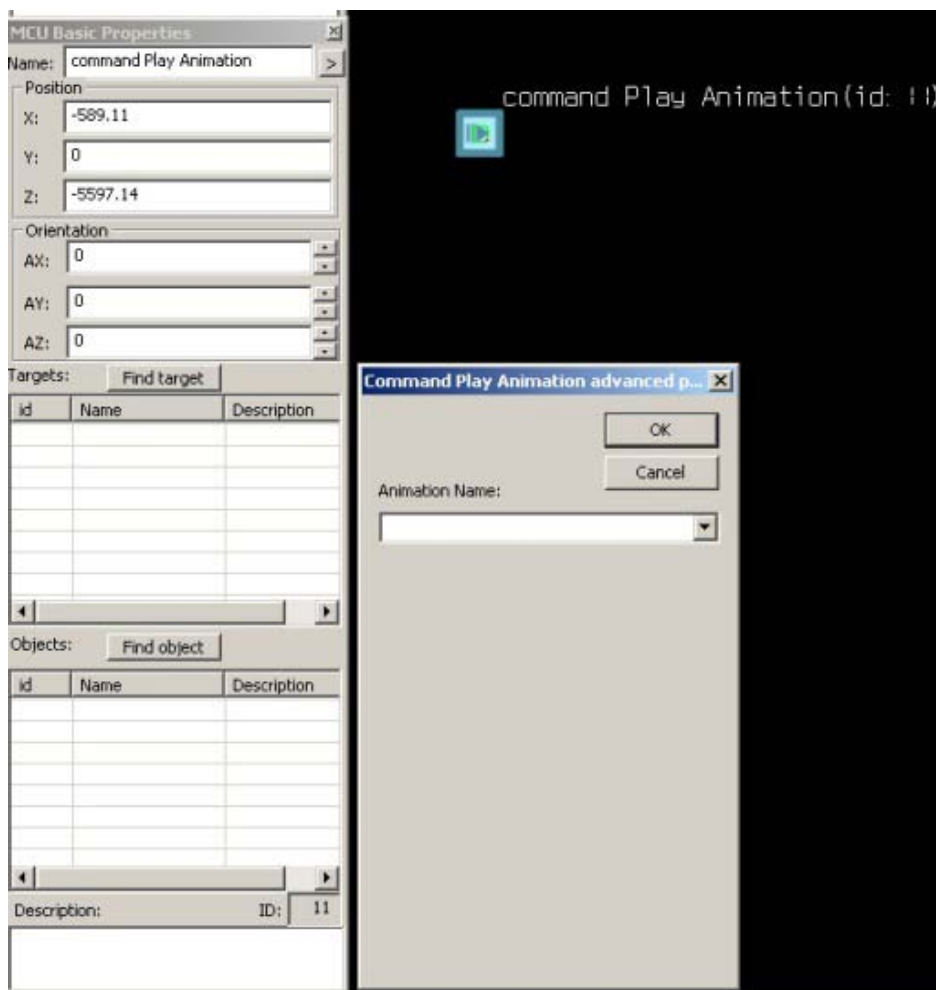
7.6.11. Command:PlayAnimation



Команда объекту Translator:Animation Operator (см.п. 7.7.1.) проиграть одну из его анимаций, с заданным именем.



Даблклик по иконке команды открывает меню Play Animation Advanced Properties.



Advanced Properties:

Поле Animation Name: - имя анимации, которую надо проиграть. Список анимаций берется из транслятора Animation Operator с которым объектной связью связана команда.

Примечание: данная команда работает только в связке с транслятором Animation Operator (см. п. 7.7.1).

Особенности использования:

Для запуска анимации нужно установить от PlayAnimation ОС на транслятор Animation Operator и выбрать в PlayAnimation тип проигрываемой анимации. Команда PlayAnimation активируется от любого исходящего из миссии события. Подробнее см.п. 9.7 и 9.8.

Текстовый формат команды:

```
MCU_CMD_PlayAnimation //тип MCU
{
  Index = 14; //ID MCU
  Name = "command Play Animation"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
```

```

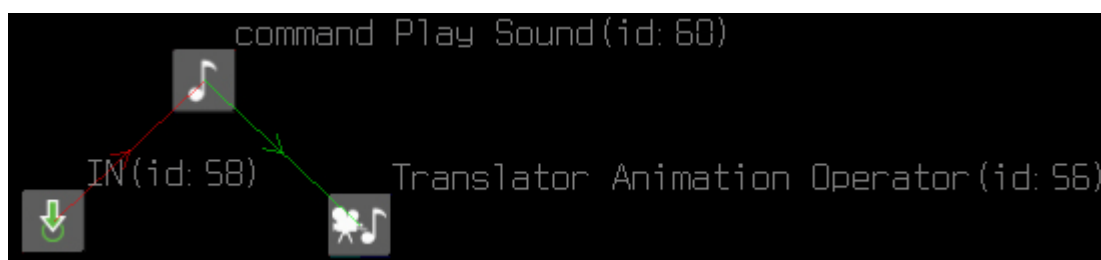
Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
XPos = 2283.32; // координата по оси X
YPos = 2107.58; // координата по оси Y
ZPos = 115.08; // координата по оси Z
ZPos = 4197.53;
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
AnimationName = ""; // выбранная для проигрывания анимация
}

```

7.6.12. Command:PlaySound



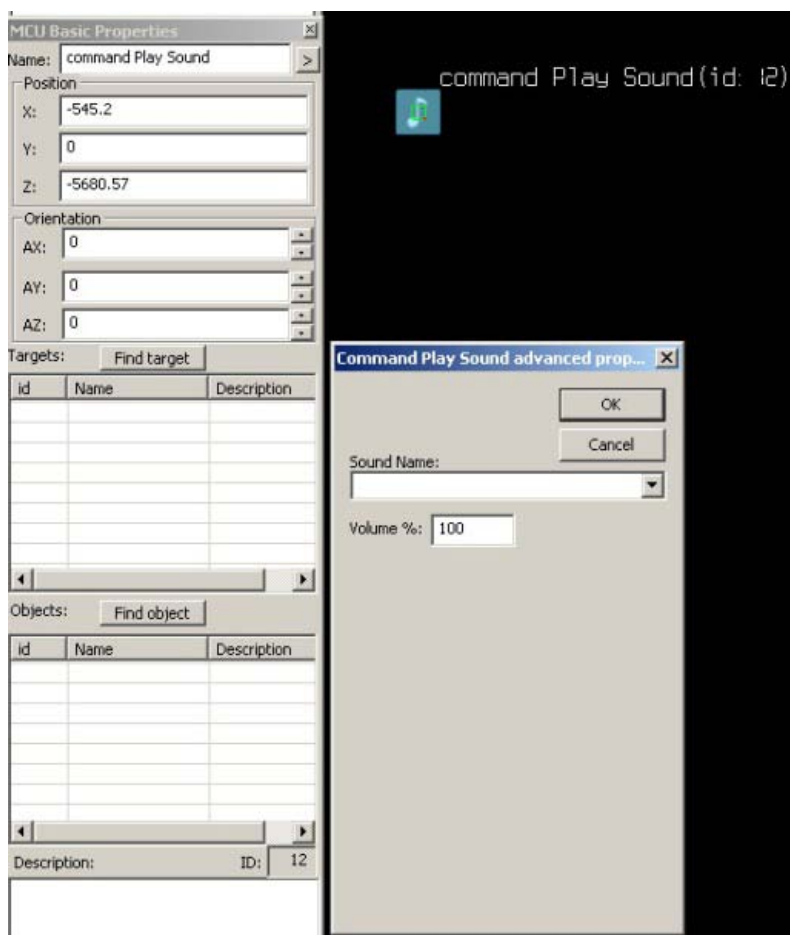
Команда объекту Translator:Animation Operator (см.п. 7.7.1.) проиграть звук.



На приведенном примере:

- Translator Animation Operator – транслятор, в свойствах которого указывается конфиг файл (схема озвучивания) / или семпл для проигрывания звука (см. п. 7.7.1.);
- IN - любое входящее событие из миссии, которое при помощи целевой связи (или Message) запускает команду Play Sound.

Даблклик по иконке команды открывает меню Play Sound Advanced Properties.



Advanced Properties:

Поле Sound Name: - имя звука, который надо проиграть. Список звуков будет соответствовать списку в Animation Operator с которым объектной связью связана команда.

Поле Volume %: - громкость проигрываемого звука в %.

Примечание: данная команда работает только в связке с транслятором Animation Operator (см. раздел 7.7.1.).

Особенности использования:

Чтобы запустить какой либо Sound необходимо поставить транслятор Animation Operator, от команды PlaySound ОС на транслятор Animation Operator, установить в команде нужный для проигрывания звук или, если звук это семпл, то просто запустить команду. Команду PlaySound можно запустить любым исходящим из миссии событием. Подробнее см. п. 9.7 и 9.8.

Текстовый формат команды:

MCU_CMD_PlaySound //тип MCU

{

Index = 15; //ID MCU

Name = "command Play Sound"; //имя MCU (не отображается в GUI)

Desc = ""; //описание MCU (не отображается в GUI)

Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной команды исходящих целевых связей быть не может

Objects = []; //ID объектов, на которые объектной связью ссылается данная команда

```

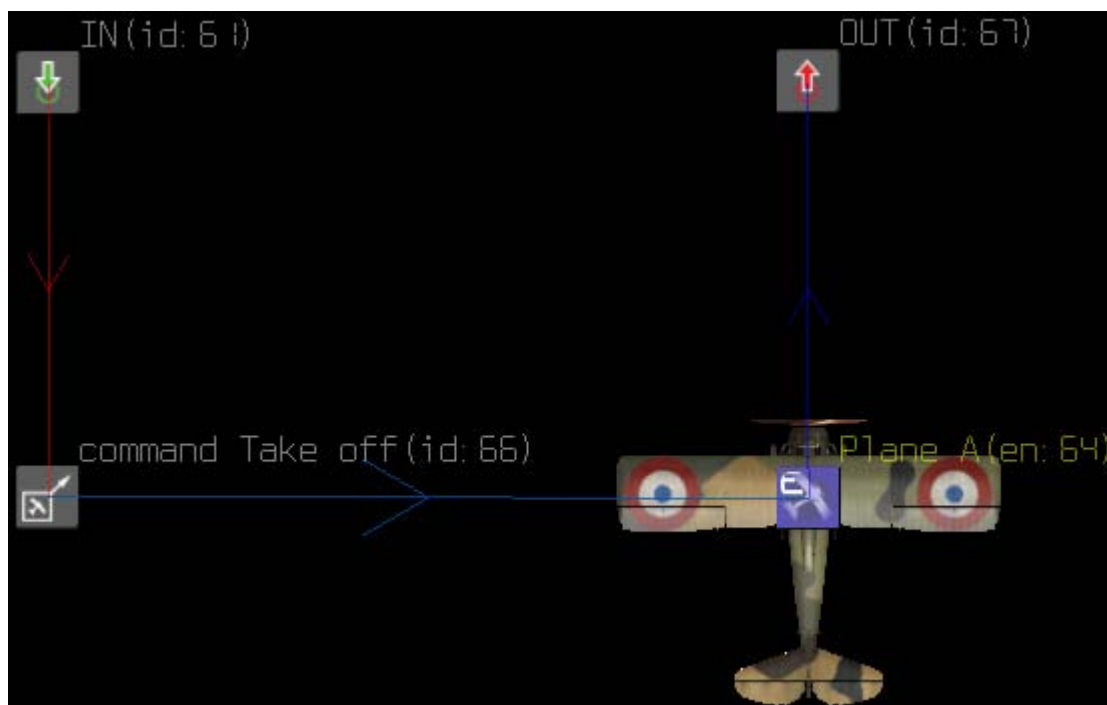
XPos = 2012.26; // координата по оси X
YPos = 107.672; // координата по оси Y
ZPos = 4385.19; // координата по оси Z
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
SoundName = ""; // звук выбранный для проигрывания, звук может быть и не указан, если тип
проигрываемого звука выбран как "config is sample"
Volume = 100; // уровень звука
}

```

7.6.13. Command:Take off



Команда самолету на взлет.



На приведенном примере:

- command Take off прилинкована объектной связью к самолету Plane A;
- IN - любое входящее событие из миссии, которое при помощи целевой связи (или Message) запускает команду Take off;
- Plane A – взлетающий самолет, в свойствах самолета задано сообщение On Take off Message (см. п 2.3.) в котором как Source указана команда Take off, а как Target любое последующее действие, которое должен совершить самолет после взлета (OUT).

В целом, для того, чтобы самолет взлетел, достаточно просто дернуть команду Take off, которая объектной связью ссылается на самолет, но обычно этого не достаточно, потому как самолет после

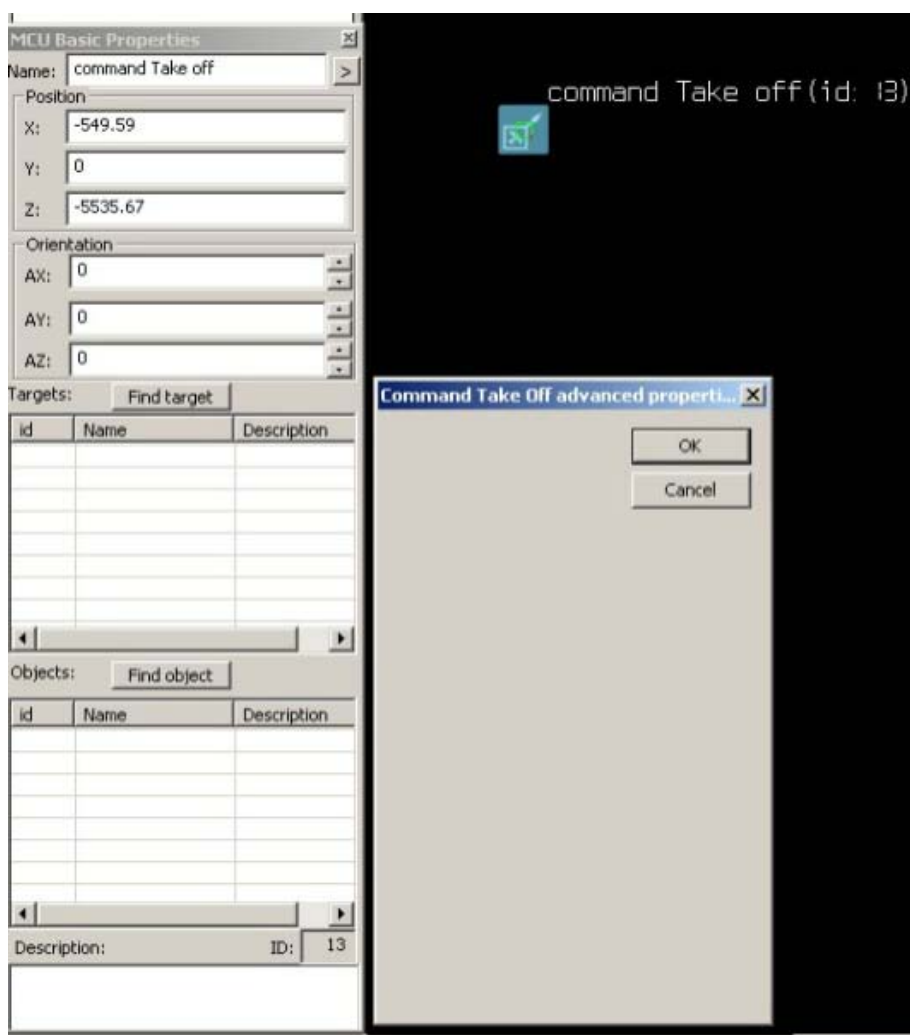
взлета (отчета о взлете) должен совершить (получить следующую команду) последующее действие.

Чаще всего, достаточно для самолета в On Take off Message указывать только Target (то есть действие после отчета AI о взлете см. п 2.3.), но иногда требуется указать и Source, то есть по выполнению какой именно команды на взлет (отчет о выполнении) последует следующее действие (Target). Периодически это требуется, когда в миссии есть несколько команд на взлет (когда это требуется сюжетом миссии), для того, чтобы самолет (AI) знал четкую последовательность цепочки Message.

Обычно первым действием после взлета указывается первая маршрутная точка, т.е. самолет по выполнению команды на взлет, направится к этой точке (см.п. 9.1.).

Взлет самолета, может быть осуществлен из любой точки на карте, поверхность которой позволит AI произвести безопасный взлет (если площадка будет неровной – AI может разбиться).

Даблклик по иконке команды открывает меню Take Off Advanced Properties:



Для этой команды не задается каких-либо дополнительных свойств.

Текстовый формат команды:

```
MCU_CMD_TakeOff //тип MCU
{
    Index = 16; //ID MCU
```



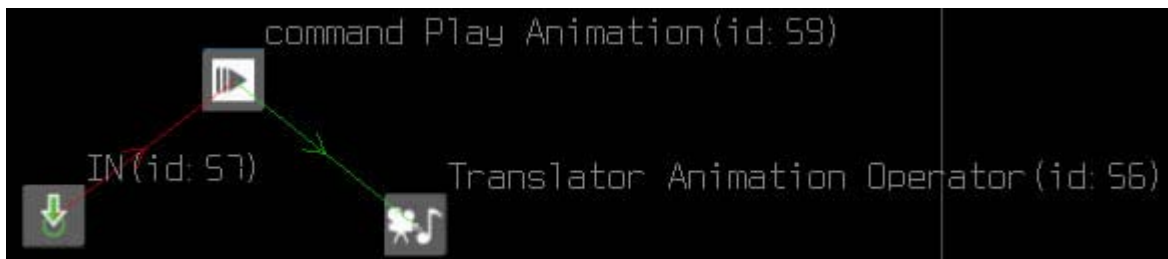
```

Name = "command Take off"; //имя MCU (не отображается в GUI)
Desc = ""; //описание MCU (не отображается в GUI)
Targets = []; // ID объектов, на которые целевой связью ссылается данная команда, у данной
команды исходящих целевых связей быть не может
Objects = []; //ID объектов, на которые объектной связью ссылается данная команда
XPos = 2095.66; // координата по оси X
YPos = 114.842; // координата по оси Y
ZPos = 4197.53; // координата по оси Z
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
}

```

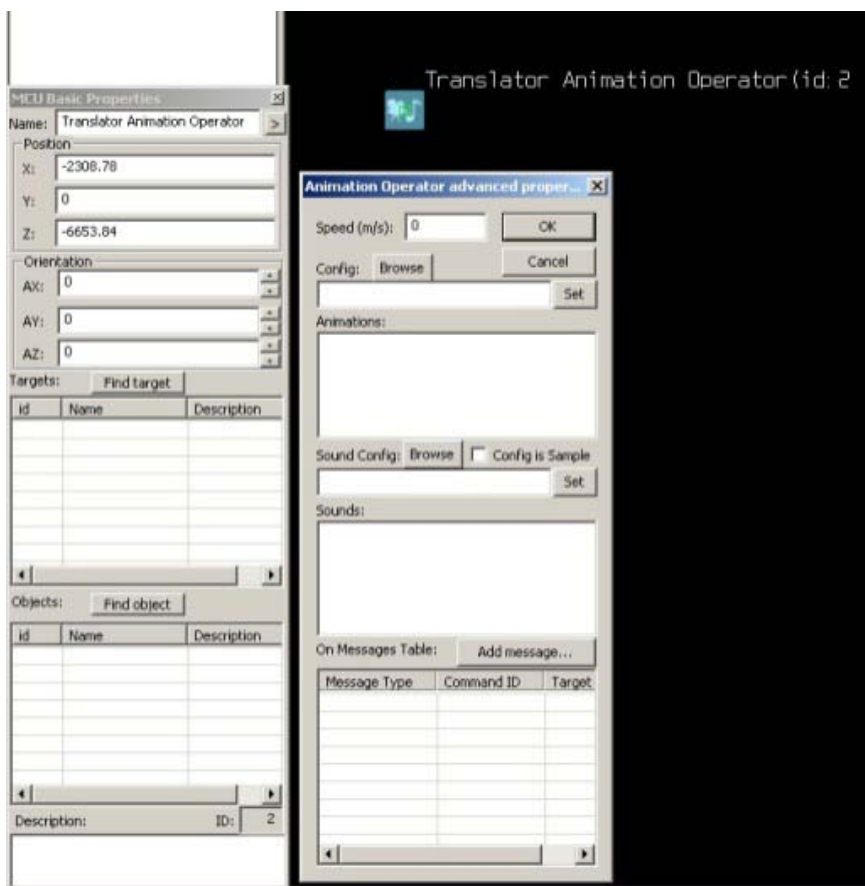
7.7. MCU Translator.

7.7.1. Translator:Animation Operator



На приведенном примере:

- Translator:Animation Operator прилинкована объектной связью к команде Play Animation (см. п.7.6.11).;
- IN - любое исходящее из миссии событие, запускающее команду Play Animation;



Проигрывает 3D анимационные сцены. При наличии в миссии этого транслятора, проигрывается анимация в состоянии idle (исходное).

Так же стоит помнить, что при наличии установленной анимации (сцена прописана в конфиге Translator:Animation Operator), эта самая сцена, в состоянии idle, появится в миссии с самого её начала. Поэтому в случае если, по сюжету миссии нужно чтобы эта сцена появилась не сначала миссии, а в заданный момент, то нужно деактивировать сцену триггером Deactivate, после чего при её запуске её нужно активировать триггером Activate (сцена появится в мире) и уже после этого, запускать анимацию или звук сцены если это нужно.

Для того, чтобы проиграть анимацию следует использовать команду Play Animation. Команда «привязывается» к этому транслятору с помощью ОС. Даблклик по иконке транслятора открывает меню Play Animation Advanced Properties.

Advanced Properties:

Поле Speed (m/s): - задает начальную скорость движения сцены в мире игры.

Поле Config: указывает путь к файлу описывающему анимации в сцене. Файл имеет расширение *.chr

Кнопка Set – подтверждает выбор пути к файлу.

Список Animations: в случае, если все выбрано корректно (правильно подключен *.chr файл), в этом окне появится перечень анимаций, которые могут быть проиграны с помощью команды Play Animation.

Sound Config: - указывает путь к файлу скрипта описывающего звуки в сцене. Файл имеет вид *.lua.

В большинстве случаев для того чтобы проиграть звук в сцене, удобнее указывать непосредственно звуковой семпл (в формате *.mp3), для этого, нужно поставить галочку Config is Sample и указать путь к звуковому файлу (*.mp3), после чего нужно при помощи команды PlaySound задать момент начала проигрывания звука (см. п. 7.6.12).

Список Sounds: - в случае, если все выбрано корректно, в поле списка появляется перечень звуков, которые могут быть проиграны с помощью команды Play Sound (см. п. 7.6.12.), кроме случая, когда звук это семпл.

Поле On Message Table: - позволяет назначить сообщение OnAnimationStop после проигрывания анимации.

Кнопка Add message... - задает тип сообщения.

Существует возможность проиграть установленную сцену в редакторе, для этого в уже настроенной сцене (в поле "Animation Name" выставлена анимация), ПКМ на команду Play Animatuon, Select Object Menu =>Play.

Текстовый формат транслятора:

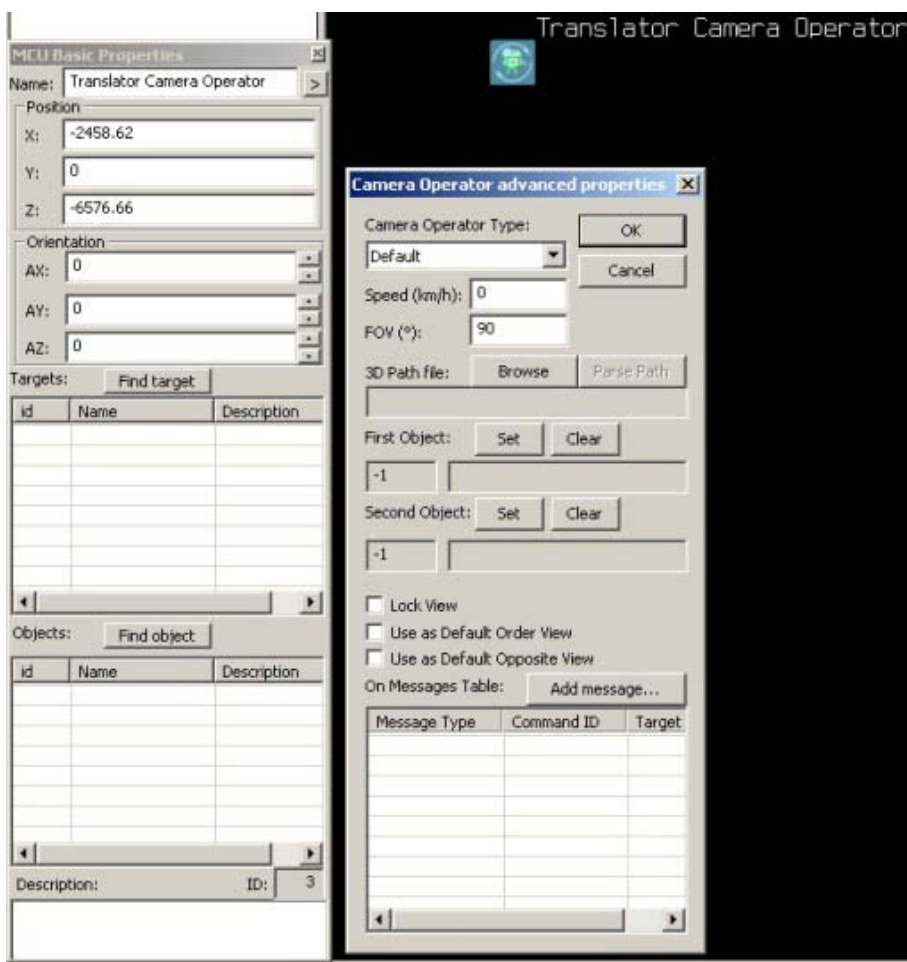
```
MCU_TR_AnimationOperator //тип MCU
{
  Index = 17; //ID MCU
  Name = "Translator Animation Operator"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного
транслятора исходящих целевых связей быть не может
  Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у
данного транслятора исходящих объектных связей быть не может
  XPos = 2089.71; // координата по оси X
  YPos = 114.516; // координата по оси Y, важно выставлять т.к. параметр определяет высоту
появления объекта или звука
  ZPos = 4182.64; // координата по оси Z
  XOri = 0; // поворот по оси X
  YOri = 0; // поворот по оси Y
  ZOri = 0; // поворот по оси Z
  Enabled = 1; //через интерфейс редактора не используется
  Speed = 0;
  Config = "";
  SoundConfig = ""; //прописанный звуковой конфиг
  ConfigIsFile = 0; //чекбокс если звуковой файл это отдельная звуковая дорожка (*.mp3)
  OnMessages //список назначенных сообщений
  {
  }
}
```

7.7.2. Translator:Camera Operator



На приведенном примере:

- Translator: Camera Operator – объект-камера в миссии, которая запускается любым событием миссии;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Camera Operator;



Позволяет разместить камеру в мире игры. Даблклик по иконке транслятора открывает меню Camera Operator Advanced Properties.

Advanced Properties:

Список Camera Operator Type: - позволяет выбрать тип камеры.

Типы камер:

Default – вид при старте миссии или вид из кокпита, так же используется как дефолтная камера после старта миссии (см. ниже).

Cockpit – НЕ работает

External Free – камера, смотрящая на самолет (или объект) выбранный в группе First Object, не зависящая от его ориентации в пространстве.

External Fixed – камера, смотрящая на самолет (или объект) выбранный в группе First Object, привязанная к определенному ракурсу зрения на самолет и меняющая положение в зависимости от изменения положения самолета.

Flyby – камера, показывающая пролет самолета (или объекта), выбранного в группе First Object.

External Chase – камера, в ракурсе которой находятся 2 объекта, указанные в группах First Object и Second Object.

Free – свободно перемещаемая игроком камера.

Self – камера которая смотрит туда, куда указано в ее базовых настройках, или камера, которая передвигается по CameraPoint или в соответствии с 3D path скриптом, то есть это камера – вид от первого лица.

Padlock – сопровождающая камера, следящая с местоположения Camera Operator за объектом указанным в группе Second Object, при этом в группе First Object должен быть указан сам транслятор.

Поле Speed – скорость перемещения камеры в км/ч.

Поле FOV (°): - ширина поля зрения камеры в градусах.

3D Path file: - возможность «подцепить» выгруженную из 3D редактора траекторию полета камеры. Файлы *.cam. Для более подробного описания использования данной функции, см. п. 9.7 и 9.8.

Группа First Object: устанавливает «первый объект» для камеры. Кнопка Set вызывает список трансляторов для выбора, кнопка Clear отменяет выбор транслятора.

Группа Second Object: устанавливает «второй объект» для камеры. Кнопка Set вызывает список трансляторов для выбора, кнопка Clear отменяет выбор транслятора.

Чекбокс Lock View – во включенном состоянии отменяет возможность переключения на другую камеру, то есть, если в миссии активируется камера с такой настройкой, то игрок не сможет переключиться на другую камеру, до момента пока не активируется следующая камера.

Чекбокс Use as Default Order View – определяет камеру как стартовую для игроков Order Coalition. Если таких камер несколько, то выбор камеры при старте миссии происходит случайным образом.

Чекбокс Use as Default Opposite View - определяет камеру как стартовую для игроков Opposite Coalition. Если таких камер несколько, то выбор камеры при старте миссии происходит случайным образом.

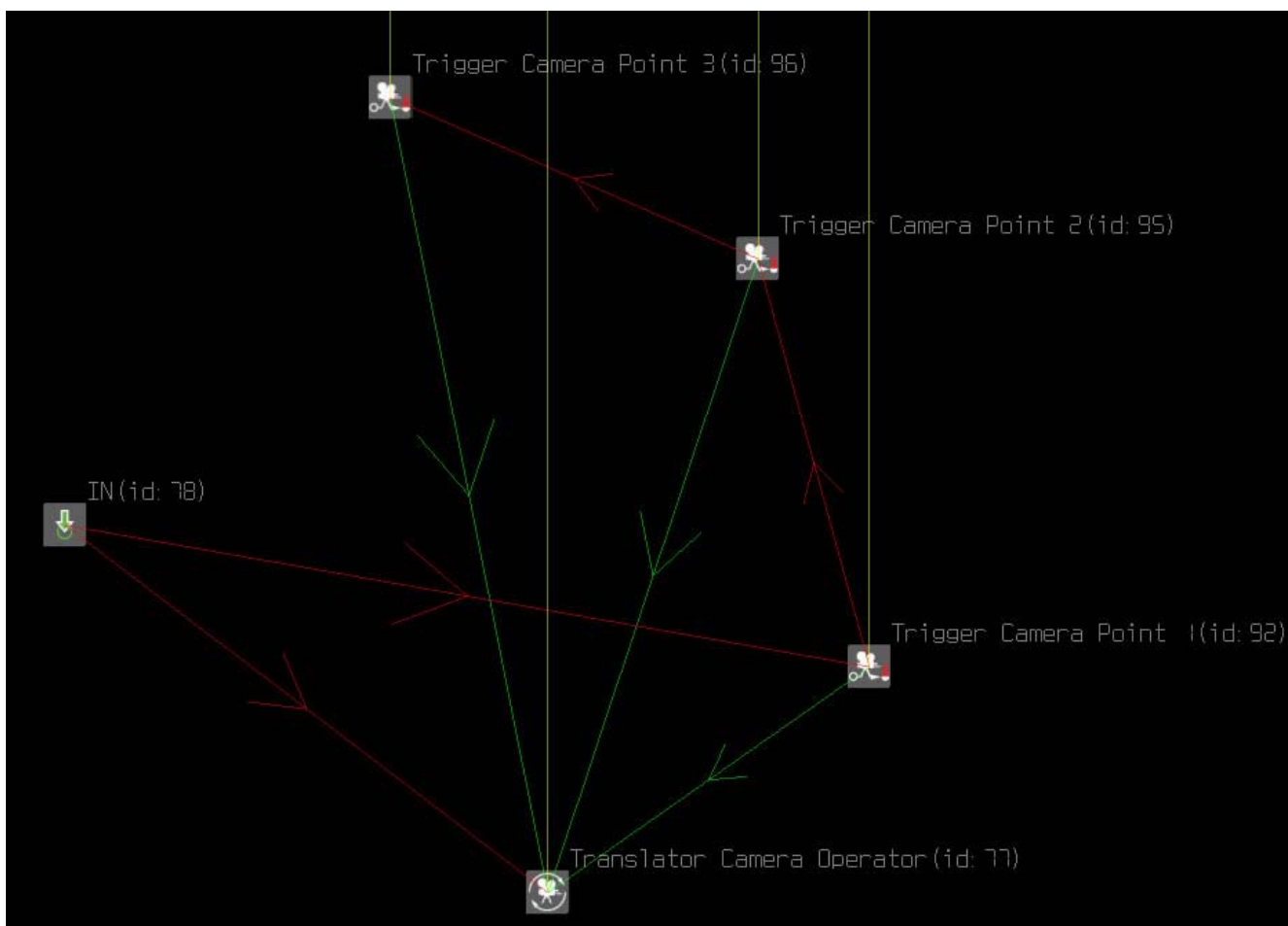
В случае, когда камер типа Default несколько, то в миссии можно переключаться по ним при помощи F12 (F12 – order, ctrl+F12 – opposite),

Если выделить Translator:Camera Operator, то можно увидеть створ камеры (ширина угла будет показываться в зависимости от установленного FOV в свойствах) и ось взгляда.

Так же, в случае если вам нужно точно выставить кадр камеры (например с случае камеры типа Self, т.е. камера от первого лица), то выделите камеру, правой кнопкой мыши Select Object Menu => Look from This Object. После выставления кадра камеры, кликните правой кнопкой мыши в рабочую область редактора Select Object Menu => Copy Camera Position and Orientation to Object, после чего положение камеры скопируется в транслятор.



В случае когда Translator:Camera Operator используется совместно с Trigger:Camera point (см. п. 7.8.2.), то есть камера должна будет двигаться по заданной триггерами Camera point траектории, существует возможность автоматизированной расстановки Camera point.



Для этого выделите Translator:Camera Operator, нажмите левый Shift и расставьте контрольные точки.

Не забудьте выставить в камере её FOV, т.к. Trigger:Camera point автоматически выставит свойства контрольным точкам взяв их из Translator:Camera Operator. При этом, скорость камеры в Camera point, возьмется из первой контрольной точки камеры.

При этом нужно помнить, что если по сюжету миссии, нужно запустить камеру по контрольным точкам сразу, то Translator:Camera Operator не может «запустить» Trigger:Camera point, т.к. сам по себе транслятор не имеет исходящих целевых связей.

Поэтому входящее событие (IN) должно одновременно запускать как сам оператор, так и его первую контрольную точку, при этом камера будет передвигаться из начального положения оператора к заданной точке, со скоростью заданной в первой контрольной точке камеры.

При передвижении камеры, скорость и положение камеры будут плавно изменяться от точки к точке в зависимости от настроек и положения точек.

Текстовый формат транслятора:

```
MCU_TR_CameraOperator //тип MCU
{
  Index = 18; //ID MCU
  Name = "Translator Camera Operator"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
```

```

Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного
транслятора исходящих целевых связей быть не может
Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у
данного транслятора исходящих объектных связей быть не может
XPos = 2012.26; // координата по оси X, выставляется в редакторе – куда смотрит камера
YPos = 110.272; // координата по оси Y (высота), выставляется в редакторе – куда смотрит камера
ZPos = 4069.45; // координата по оси Z, выставляется в редакторе – куда смотрит камера
XOri = 0; //выставляется в редакторе – наклон кадра камеры (крен)
YOri = 0; //выставляется в редакторе – поворот камеры в горизонтальной плоскости
ZOri = 0; // выставляется в редакторе – поворот камеры в вертикальной плоскости
Enabled = 1; //через интерфейс редактора не используется
Speed = 0; //начальная скорость камеры
FOV = 90; //угол обзора камеры
FirstId = -1; //ID первого объекта на который ссылается камера (-1 – объект не назначен)
SecondId = -1; //ID второго объекта на который ссылается камера (-1 – объект не назначен)
LockView = 0; //переключение с камеры заблокировано
CameraType = 0; //тип камеры
DefOrderView = 0; //используется как дефолтная камеры для Order
DefOppositeView = 0; //используется как дефолтная камеры для Opposite
Config = ""; //имя прописанного конфига камеры (*.cam)
OnMessages //список назначенных сообщений
{
}
}

```

7.7.3. Translator:Icon



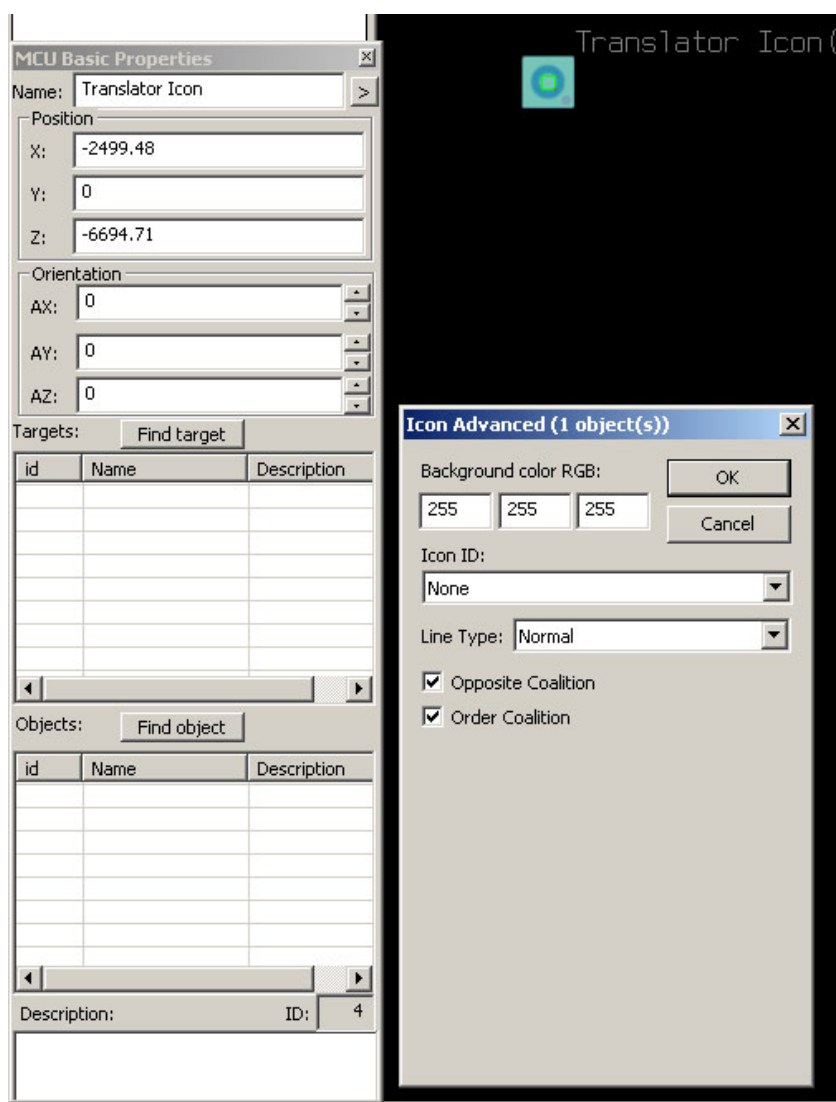
Translator:Icon выполняет функцию визуализации для полетной карты.

То есть, если вам нужно указать на карте маршрут полета, зону атаки, зону патрулирования, то это нужно делать с помощью данного транслятора, т.к. элементы миссии, такие как Trigger:Waypoint или другие элементы, не будут самостоятельно отображаться на полетной карте.

Данный механизм преследует следующие цели:

- дизайнер миссии может визуализировать любой элемент миссии на полетной карте;

- система локализации ROF вынесет все названия и описания карты в локализационный файл миссии (см. п. 5.2.);
- возможность показывать маршрут разным сторонам индивидуально (в случае мультиплеерной миссии);
- возможность по разному визуализировать карту (цвет линий, их толщина, тип иконки).



Advanced Properties:

Поле Background color: - определяет цвет исходящей из иконки линии маршрута на полетной карте в стандартной палитре RGB, задавая интенсивность красной, зеленой и синей составляющей цвета линии.

Для визуального отображения линий, иконки должны быть соединены между собой целевой связью.

При выставлении иконки на карте, если ввести в поле Name текст, это текст будет отображаться на карте, например «№1», или «Точка атаки».

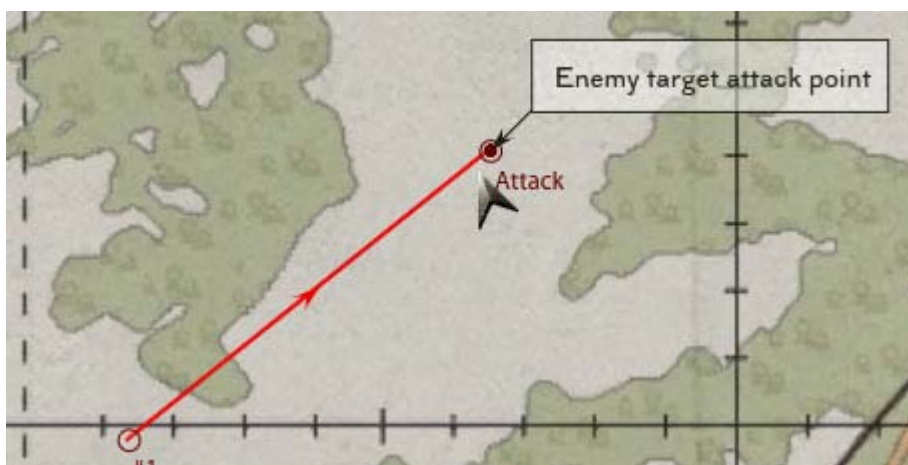
Если в поле Description ввести текст, то этот текст будет отображаться при наведении курсора на точку полетной карты, например, при наведении курсора на точку с названием «№1» или

«Attack», появится всплывающее окно (tooltip) карты, в котором можно указать подробное описание точки, например «Waypoint №1» или «Enemy target attack point» соответственно:

вид в редакторе:



вид на полетной карте (GUI):



Так же, дизайнер миссии может составлять любые фигуры, при помощи данного транслятора, для этого нужно выбрать тип иконки как “None”, например:

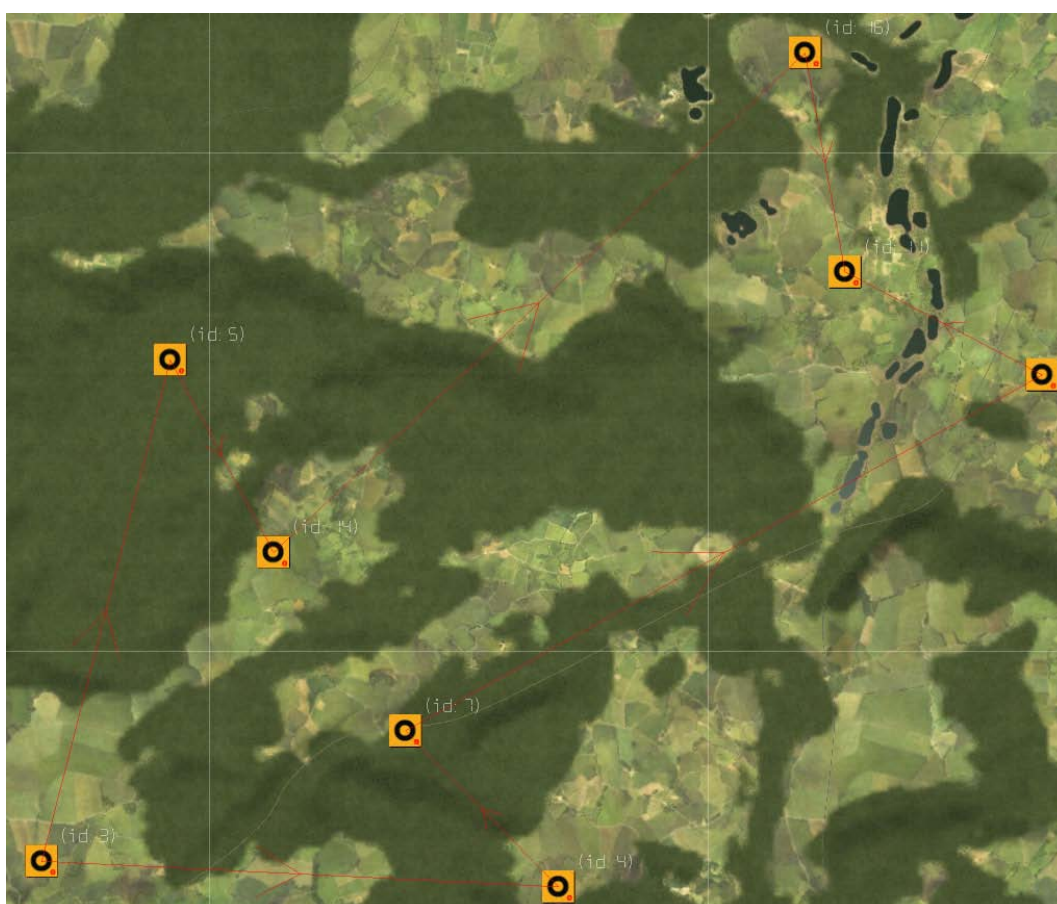
вид в редакторе:



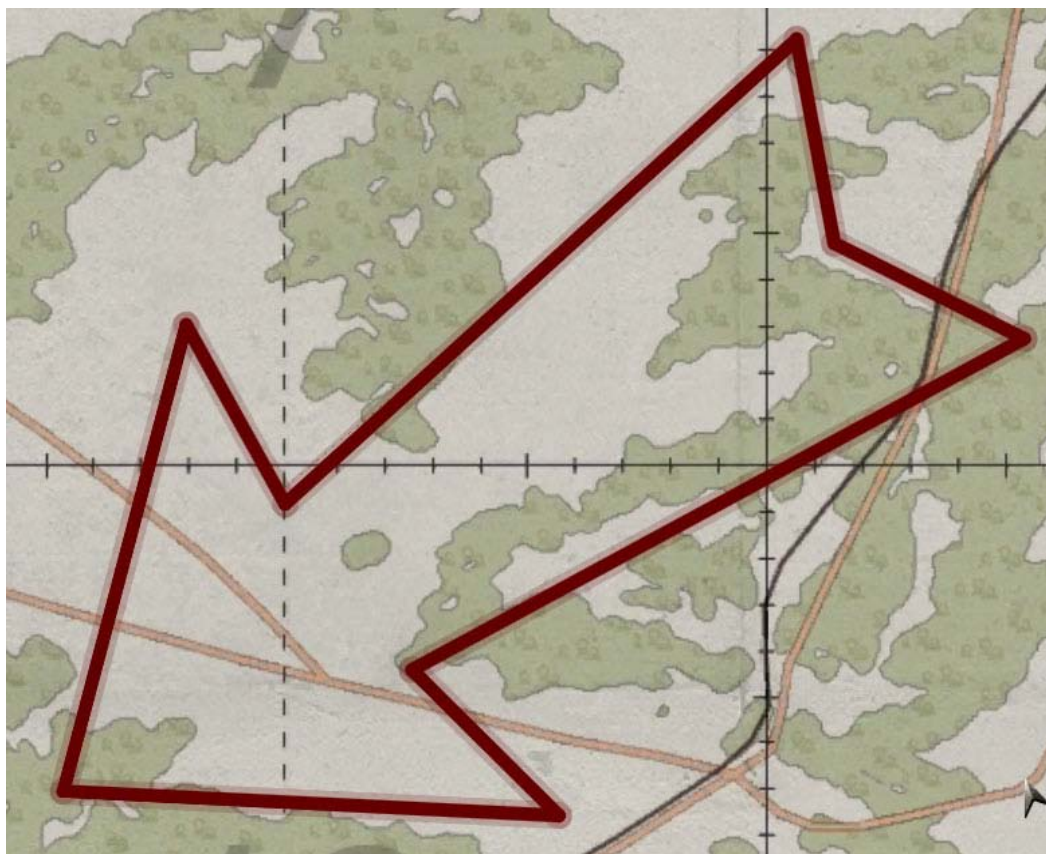
вид на полетной карте (GUI):



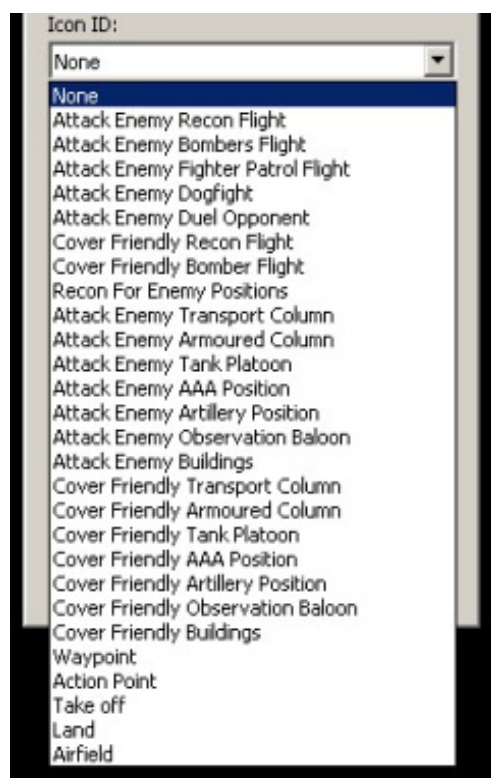
вид в редакторе:



вид на полетной карте (GUI):



Поле Icon ID: - определяет тип иконки на карте.



Типы:

None – нет иконки и ее изображения на карте;



Attack Enemy Recon Flight – атака разведчиков противника;



Attack Enemy Bombers Flight – атака бомбардировщиков противника;



Attack Enemy Fighter Patrol Flight – атака истребителей противника;



Attack Enemy Dogfight – помощь своим в воздушном бою с истребителями противника;



Attack Enemy Duel Opponent – воздушный бой - дуэль;



Cover Friendly Recon Flight – прикрытие своих разведчиков;



Cover Friendly Bomber Flight – прикрытие своих бомбардировщиков;



Recon For Enemy Position – район разведки вражеских позиций;



Attack Enemy Transport Column – атака транспортной колонны противника;



Attack Enemy Armoured Column – атака бронированной колонны противника;



Attack Enemy Tank Platoon – атака танков противника;



Attack Enemy AAA Position – атака позиций зенитной артиллерии противника;



Attack Enemy Artillery Position – атака позиций артиллерии противника;



Attack Enemy Observation Balloon – атака аэростата наблюдения противника;



Attack Enemy Buildings – атака наземных строений противника;



Cover Friendly Transport Column – прикрытие дружественной транспортной колонны;



Cover Friendly Armoured Column – прикрытие дружественной бронированной колонны;



Cover Friendly Tank Platoon – прикрытие дружественных танков;



Cover Friendly AAA Position – прикрытие дружественной позиций зенитной артиллерии;



Cover Friendly Artillery Position – прикрытие позиций дружественной артиллерии;



Cover Friendly Observation Balloon – прикрытие дружественного аэростата наблюдения;



Cover Friendly Buildings – прикрытие дружественных наземных строений;



Waypoint – контрольная точка маршрута полета;



Action point – точка основных действий по маршруту полета;

Take off - ;



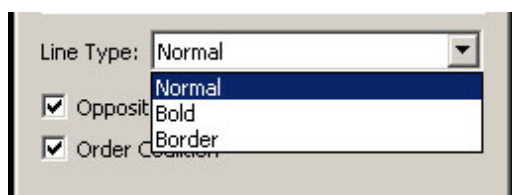
Land – место посадки;



Airfield - аэродром.

Поле Line Type: - определяет тип линий маршрута полета на карте.

Типы:

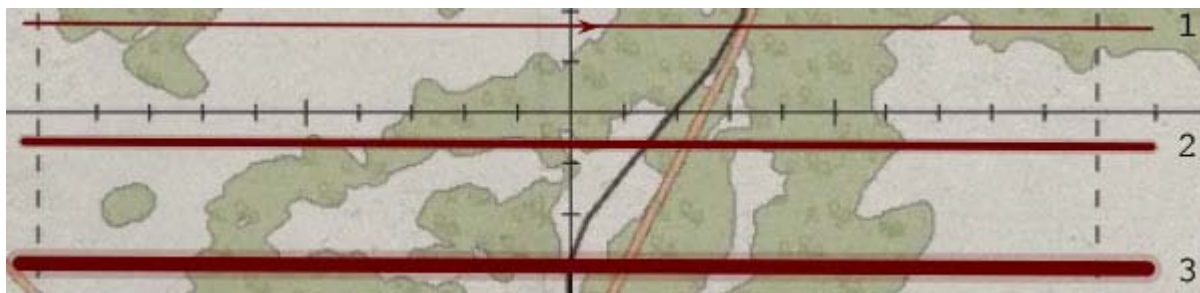


Normal (1) – тонкая линия со стрелкой, указывающей направление полета, при этом между соединенными иконками будет рисоваться стрелка указывающая направление «от» и «к», обычно используется для визуализации маршрута полета ;

Bold (2)– толстая линия;

Border (3)– толстая линия с границей.

вид разных типов линий на полетной карте в игре:



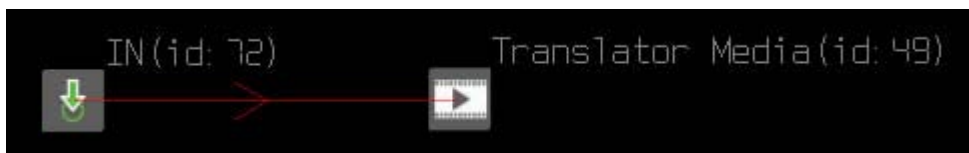
Чекбоксы Opposite Coalition и Order Coalition: - во включенном состоянии определяют сторону, которой принадлежит иконка на карте, а также сторону, которая будет видеть маршрут на карте.

Существует возможность автоматизированной расстановки иконок, которые связаны между собой целевой связью, для этого выделите иконку, нажмите Shift и расставьте нужное вам количество иконок на карте.

Текстовый формат транслятора:

```
MCU_Icon //тип MCU
{
    Index = 19; //ID MCU
    Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор
    Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у данного
    транслятора исходящих объектных связей быть не может
    XPos = 2045.03; // координата по оси X
    YPos = 112.067; // координата по оси Y
    ZPos = 4283.91; // координата по оси Z
    XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
    YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
    ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
    Enabled = 1; //через интерфейс редактора не используется
    LCName = 3; //индекс строки локализации в локализационном файле миссии, в GUI отобразится
    как название иконки
    LCDesc = 4; //индекс строки локализации в локализационном файле миссии, в GUI отобразится
    как описание (всплывающее окно при наведении курсора) иконки
    OppositeAirplanes = 1; //будет видна только для команды Order
    OrderAirplanes = 1; // будет видна только для команды Opposite
    IconId = 0; //тип иконки
    RColor = 255; //цветовая составляющая красного
    GColor = 255; //цветовая составляющая зеленого
    BColor = 255; //цветовая составляющая синего
    LineType = 0; //тип линии
}
```

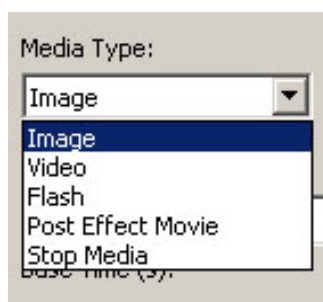
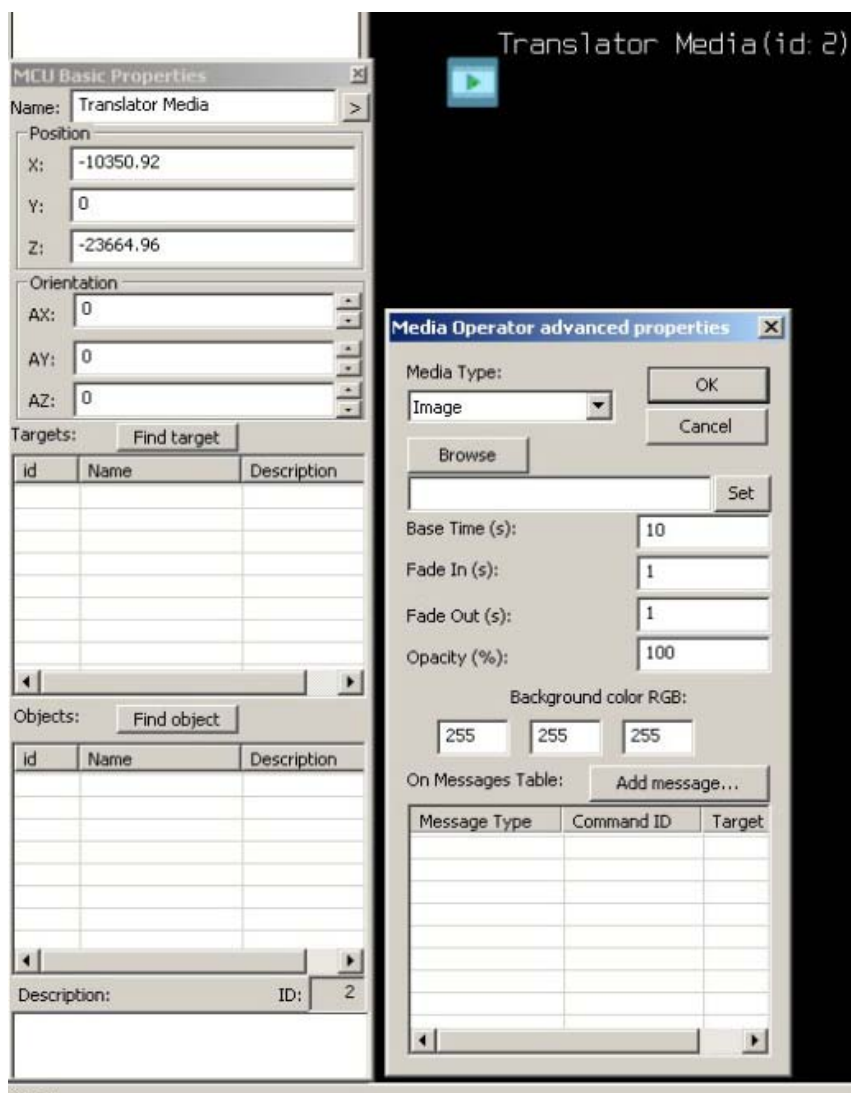
7.7.4. Translator:Media



На приведенном примере:

- Translator: Media – медиа вставка в миссию;

- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Translator: Media;



Различные варианты Media в миссии, графические, видео или флэш вставки в миссию.

Даблклик по иконке транслятора открывает меню Media Advanced Properties.

Advanced Properties:

Список Media Type: возможные варианты media, которые проигрываются в миссии.

Image – картинка (поддержка *.jpg, *.dds файлов);

Video – видео вставка, поддержка bink video (*.bik);

Flash – флэш вставка(*.swf);

Post Effect Movie – наложение постэффекта на текущую картинку, накладывается «эффект старого кино» на экран;

Stop Media - окончание проигрывания любой Media.

Кнопка Browse – указание пути к нужному файлу Media.

Кнопка Set – подтверждение выбранной Media.

Поле Base Time (s): - время, которое Media находится на экране, исключая время проявления и постепенного исчезновения картинки.

Поле Fade In (s): - время постепенного проявления Media, т.е. время, необходимое для того, чтобы насыщенность Media поменялось с 0 до значения заданного в поле Opacity (%):

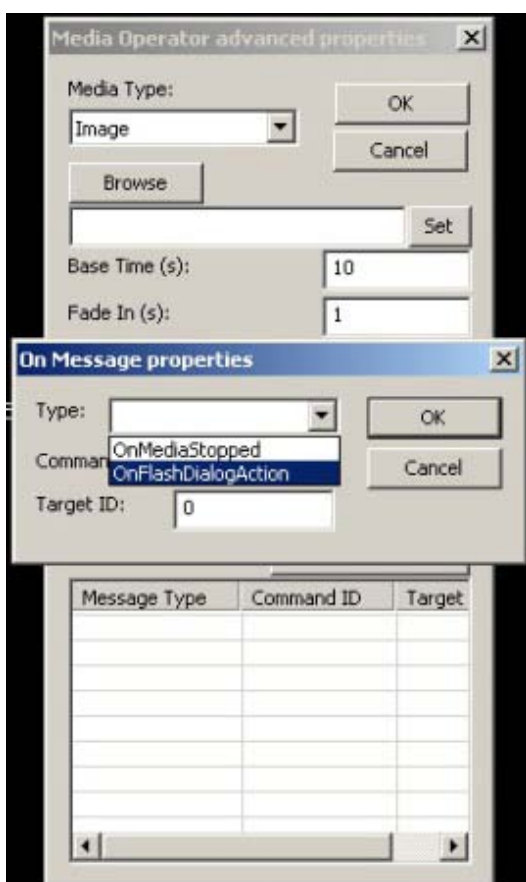
Поле Fade Out (s): - время постепенного исчезновения Media, т.е. время, необходимое для того, чтобы насыщенность Media поменялось от значения заданного в поле Opacity (%): до 0.

Поле Opacity (%): - задает значение насыщенности (непрозрачности) Media в % в течении Base Time.

Группа Background color RGB: - регулирует цвет заднего фона, задавая интенсивность красной, зеленой и синей составляющих цвета.

Поле On Message Table: - позволяет назначить сообщение OnAnimationStop после проигрывания анимации, то есть, после окончания проигрывания Media – можно выдать событие в миссию.

Кнопка Add message... - работает только со Stop Media Type и задает тип сообщения.



OnMediaStopped – по окончании проигрывания Media автоматически запускает следующее событие в миссии (через Message).

OnFlashDialogAction – «вручную» (по нажатии определенной клавиши или поля в миссии) запускает следующее событие в миссии.

Например, можно назначить Flash кнопку Skip и тогда, в процессе проигрывания Media, нажав кнопку Skip, транслятор «запустит» следующее событие.

ВНИМАНИЕ!: в настоящей версии ППМ работает только OnMediaStopped, причем с функционалом OnFlashDialogAction, например – по нажатию Skip в тренировочных миссиях.

ВНИМАНИЕ!: - после запуска любого Translator:Media будет недоступно ускорение времени в миссии. Поэтому, после окончания работы Translator:Media вам нужно «дернуть» Translator:Media с установленным в нем Media Type: Stop Media (в его свойствах).

ВНИМАНИЕ!: - на данном этапе развития проекта, механизм игры не управляет временем внутри Flash (весь механизм визуализации построен на этом движке), поэтому, во время проигрывания Media игру НЕЛЬЗЯ СТАВИТЬ НА ПАУЗУ (это сделано как механизм-предохранитель).

Поэтому, если вы используете Media сразу после загрузки миссии (а при загрузке игра становится на паузу примерно через 1-2 секунды после загрузки миссии) – помните об этой тонкости. Для примера использования Media в таком случае смотрите пункт 8.8. с разбором примера использования.

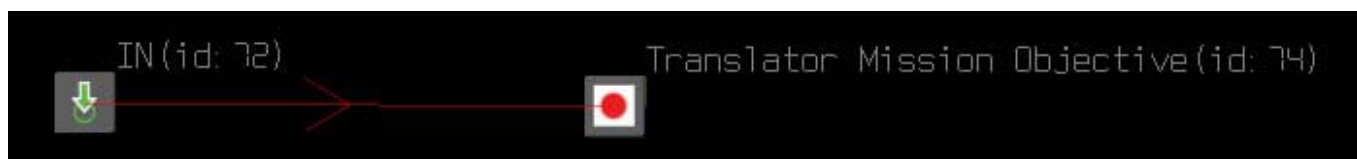
Текстовый формат транслятора:

```
MCU_TR_Media //тип MCU
{
  Index = 20; //ID MCU
  Name = "Translator Media"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного
типа транслятора целевых связей быть не может
  Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у
данного транслятора исходящих объектных связей быть не может
  XPos = 2024.18; // координата по оси X
  YPos = 109.304; // координата по оси Y
  ZPos = 4355.4; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Enabled = 1; //через интерфейс редактора не используется
  Config = ""; //путь к прописанному в трансляторе конфигу
  TotalTime = 10; //общее время действия транслятора (Bas Time)
  ExpandTime = 1; //время проявления медиа вставки транслятора (Fade In)
  FadeTime = 1; //время пропадания медиа вставки транслятора (Fade Out)
```

Opacity = 100; //прозрачность медиа вставки транслятора
 RColor = 255; //красная составляющая цвета заднего фона
 GColor = 255; //зеленая составляющая цвета заднего фона
 BColor = 255; //синяя составляющая цвета заднего фона
 MediaType = 0; //тип медиа вставки, 0=Image, 1=Video, 2=Flash, 3=Post Effect Movie, 4=Stop Media
 OnMessages //список назначенных сообщений

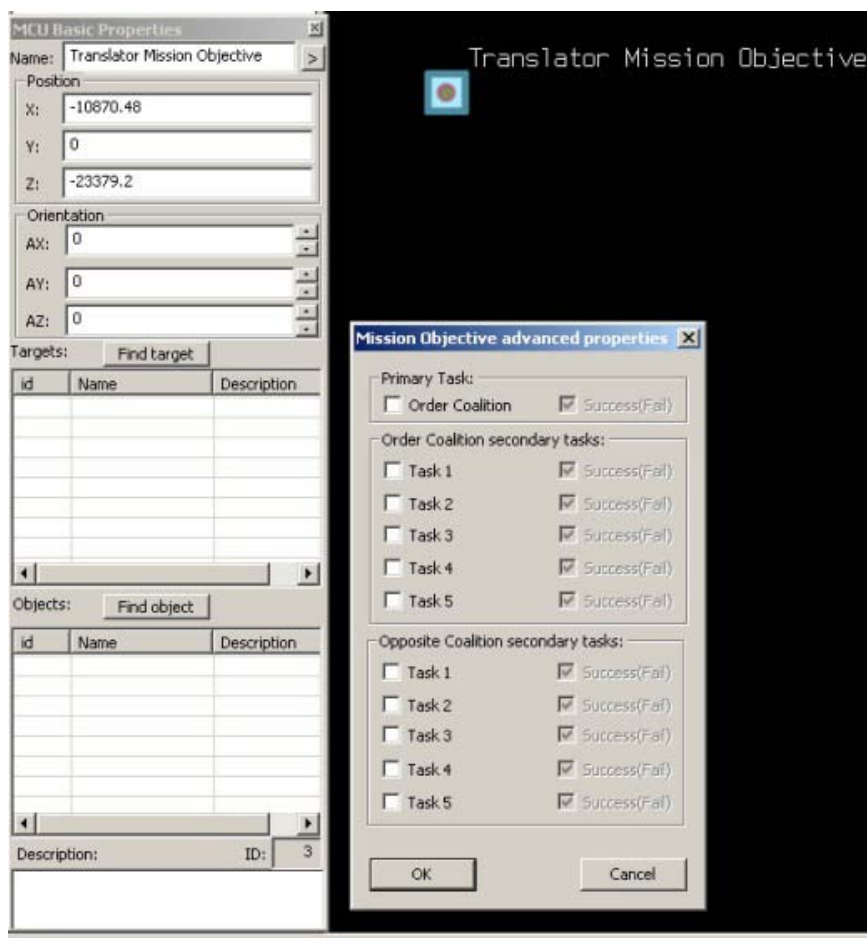
```
{
}
}
```

7.7.5. Translator:Mission Objective



На приведенном примере:

- Translator:Mission Objective – транслятор, определяющий выполнение/невыполнение задач миссии;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Translator: Mission Objective.



Транслятор, определяющий первичную (основную) или вторичные (дополнительные) цели миссии для противоборствующих сторон. Даблклик по иконке транслятора открывает меню Mission Objective Advanced Properties.

Advanced Properties:

Для назначения первичной задачи:

Группа Primary Task: определяет первичную задачу коалиции (Order) и ее результат (Success/Fail). Чекбокс Order Coalition всегда должен быть включен! Иначе транслятор не работает.

Например, если включить чекбокс Success/Fail, это будет означать, что Order Coalition будет засчитано выполнение первичной задачи. Если этот чекбокс выключить (убрать галочку), то коалиции Order будет засчитано поражение. При помощи проверок заданных условий, дизайнер миссии определяет, в каком случае и что именно работает.

Группы Oder Coalition secondary tasks и Opposite Coalition secondary tasks: в этом случае не используются.

Для назначения вторичной задачи:

Группа Primary Task: в этом случае не используется.

Группа Oder Coalition secondary tasks: определяет вторичную (ые) задачу (и) для стороны Order.

Чекбоксы Task 1 – 5 при включенном состоянии позволяют определить до 5 вторичных задач;

Чекбоксы Success(Fail) при включенном состоянии (Success), в случае успешного выполнения, будет засчитано в статистику, как выполнено и отображено на полетной карте. При выключенном состоянии - будет засчитано в статистику, как провалено и отображено на полетной карте.

Группа Opposite Coalition secondary tasks: определяет вторичную (-ые) задачу (-и) для стороны Opposite.

Чекбоксы Task 1 – 5 при включенном состоянии позволяют определить до 5 вторичных задач.

Чекбоксы Success(Fail) при включенном состоянии (Success), в случае успешного выполнения, будут засчитаны в статистику, как выполненные и отображены на полетной карте. При выключенном состоянии - будут засчитаны в статистику, как проваленные и отображены на полетной карте (см.ниже).

Ограничение: нельзя использовать один и тот же транслятор для определения более одной задачи одновременно.

Количество вторичных целей в миссии не может быть более пяти для одной стороны.

В случае, если в миссии работает хотя бы один Mission Objective Fail– данная задача будет засчитана как проваленная, вне зависимости от количества работавших Mission Objective Success.

Количество установленных в миссии Translator:Mission Objective может быть любым - в зависимости от активной ветви сценария миссии.

Поле Name – если ввести в данное поле имя, то когда транслятор работает – на карте будет показан этот текст, так же, если в поле Description ввести текст, то при наведении на карте (в

дебрифинге миссии) курсора на название выполненной задачи (текст из поля Name), во всплывающем окне выведется данный текст.

Данный транслятор по-разному отобразится в GUI (в игре, на полетной карте), в зависимости от типа выполненного (невыполненного) условия установленного в свойствах , при этом, его нужно (но не обязательно) использовать совместно с Translator:Icon – если это иконка цели и по смыслу задача (цель) находится примерно на месте иконки, в этом случае нужно совместить оба транслятора в миссии. Визуально это будет выглядеть как иконка с разными видами подложек на заднем фоне.

Текстовый формат транслятора:

```
MCU_TR_MissionObjective //тип MCU
{
  Index = 21; //ID MCU
  Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного транслятора исходящих целевых связей быть не может
  Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у данного транслятора исходящих объектных связей быть не может
  XPos = 2113.54; // координата по оси X
  YPos = 116.999; // координата по оси Y
  ZPos = 4257.11; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Enabled = 1; //через интерфейс редактора не используется
  LCName = 3; //индекс строки локализации в локализационном файле миссии, в GUI отобразится как название сработавшей задачи
  LCDesc = 4; //индекс строки локализации в локализационном файле миссии, в GUI отобразится как описание сработавшей задачи
  TaskArray //распределении настроек выполнения задачи (при выставлении разных галочек в трансляторе – разное сочетание)
  {
  }
}
```

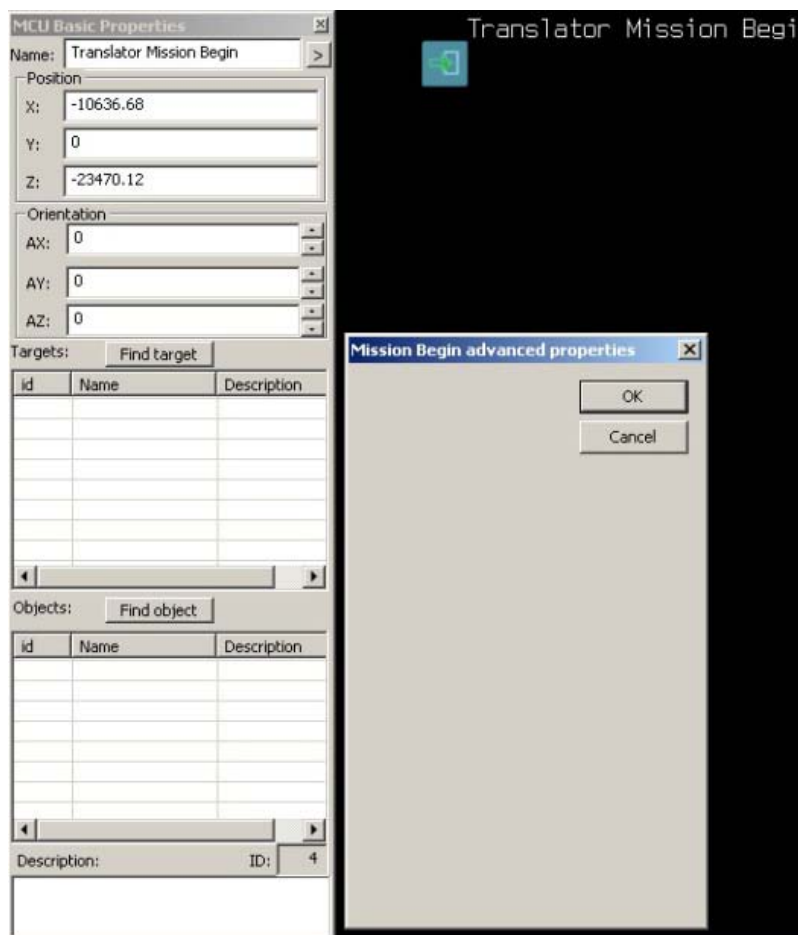
7.7.6. Translator:Mission begin



На приведенном примере:

- Translator:Mission begin – транслятор, определяющий начальное действие миссии, фактически – стартер всех событий миссии;

- OUT - любое начальное событие в миссии, которое активируется посредством :
Translator:Mission begin.



Старт для событий в миссии после ее загрузки. Транслирует из движка игры событие «начало миссии». Даблклик по иконке транслятора открывает меню Mission Begin Advanced Properties.

В миссии может быть сколько угодно таких трансляторов. Если их несколько, все они сработают при начале миссии.

Эту возможность можно использовать при импорте в миссию функциональных блоков с уже выставленными Mission Begin.

Текстовый формат транслятора:

```
MCU_TR_MissionBegin //тип MCU
{
    Index = 22; //ID MCU
    Name = "Translator Mission Begin"; //имя MCU (не отображается в GUI)
    Desc = ""; //описание MCU (не отображается в GUI)
    Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор
    Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у
данного транслятора исходящих объектных связей быть не может
    XPos = 2101.62; // координата по оси X
    YPos = 115.08; // координата по оси Y
    ZPos = 4272; // координата по оси Z
    XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
```

```

YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
Enabled = 1; //через интерфейс редактора не используется
}

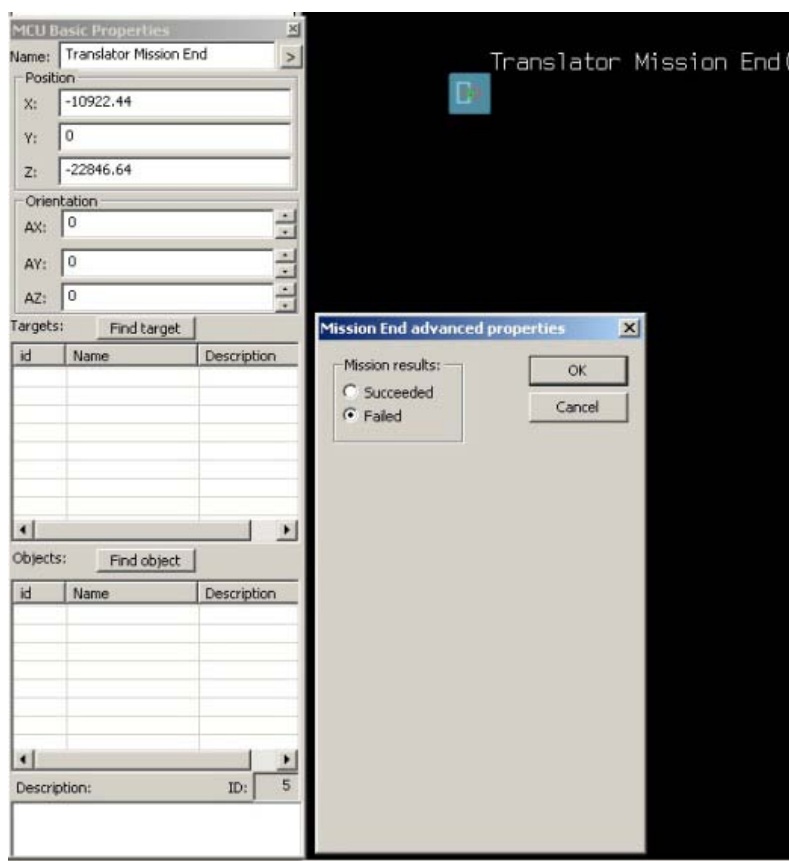
```

7.7.7. Translator:Mission End



На приведенном примере:

- Translator: Translator: Mission End – транслятор определяющий окончание миссии, переход в меню статистики миссии;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Mission End.



При активации завершает миссию в принудительном порядке без вмешательства игрока. Даблклик по иконке транслятора открывает меню Mission End Advanced Properties.

Advanced Properties:

Поле Mission Results: - имеет 2 положения Succeeded и Failed, означающие, соответственно, успешное завершение миссии и проигрыш – в текущей версии ROF данная функция Advanced Properties транслятора Mission End - НЕ РАБОТАЕТ.

Текстовый формат транслятора:

```
MCU_TR_MissionEnd //тип MCU
```

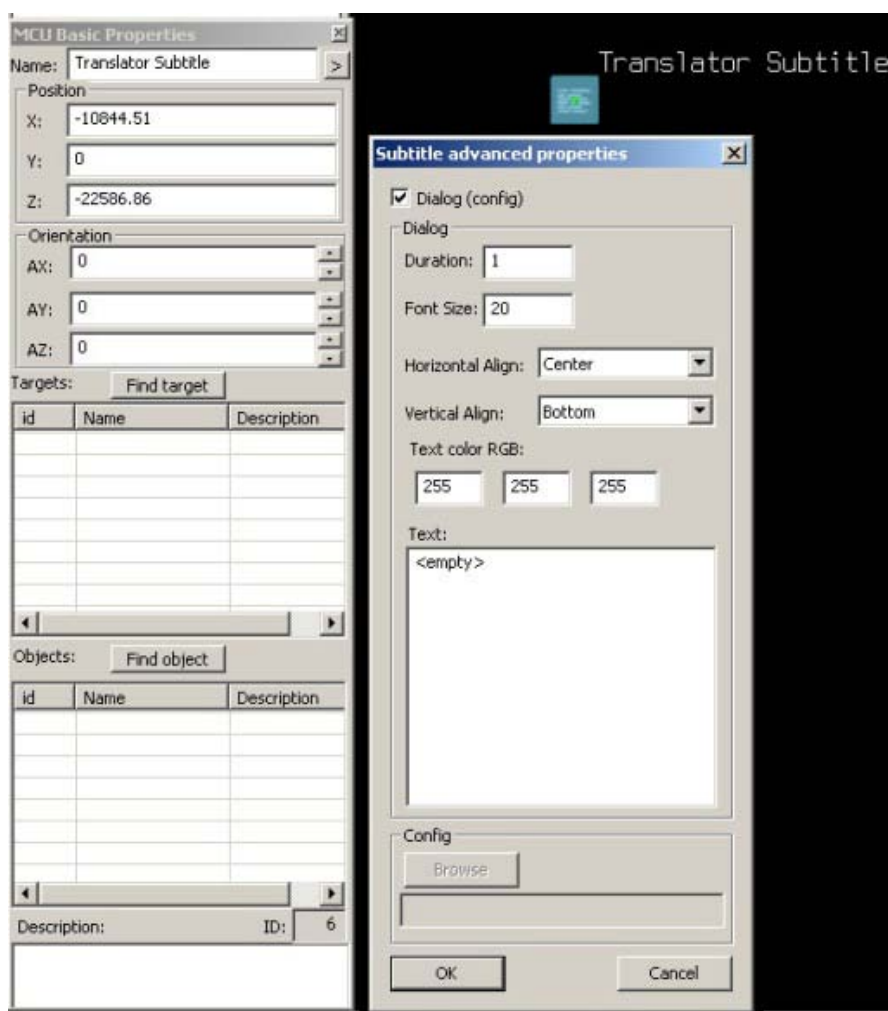
```
{  
  Index = 23; //ID MCU  
  Name = "Translator Mission End"; //имя MCU (не отображается в GUI)  
  Desc = ""; //описание MCU (не отображается в GUI)  
  Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного  
транслятора исходящих целевых связей быть не может  
  Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у  
данного транслятора исходящих объектных связей быть не может  
  XPos = 2050.98; // координата по оси X  
  YPos = 112.394; // координата по оси Y  
  ZPos = 4209.45; // координата по оси Z  
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)  
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)  
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)  
  Enabled = 1; //через интерфейс редактора не используется  
  Succeeded = 1; //через интерфейс редактора не используется, не функционально  
}
```

7.7.8. Translator:Subtitle



На приведенном примере:

- Translator: Subtitle - вставка субтитров в миссию;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Translator: Subtitle;



Позволяет показать субтитры в игре. Даблклик по иконке транслятора открывает меню Subtitle Advanced Properties.

Advanced Properties:

Чекбокс Dialog (config) – должен быть включен, чтобы иметь возможность настройки субтитра (включен по умолчанию).

Поле Dialog.

Duration – время отображения субтитра на мониторе пользователя в секундах.

Font Size – размер шрифта субтитра.

Horizontal Align – расположение субтитра по горизонтали (слева, в центре, справа).

Vertical Align - расположение субтитра по вертикали (сверху, в центре, снизу).

Text Color RGB - определяет цвет субтитра в стандартной палитре RGB, задавая составляющие красного, зеленого и синего цветов.

Text – поле, в котором набирается текст субтитра.

Translator:Subtitle позволяет использовать html теги:

 - перенос на следующую строку;

<u>...</u> - подчеркивание текста;

<i>...</i> - курсив;

... - bold (жирный).

Например: MISSION DESCRIPTION
<u>Date:</u> 03.09.1918.

В окне игры эта надпись будет выглядеть следующим образом:

—

MISSION DESCRIPTION

Date: 03.09.1918.

—

В миссии может быть любое количество Translator:Subtitle.

Для того, чтобы вывести субтитр в миссию – нужно активировать его целевой связью.

При активации Translator:Media типа Stop Media (см. п. 7.7.4) текущий субтитр так же прекратится.

ВНИМАНИЕ!: - на данном этапе развития проекта, механизм игры не управляет временем внутри Flash (весь механизм визуализации построен на этом языке, в том числе и субтитры), поэтому, во время проигрывания Subtitle, если поставить игру на паузу, то Subtitle НЕ УСТАНОВИТСЯ НА ПАУЗУ, а будет работать независимо (по своему внутреннему времени).

Поэтому, если вы используете Subtitle сразу после загрузки миссии, например вступительный текст (а при загрузке игра становится на паузу примерно через 1-2 секунды после загрузки) – помните об этой тонкости и запускайте субтитры через таймер-проставку в ~3 секунды. Для примера использования Subtitle в таком случае смотрите пункт 8.8. с разбором примера использования.

Текстовый формат транслятора:

MCU_TR_Subtitle //тип MCU

{

Index = 24; //ID MCU

Name = "Translator Subtitle"; //имя MCU (не отображается в GUI)

Desc = ""; //описание MCU (не отображается в GUI)

Targets = []; // ID объектов, на которые целевой связью ссылается данный транслятор, у данного транслятора исходящих целевых связей быть не может

Objects = []; // ID объектов, на которые объектной связью ссылается данный транслятор, у данного транслятора исходящих объектных связей быть не может

XPos = 2158.22; // координата по оси X

YPos = 115.53; // координата по оси Y

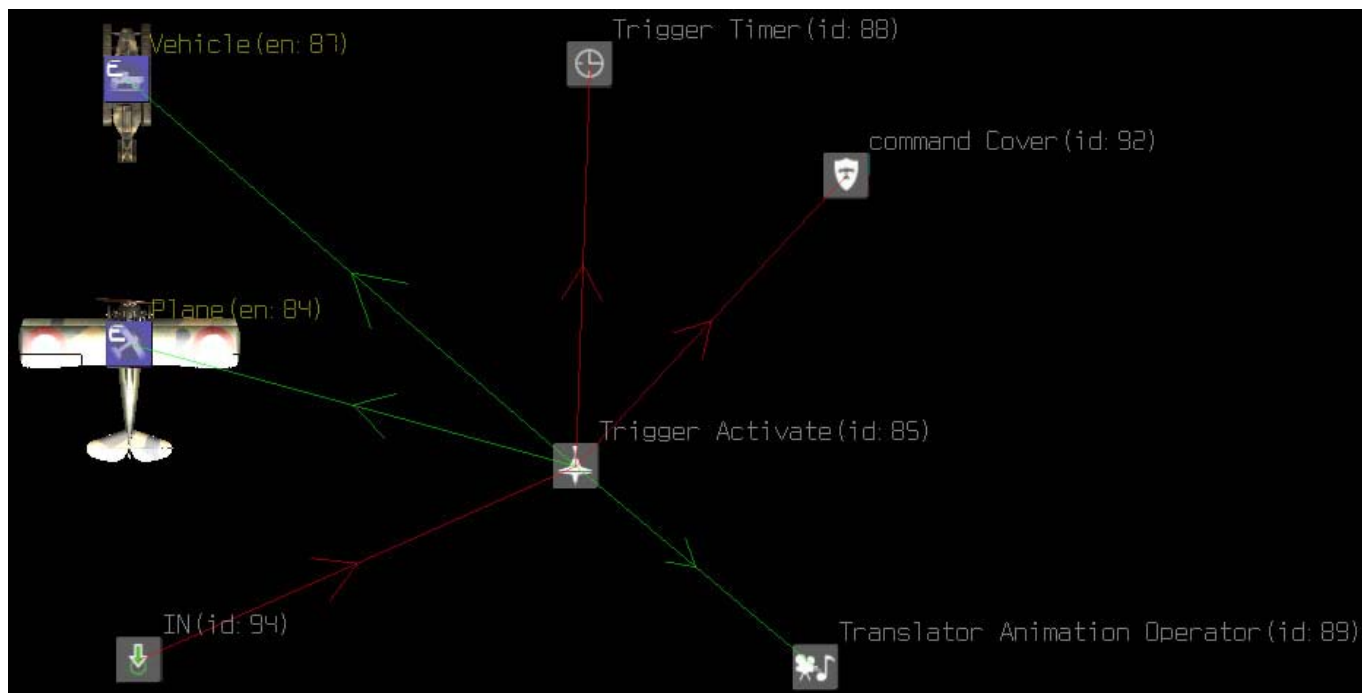
```

ZPos = 4194.55; // координата по оси Z
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
Enabled = 1; //через интерфейс редактора не используется
OnMessages //не используется
{
}
Dialog
{
    Duration = 1; //длительность субтитры
    FontSize = 20; //размер шрифта субтитры
    HAlign = 1; //расположение субтитры по горизонтали 0=слева, 1=по центру, 2=справа
    VAlign = 2; //расположение субтитры по вертикали 0=сверху, 1=по центру, 2=снизу
    RColor = 255; //красная составляющая цвета текста субтитры
    GColor = 255; //зеленая составляющая цвета текста субтитры
    BColor = 255; //синяя составляющая цвета текста субтитры
    LCText = 3; //индекс строки в файле локализации
}
}

```

7.8. MCU Trigger.

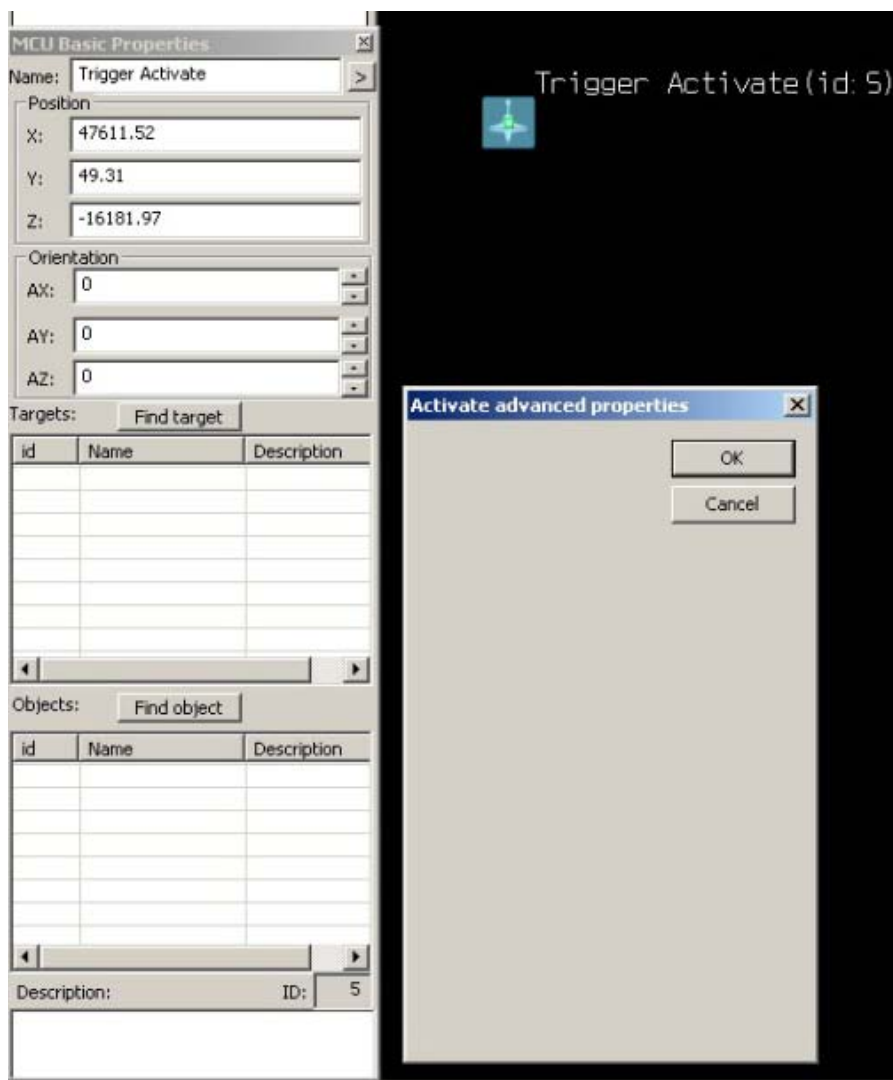
7.8.1. Trigger:Activate



На приведенном примере:

- Trigger:Activate – триггер активирующий объекты миссии, если первоначально они были деактивированы, или они были не активны (см.п.7.2.1.), объектными связями - самолеты/наземную технику и трансляторы, целевыми связями – триггеры/команды
- IN - любое исходящее из миссии событие, запускающее Trigger:Activate.

Важно помнить, что не активированный объект не воспринимает команды из миссии и не взаимодействует с объектами миссии.



Триггер, активирующий ранее деактивированные объекты или MCU миссии.

Даблклик по иконке триггера открывает меню Activate Advanced Properties (none).

Данный триггер часто используется в связке с триггером Deactivate (см.п. 7.8.5.).

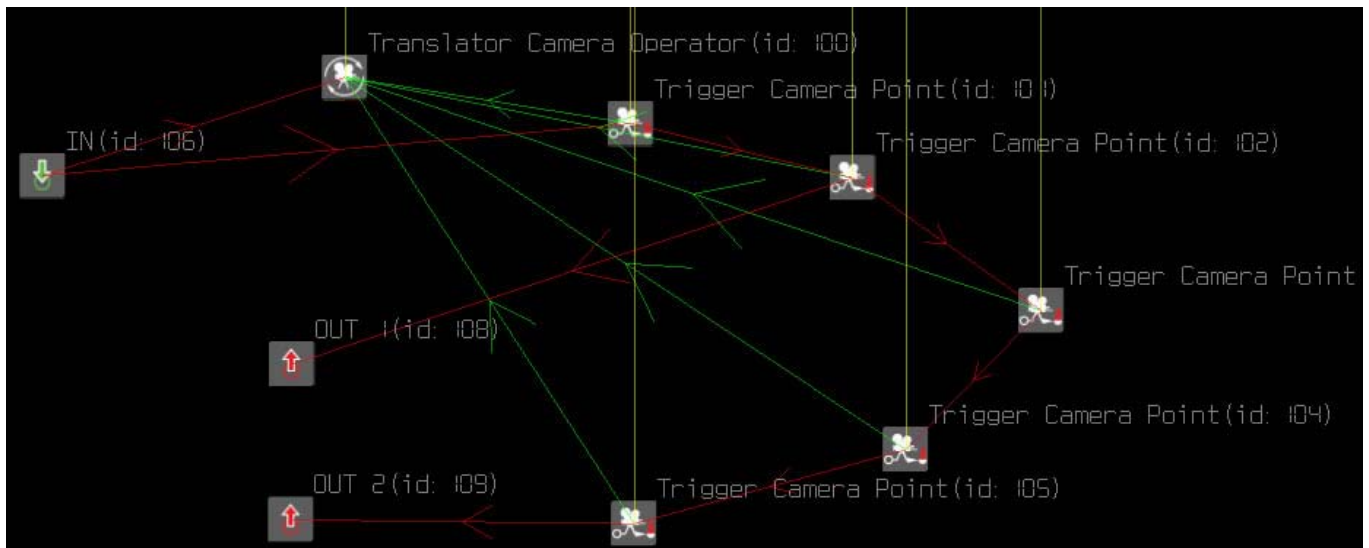
Текстовый формат триггера:

```
MCU_Activate //тип MCU
{
  Index = 25; //ID MCU
  Name = "Trigger Activate"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2107.58; // координата по оси X
  YPos = 112.755; // координата по оси Y
  ZPos = 4292.85; // координата по оси Z
  XOri = 0;
  YOri = 0;
  ZOri = 0;
}
```

7.8.2. Trigger:Camera point



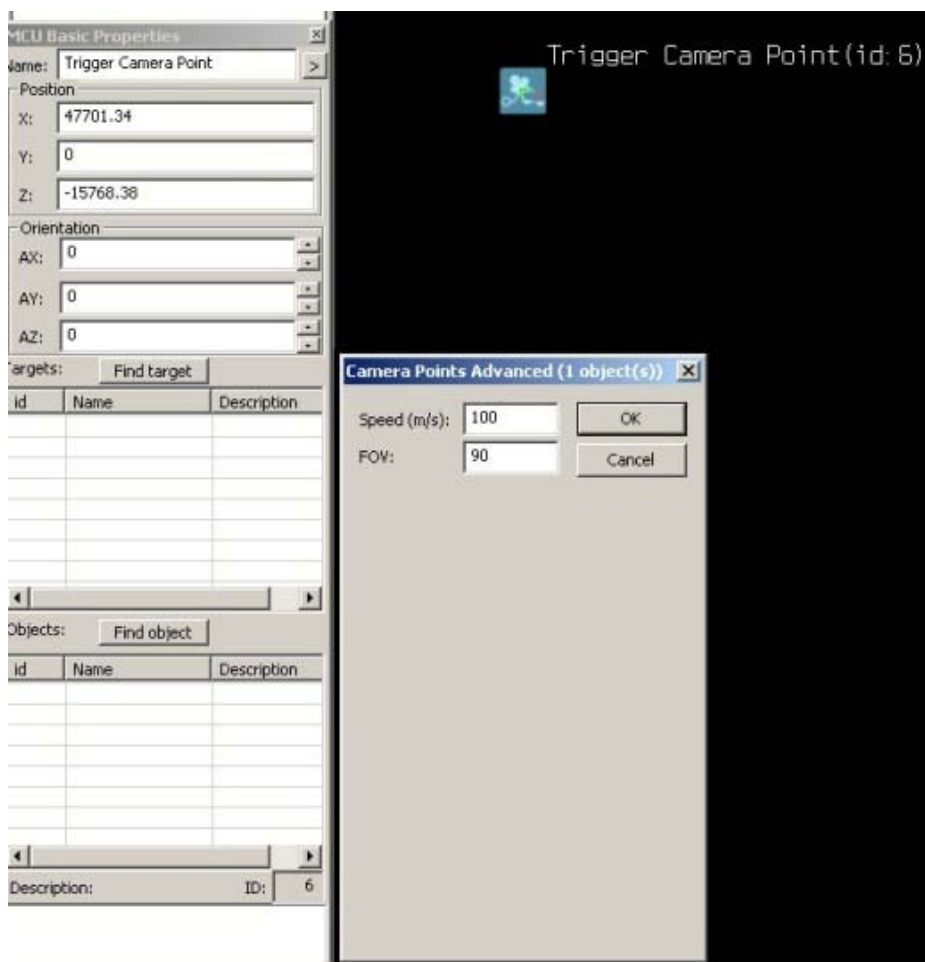
Смотри пункт 7.7.2.



На приведенном примере:

- Trigger:Camera point – триггер контрольной точки движения камеры, сам по себе Trigger:Camera point не имеет смысла т.к. по сути он является подчиненным объектом Translator:Camera Operator и поэтому привязан к нему объектной связью, т.е. Camera Operator задает свойства камеры и её тип в миссии, а Camera point задает характер движения камеры в миссии;
- IN - любое исходящее из миссии событие, запускающее Trigger:Camera point и Translator:Camera Operator;
- OUT 1 & OUT 2 - Camera point является триггером, поэтому при своем срабатывании он может «дергать» при помощи целевой связи другие события миссии.

При движении камеры по точкам, FOV будет плавно изменяться от точки к точке, в зависимости от настроек в каждой Camera Point.



Триггер, срабатывающий по камере (по камере как движущемуся объекту), и одновременно является командой камере двигаться в заданную точку с заданной скоростью, а также занять положение и ориентацию, указанную в Basic Properties триггера.

Даблклик по иконке триггера открывает меню Camera Points Advanced Properties.

Camera Point Advanced Properties:

Поле Speed (m/s): - указывает с какой скоростью (в метрах/секунду) камера должна перемещаться к точке.

Поле FOV: - поле зрения камеры.

Скорость перемещения камеры не должна превышать 200 м/с!

Текстовый формат триггера:

MCU_CameraPoint //тип MCU

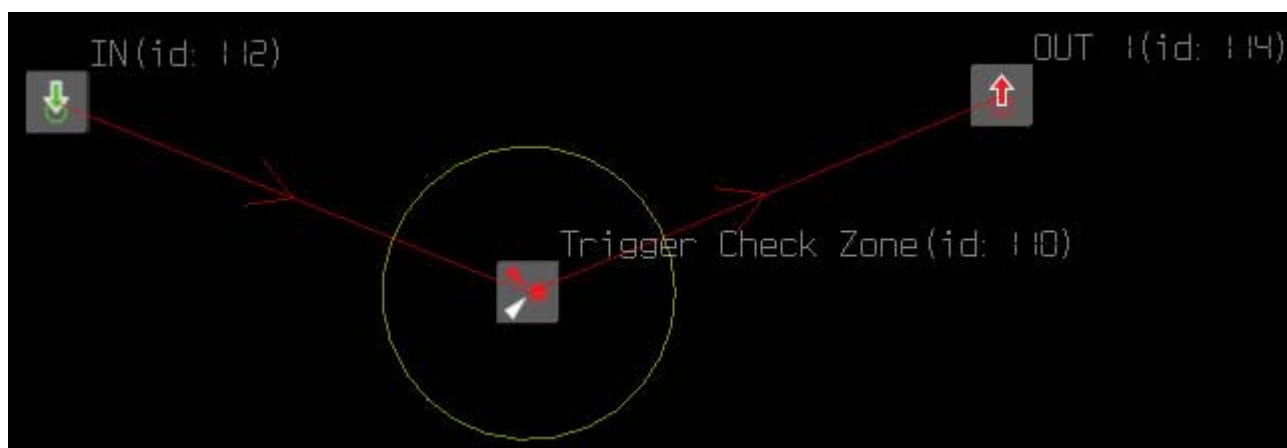
```
{
  Index = 26; //ID MCU
  Name = "Trigger Camera Point"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2104.6; // координата по оси X
  YPos = 113.408; // координата по оси Y
  ZPos = 4367.32; // координата по оси Z
```

```

XOri = 0; // ориентация по оси X – наклон (крен) камеры в горизонтальной плоскости
YOri = 0; // ориентация по оси Y – поворот камеры в горизонтальной плоскости
ZOri = 0; //ориентация по оси Z – наклон камеры в вертикальной плоскости
Speed = 100; //скорость камеры
FOV = 90; //ширина угла зрения камеры
}

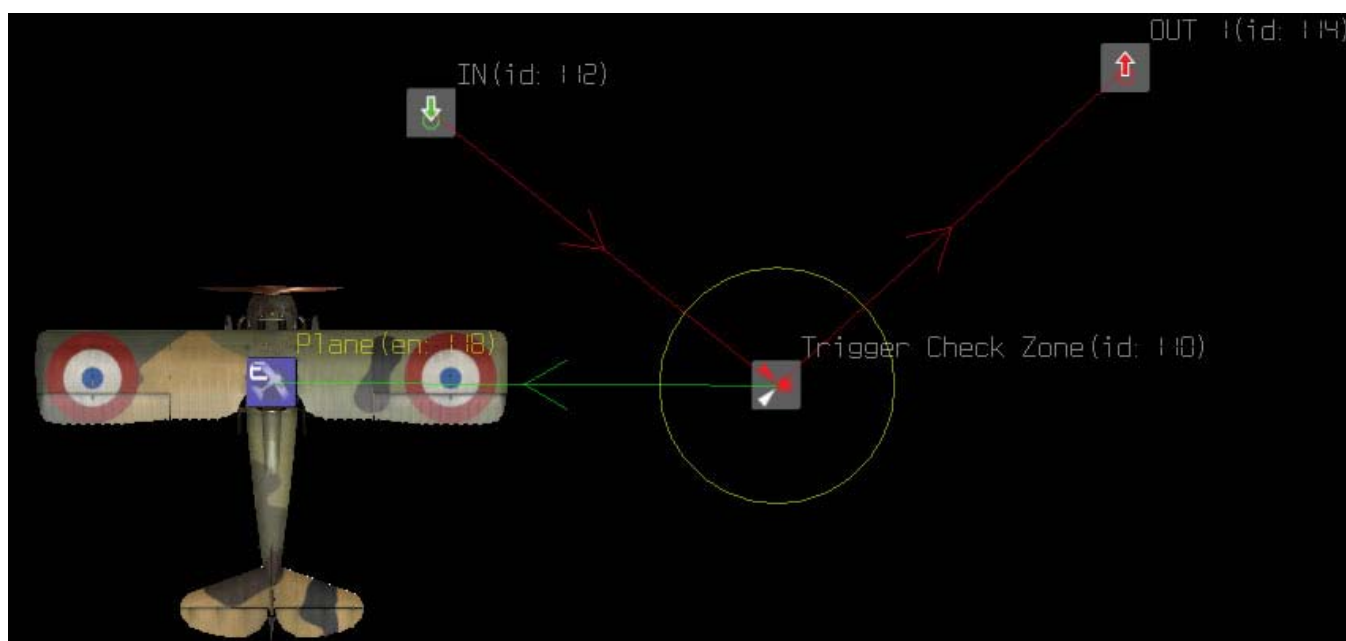
```

7.8.3. Trigger:Check zone



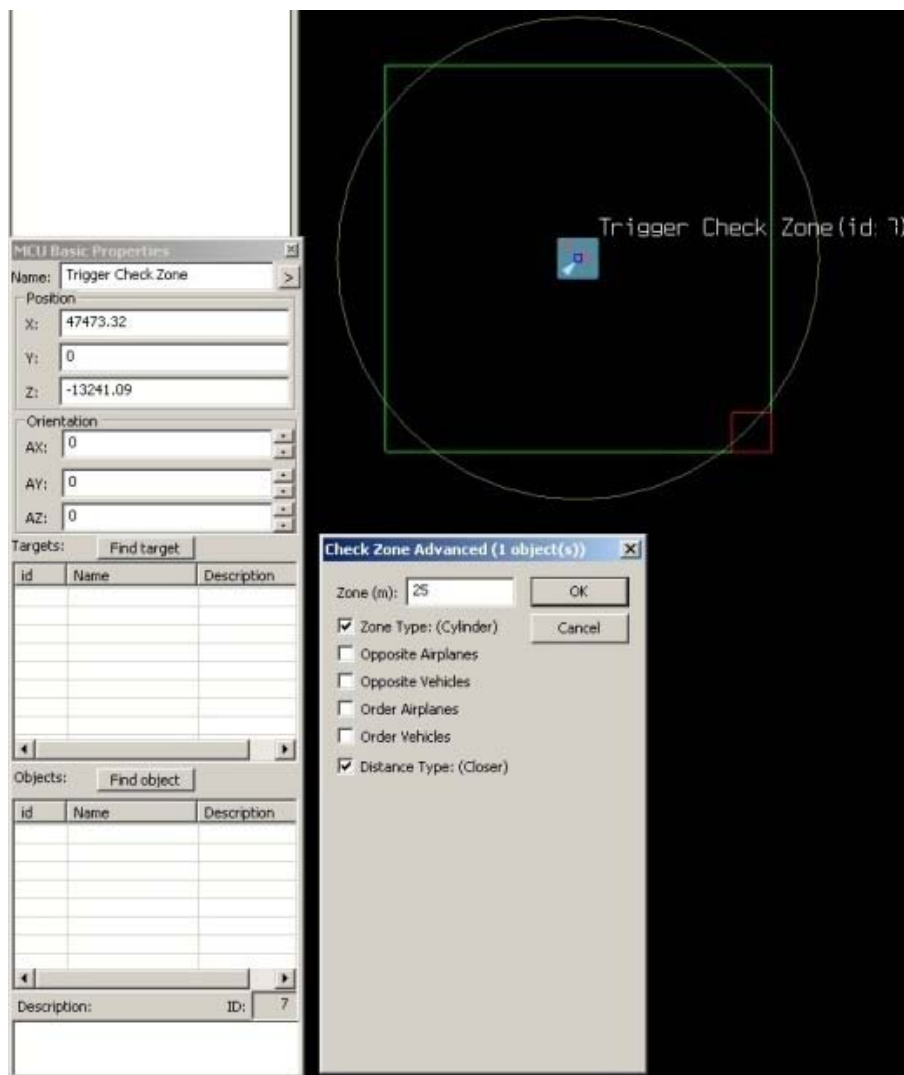
На приведенном примере (условие срабатывания установлены в Properties):

- Trigger: Check zone – триггер проверки нахождения объектов заданного типа внутри или вне заданной зоны (зона обозначена радиусом желтого цвета);
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Trigger:Check zone;
- OUT – любое событие миссии, которое активируется триггером Check zone после его срабатывания.



На приведенном примере (условие срабатывания по прилинкованному объекту):

- Trigger: Check zone – триггер проверки нахождения объекта (в данном примере - самолета) внутри или вне зоны, триггер объектной связью ссылается на самолет и срабатывает только по нему;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Trigger:Check zone;
- OUT – любое событие миссии, которое активируется триггером Check zone после его срабатывания



Два режима работы: проверка присутствия объектов внутри заданного радиуса действия (close) и вне заданного радиуса действия (further).

Даблклик по иконке триггера открывает меню Check Zone Advanced Properties.

Advanced Properties:

Поле Zone (m): - указывает величину радиуса зоны срабатывания триггера в метрах.

Чекбокс Zone Type:Cylinder (Sphere) – устанавливает форму зоны в виде цилиндра (когда включен) или сферы (в выключенном состоянии). При установке опции Sphere не забывайте устанавливать высоту триггера (YPos).

Чекбокс Opposite Airplanes – срабатывание (вкл.)/не срабатывание (выкл.) триггера на самолеты Opposite Coalition.

Чекбокс Opposite Vehicles – срабатывание (вкл.)/не срабатывание (выкл.) триггера на наземную технику Opposite Coalition.

Чекбокс Order Airplanes – срабатывание (вкл.)/не срабатывание (выкл.) триггера на самолеты Order Coalition.

Чекбокс Order Vehicles – срабатывание (вкл.)/не срабатывание (выкл.) триггера на наземную технику Order Coalition.

Чекбокс Distance Type:(Closer) – включение этой опции означает, что триггер работает при приближении объекта к центру зоны на расстояние, указанное в поле Distance.

Чекбокс Distance Type:(Further) – включение этой опции означает, что триггер работает при удалении объекта от центра зоны на расстояние, указанное в поле Distance.

Не забывайте «включать» триггер с помощью целевой связи от какого либо события (Mission Begin, пролет Way Point и т.д.), иначе триггер не будет работать!

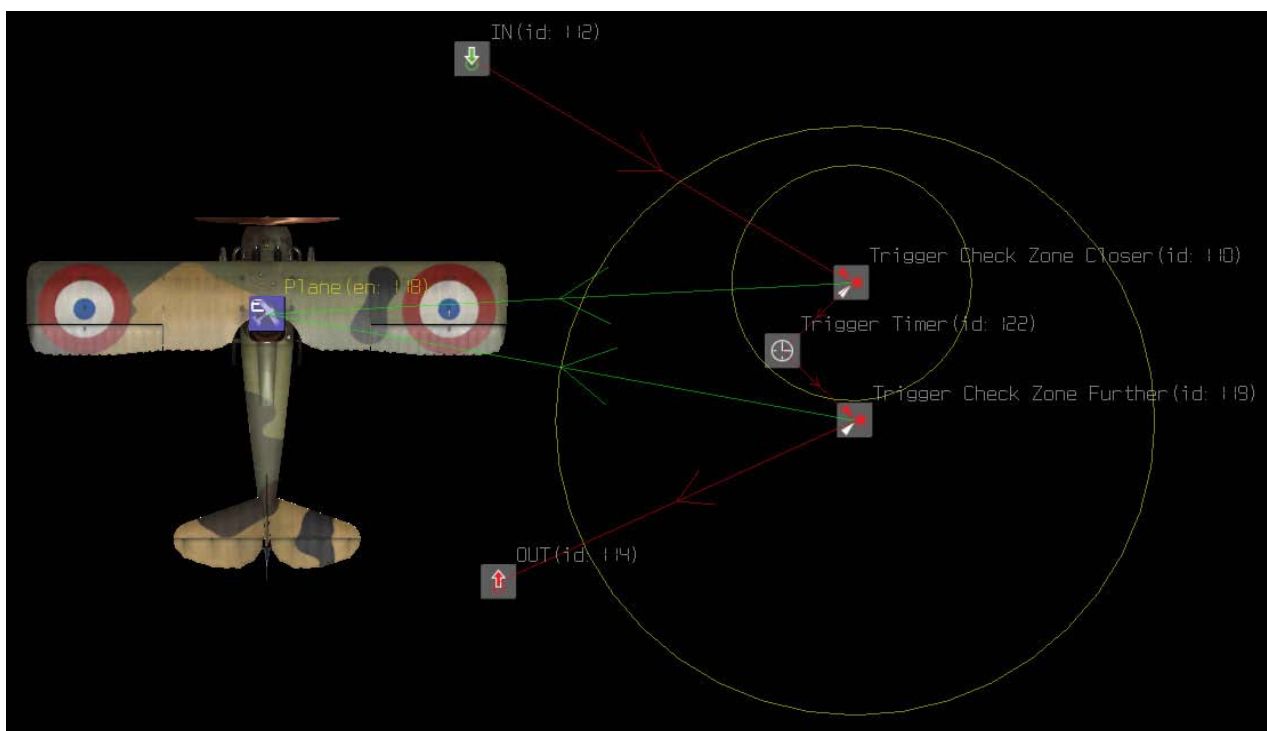
В случае, если для одного и того же объекта производится проверка closer и further, должен присутствовать гистерезис (разница в размере радиуса close и further).

Причем, сначала включается проверка closer, а уже после того, как сработала эта проверка – можно через таймер или по событию включать проверку further, иначе логика проверок будет нарушена.

Например: если проверяемый объект «рождается» внутри заданного радиуса Check Zone (further) – все работает корректно, т.е. проверка не включится, т.к. по факту объект не находится снаружи заданного радиуса зоны и Check Zone (close) в этом случае не нужна.

Если проверяемый объект «рождается» снаружи заданного радиуса Check Zone (further), то сначала включите проверку close для того, чтобы триггер «понял», что объект влетел в зону проверки, а уже после этого включайте further, чтобы проконтролировать «вылет» из заданной зоны.

В случае активации деактивированной (триггером Deactivate) Check Zone, она работает повторно по условиям, заданным в Advanced Properties.



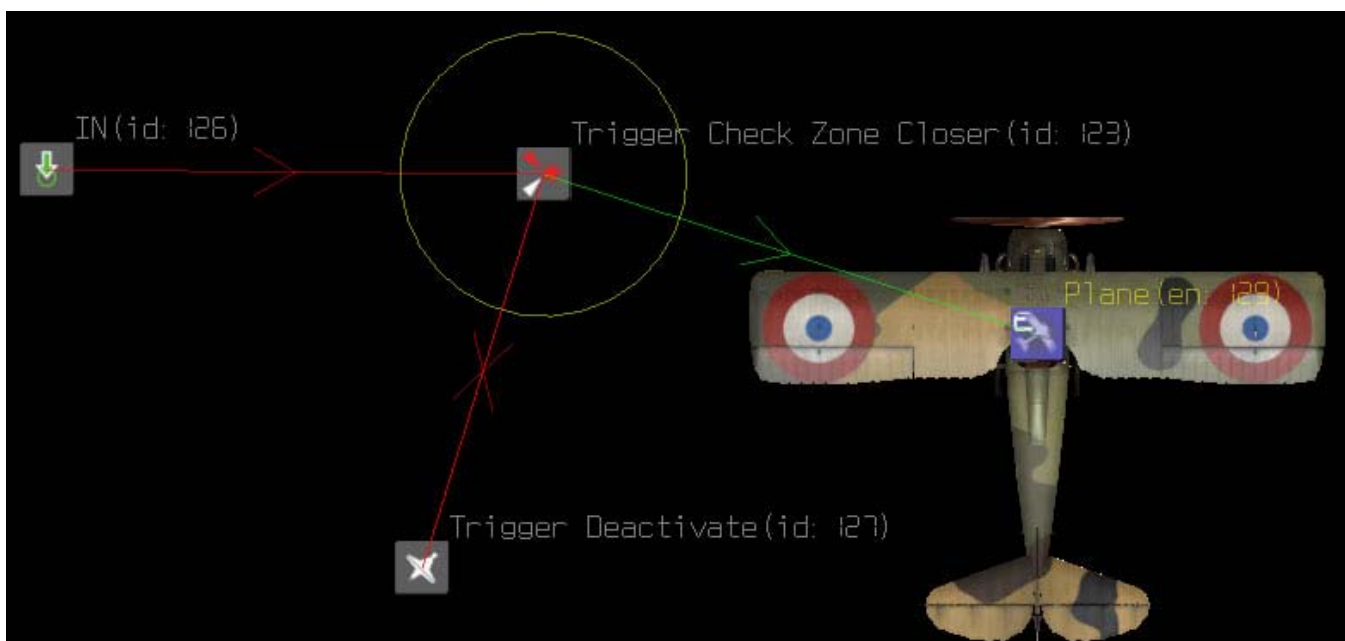
На приведенном примере (условие срабатывания по прилинкованному объекту):

- Trigger: Check zone Closer – триггер проверки нахождения объекта (в данном примере - самолета) внутри зоны, триггер объектной связью ссылается на самолет и срабатывает только по нему;
- Trigger: Check zone Further – триггер проверки нахождения объекта (в данном примере - самолета) вне зоны, триггер объектной связью ссылается на самолет и срабатывает только по нему;
- Trigger Timer – триггер, который запускает Check zone Further после срабатывания Check zone Closer;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Trigger:Check zone;
- OUT – любое событие миссии, которое активируется триггером Check zone Further после его срабатывания.

Таким образом, вышеприведенный пример осуществляет проверку по самолету, который влетел в заданную зону и вылетел из неё, после чего OUT может активировать любое событие миссии.

Так же обратите внимание на то, что после «запуска» триггера Check Zone он активен на протяжении всего времени миссии.

Обычно, требуется одноразовое срабатывание данного триггера (если не нужна постоянная проверка зоны) по объекту, поэтому советуем вам деактивировать неиспользуемые миссией триггеры Check Zone:



То есть, в данном примере Check Zone Closer сработает по заданному объекту один раз и сама себя деактивирует при помощи триггера Deactivate (см.п.7.8.5.). Практически это требуется для того, чтобы в миссии не оставалось «хвостов» из ненужных активных Check Zone, которые могут случайно повторно сработать (опять же – если это не требуется сюжетом миссии).

Триггер имеет ФИКСИРОВАННЫЕ координаты в миссии в отличие от триггера Proximity (см.п. 7.8.6.).

Текстовый формат триггера:

MCU_CheckZone //тип MCU

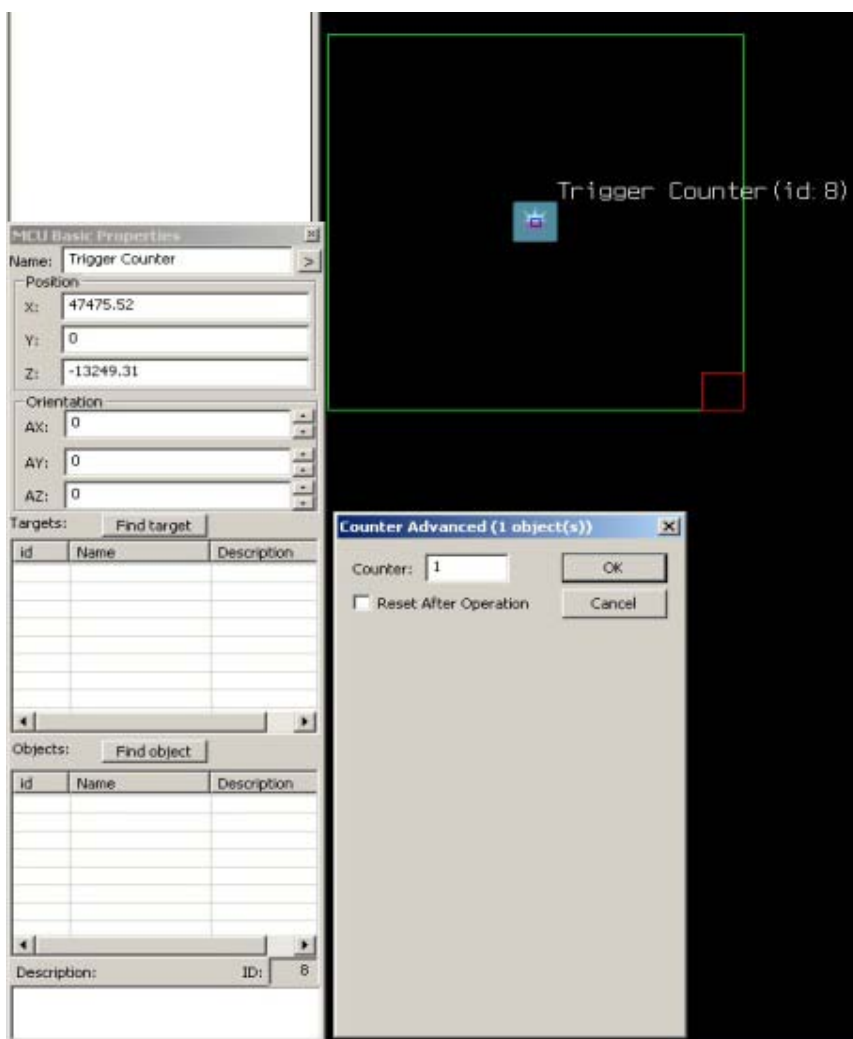
```
{
  Index = 27; //ID MCU
  Name = "Trigger Check Zone"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2095.66; // координата по оси X
  YPos = 112.102; // координата по оси Y
  ZPos = 4307.74; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Zone = 1; //радиус срабатывания зоны в метрах
  Cylinder = 1; //вид зоны, 1=цилиндр, 0=сфера
  OppositeAirplanes = 0; //срабатывание триггера по Opposite самолетам, 0=выкл/1=вкл
  OppositeVehicles = 0; //срабатывание триггера по Opposite наземной технике, 0=выкл/1=вкл
  OrderAirplanes = 0; //срабатывание триггера по Order самолетам, 0=выкл/1=вкл
  OrderVehicles = 0; //срабатывание триггера по Order наземной технике, 0=выкл/1=вкл
  Closer = 1; //типа зоны, 1=closer, 0=further
}
```

7.8.4. Trigger:Counter



На приведенном примере:

- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Trigger:Counter, т.е. каждый раз, когда приходит данное событие, Counter его считает (+1);
- OUT – любое событие миссии, которое активируется триггером Counter после его срабатывания.



Триггер-счетчик событий (при каждом срабатывании счетчик увеличивается на единицу).

Даблклик по иконке триггера открывает меню Counter Advanced Properties.

Advanced Properties:

Поле Counter: - пороговое значение, при достижении которого счетчик активирует свои целевые связи.

Чекбокс Reset After Operation – сбрасывает значение счетчика при достижении порогового значения. Данная опция обычно используется, если нужно повторное использование условий установленных в данном триггере.

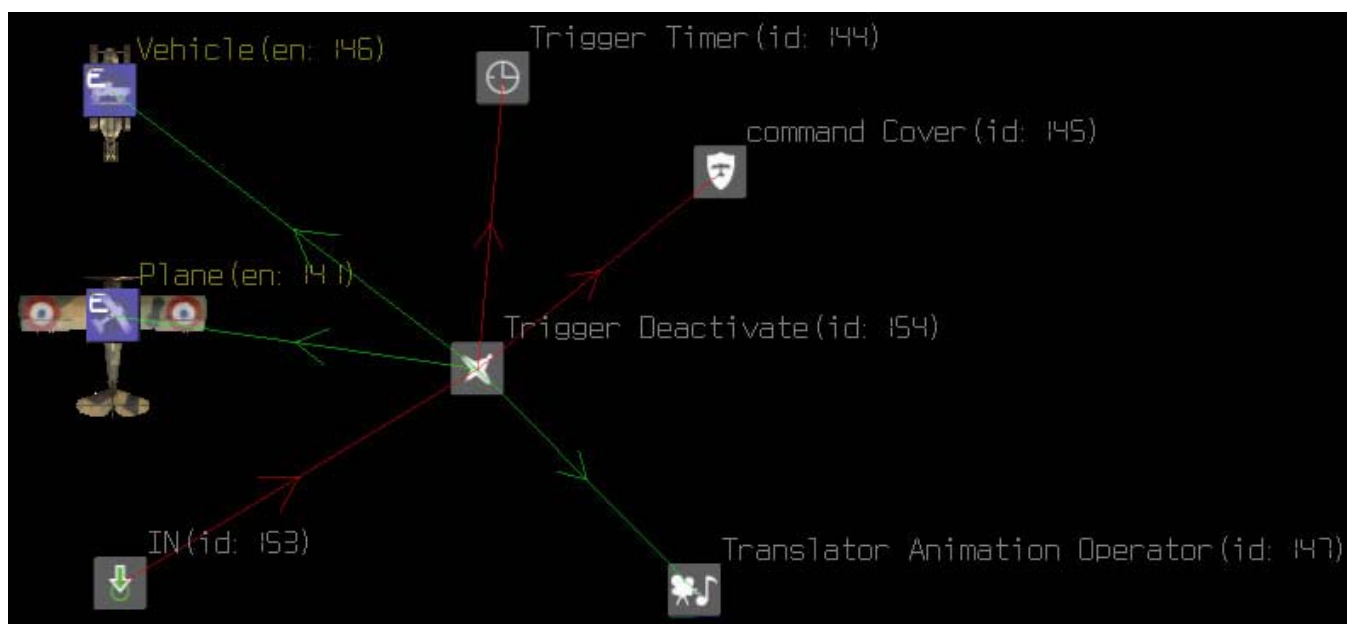
В случае деактивации этого триггера через триггер Deactivate (см.п.7.8.5.), все просчитанные события не сбрасываются, а «замораживаются» и остаются в «памяти» триггера. Если его активировать повторно, триггер продолжить считать события с учетом данных, которые остались в его «памяти» и сработает по достижении порогового значения.

Текстовый формат триггера:

MCU_Counter //тип MCU

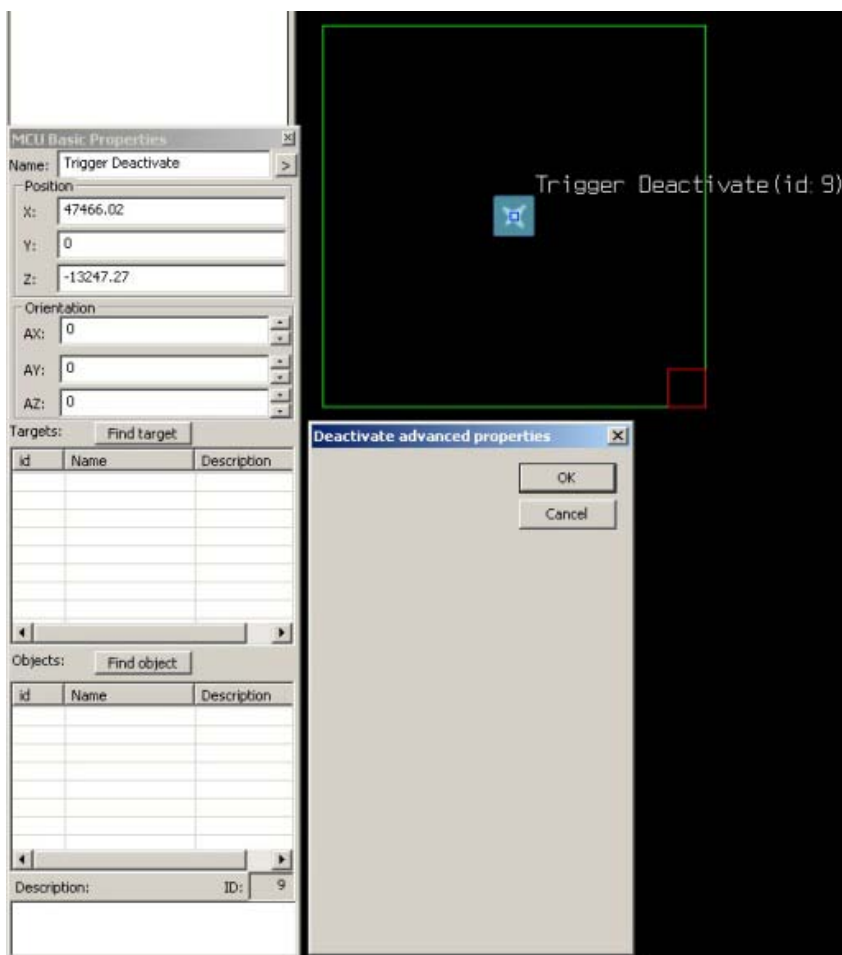
```
{
  Index = 28; //ID MCU
  Name = "Trigger Counter"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер, у данного
триггера исходящих объектных связей быть не может
  XPos = 2080.77; // координата по оси X
  YPos = 114.026; // координата по оси Y
  ZPos = 4170.72; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Counter = 1; //значение порога срабатывания триггера
  Dropcount = 0; //сбрасываемый (1) /не сбрасываемый (0) в ноль после срабатывания триггер
}
```

7.8.5. Trigger:Deactivate



На приведенном примере:

- Trigger:Deactivate – триггер, деактивирующий объекты миссии объектными связями - самолеты/наземную технику и трансляторы, целевыми связями – триггеры/команды;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает Trigger:Deactivate.



Даблклик по иконке триггера открывает меню Deactivate Advanced Properties.

Данный триггер часто используется в связке с триггером Activate (см.п.7.8.1).

БУДЬТЕ ВНИМАТЕЛЬНЫ – целевая связь данного триггера **ДЕАКТИВИРУЕТ** объекты, т.е. при помощи целевой связи от данного триггера **НЕЛЬЗЯ** «дернуть» последующие события в миссии.

БУДЬТЕ ВНИМАТЕЛЬНЫ – если объект деактивирован, то он не воспринимает команды из миссии и не взаимодействует с другими объектами в миссии.

Данный триггер по разному воздействует на MCU и объекты в миссии:

- при деактивации самолетов, танков, автомобилей, паровозов, артиллерии и аэростатов, данные объекты остаются в миссии в НЕ активном состоянии и не воспринимают воздействие миссии до момента, пока не будут активированы;
- объекты из раздела Blocks и Battlefield при деактивации остаются видимыми в игре;
- при деактивации триггера Timer (см.п.7.8.8.) значение времени внутри него сбросится в исходное состояние;
- при деактивации триггера Check Zone и его последующей активации, триггер сработает повторно (см.п.7.8.3.) по установленным внутри него условиям;
- при деактивации триггера Counter текущее значение «заморозится»;

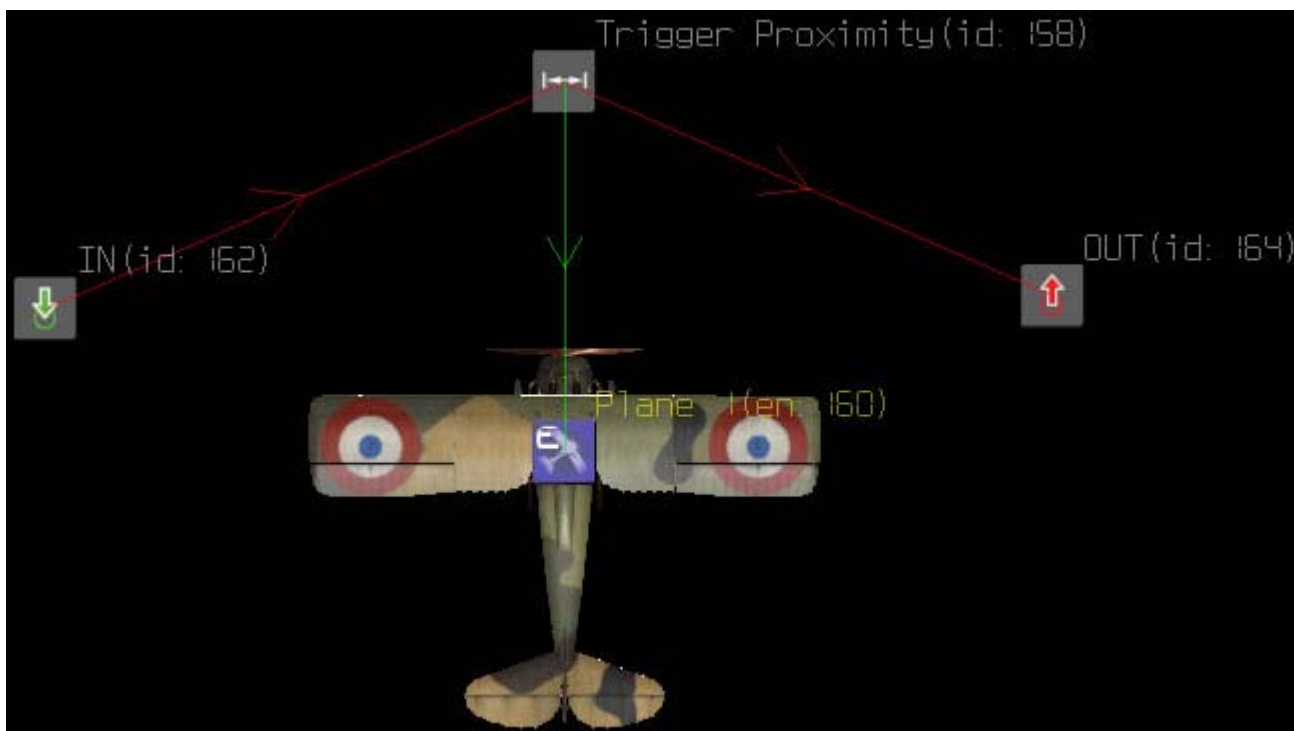
Текстовый формат триггера:

```
MCU_Deactivate //тип MCU
{
  Index = 29; //ID MCU
  Name = "Trigger Deactivate"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2021.2; // координата по оси X
  YPos = 110.762; // координата по оси Y
  ZPos = 4149.87; // координата по оси Z
  XOri = 0;
  YOri = 0;
  ZOri = 0;
}
```

7.8.6. Trigger:Proximity



Пример по одному объекту и заданными в Properties объектами:

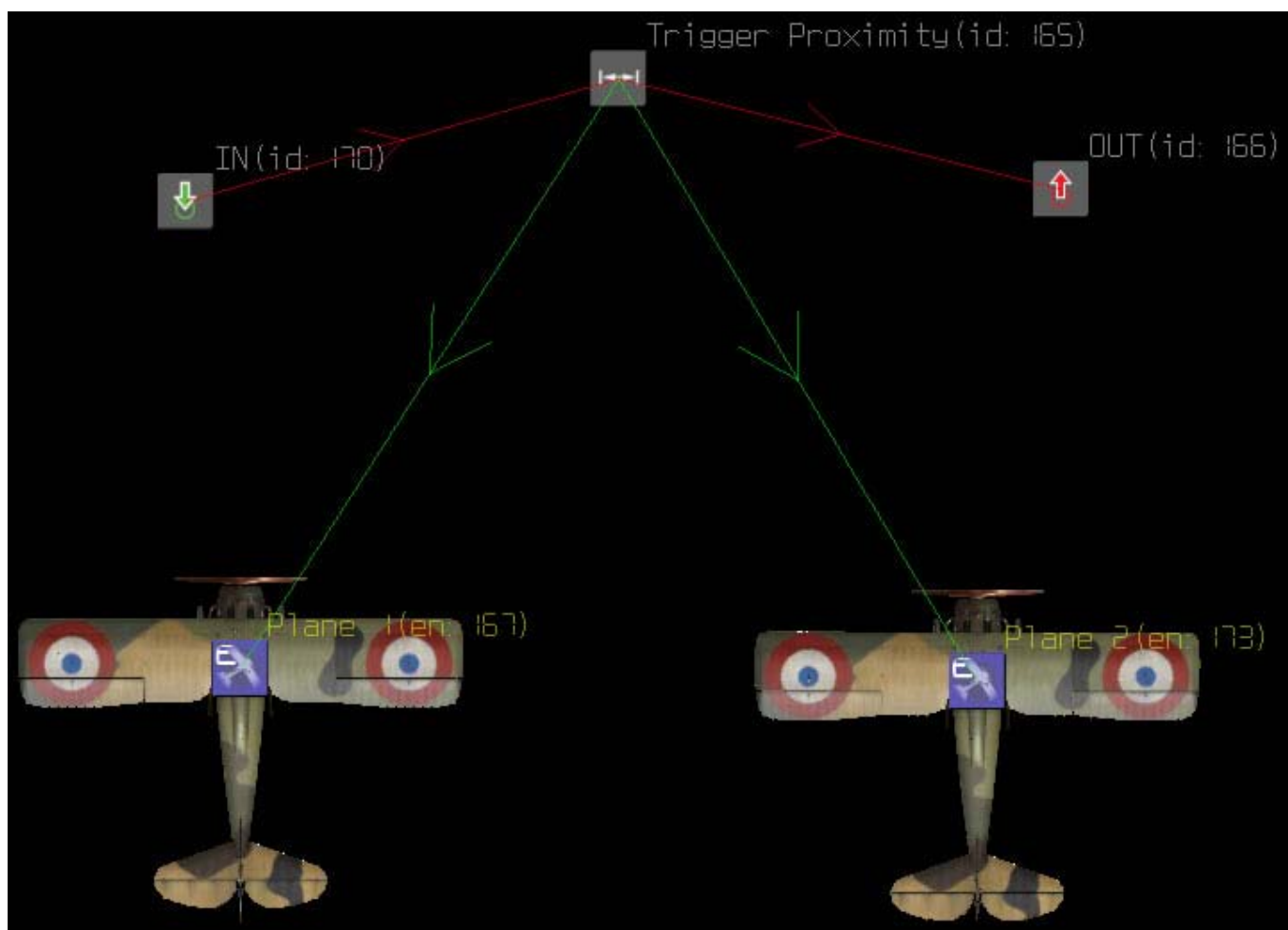


На приведенном примере:

- Trigger: Proximity – триггер, проверяющий условие наличия ближе/дальше заданной дистанции в Properties объектов;
- IN - любое исходящее из миссии событие, запускающее триггер Proximity .

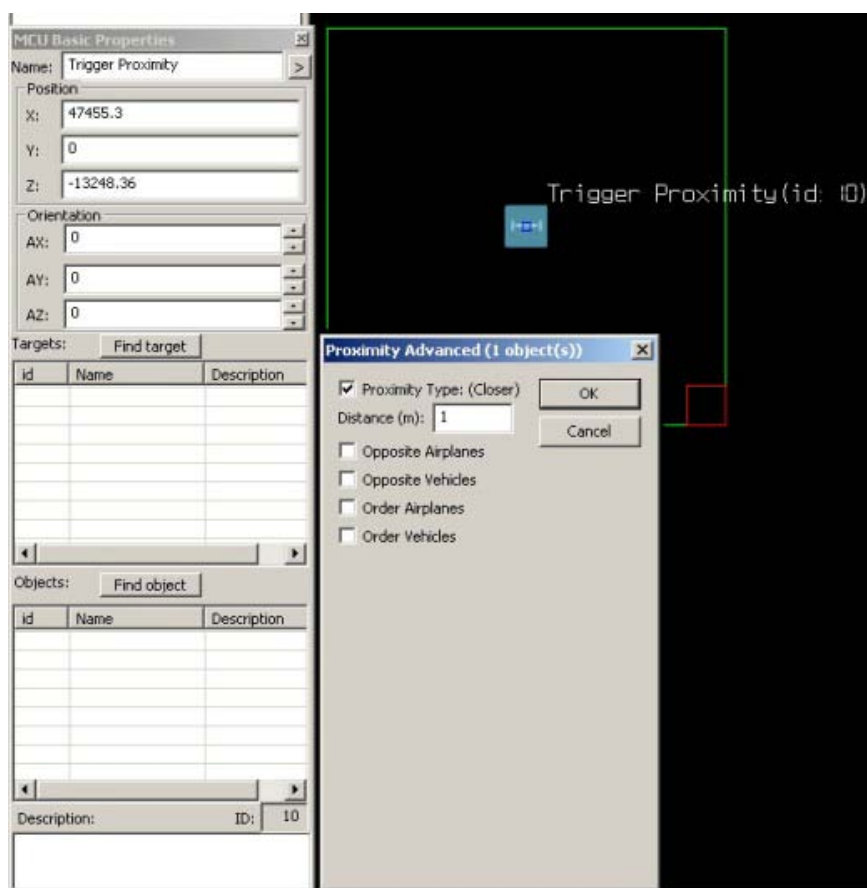
- OUT – любое событие миссии, которое активируется триггером Proximity после его срабатывания.

Пример Proximity между двумя заданными объектами:



На приведенном примере:

- Trigger: Proximity – триггер проверяющий условие по дистанции (ближе или дальше) заданных объектов, в данном случае самолетов Plane 1 и Plane 2;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает триггер Proximity .
- OUT – любое событие миссии, которое активируется триггером Proximity после его срабатывания.



Триггер динамического измерения расстояния между назначенными через объектную связь объектами, как на приближение друг к другу, так и на удаление друг от друга. Два режима работы: измерение расстояния на сближение объектов между собой (closer) и измерение расстояния на удаление объектов друг от друга (further). Даблклик по иконке триггера открывает меню Proximity Advanced Properties.

Advanced Properties:

Чекбокс Proximity Type: (Closer) – триггер сработает при сближении объектов на расстояние, равное или меньше указанному в поле Distance.

Чекбокс Proximity Type: (Further) – триггер сработает при отдалении объектов на расстояние, равное или больше указанному в поле Distance.

Поле Distance (m): - указывает радиус действия триггера в метрах.

Чекбокс Opposite Airplanes – срабатывание (вкл.)/не срабатывание (выкл.) триггера на самолеты Opposite Coalition.

Чекбокс Opposite Vehicles – срабатывание (вкл.)/не срабатывание (выкл.) триггера на наземную технику Opposite Coalition.

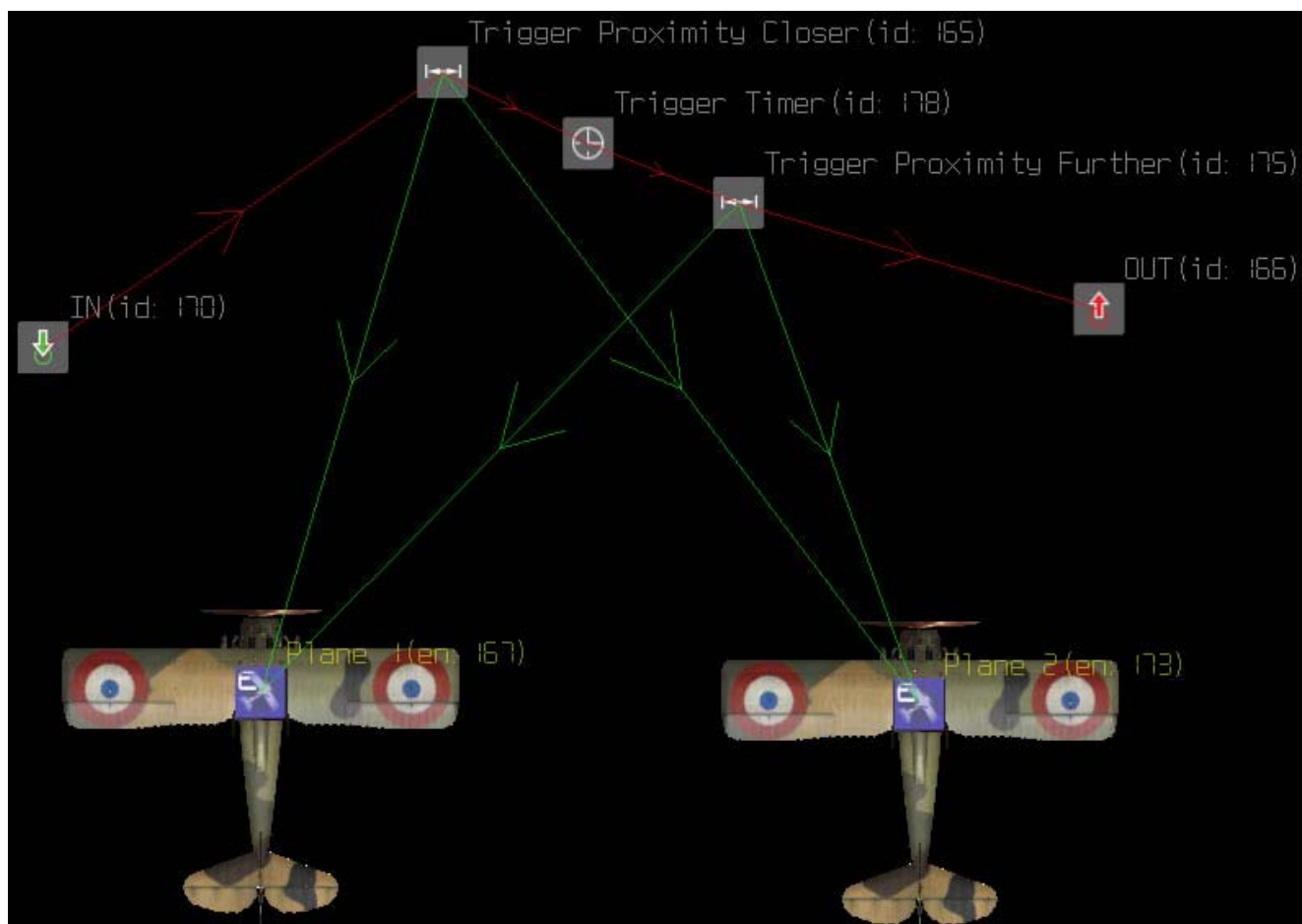
Чекбокс Order Airplanes – срабатывание (вкл.)/не срабатывание (выкл.) триггера на самолеты Order Coalition.

Чекбокс Order Vehicles – срабатывание (вкл.)/не срабатывание (выкл.) триггера на наземную технику Order Coalition.

В случае, если для объектов производится измерение расстояния и по closer и по further, то после срабатывания closer включение further производится через 1-2 секундный триггер Timer.

Данную «проставку» в виде таймера нужно делать т.к. Proximity не имеет определенных координат и объекты в миссии могут находиться где угодно.

То есть, Proximity Further может сработать раньше чем Proximity Closer (т.к. Further изначально проверяет дистанцию между объектами по удалению, но при инициализации в миссии в первый момент имеет координаты 0;0), и для того, чтобы Proximity Further успел измерить заданное между объектами расстояние (успел проанализировать дистанцию между объектами с учетом их координат в мире), по условию further, нужна задержка в виде данного таймера.



Текстовый формат триггера:

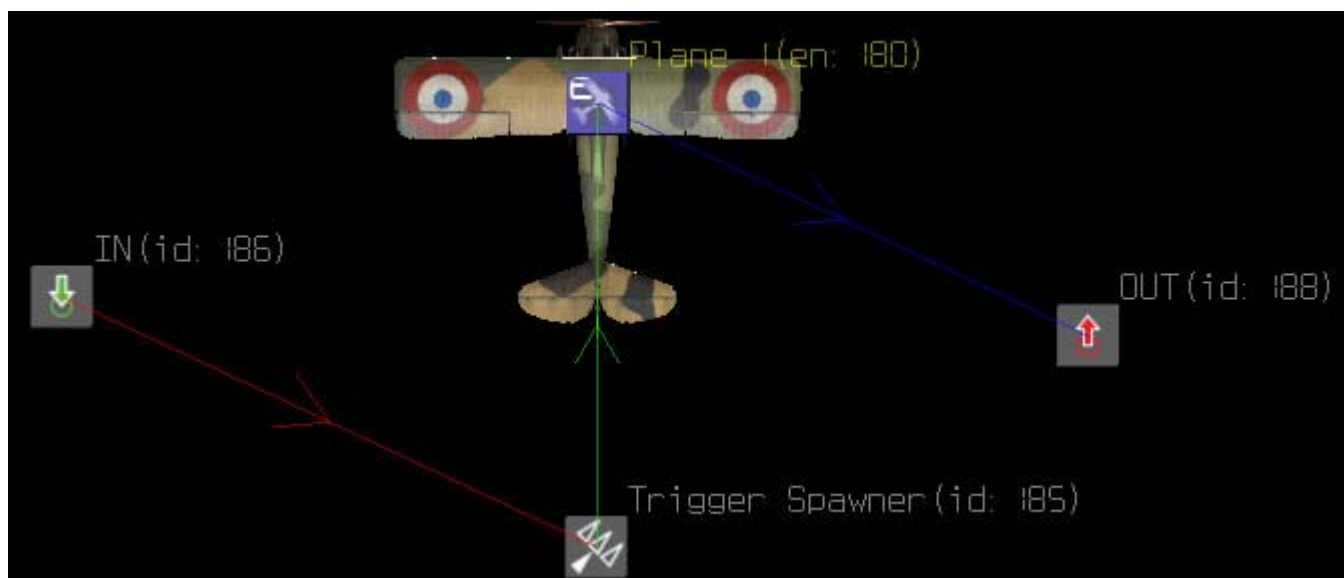
```
MCU_Proximity //тип MCU
{
  Index = 2; //ID MCU
  Name = "Trigger Proximity"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2177.36; // координата по оси X
  YPos = 112.34; // координата по оси Y
  ZPos = 4197.41; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
```

```

ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
Distance = 1; //установка дистанции для срабатывания триггера
OppositeAirplanes = 0; //измеряет дистанцию относительно Opposite самолетов
OppositeVehicles = 0; //измеряет дистанцию относительно Opposite наземной техники
OrderAirplanes = 0; //измеряет дистанцию относительно Order самолетов
OrderVehicles = 0; //измеряет дистанцию относительно Order наземной техники
Closer = 1; //тип триггера, 1=closer, 0=further
}

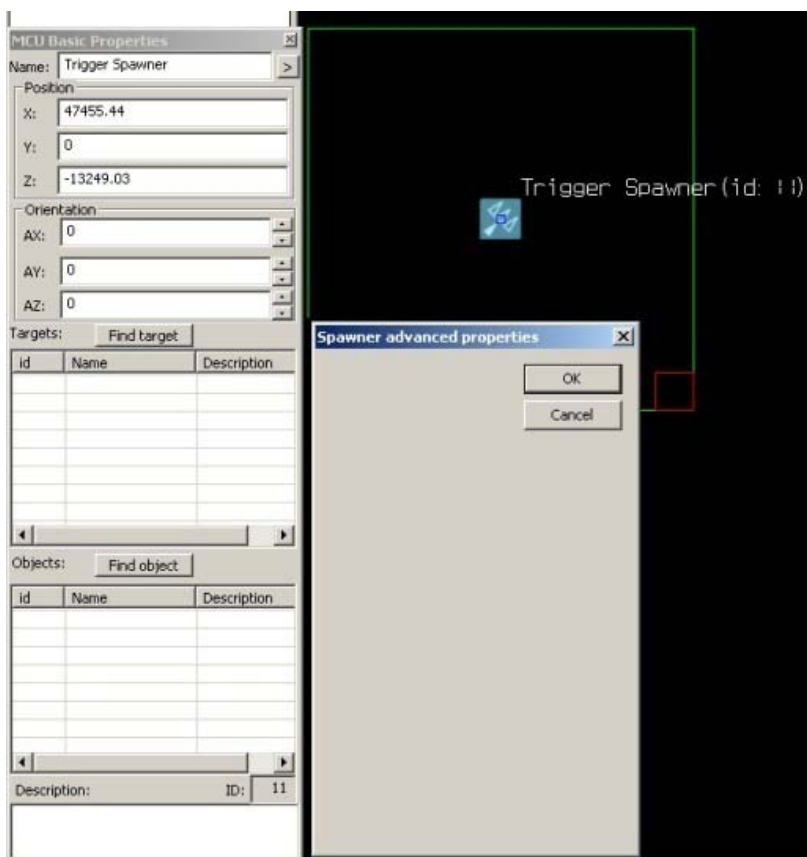
```

7.8.7. Trigger:Spawner



На приведенном примере:

- Trigger:Spawner – триггер который «рождает» самолет (в данном примере) в мире;
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает триггер Spawner .
- OUT – любое событие миссии, которое активируется по сообщению On Spawned Message т.е. после «рождения» заданного объекта в миссии.



Триггер «рождающий» копии объектов с одинаковыми свойствами в игровом мире.

Даблклик по иконке триггера открывает меню Spawner Advanced Properties.

Чтобы «родить» объект достаточно установить объектную связь от триггера к объекту и запустить триггер.

Если после «рождения» объекту необходимо что то сделать (например, полететь или поехать в Way Point), тогда нужно установить связь «сообщение» от объекта к следующей команде или триггеру этого же объекта (сообщение On Spawned Message: Target).

Spawner может рождать объект сколько угодно раз.

ВНИМАНИЕ! - «рождать» можно только одиночные объекты и нельзя «рождать» игрока, или группу объектов.

ВНИМАНИЕ! – если данный триггер прилинкован объектной связью к объекту в миссии, данный объект в миссии не появится если не запустить триггер Spawner.

ВНИМАНИЕ! – скрипт миссии не различает «копии» (то есть рожденные через данный триггер) объекта в миссии.

Текстовый формат триггера:

MCU_Spawner //тип MCU

{

Index = 31; //ID MCU

Name = "Trigger Spawner"; //имя MCU (не отображается в GUI)

Desc = ""; //описание MCU (не отображается в GUI)

```

Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер, у данного
триггер исходящих целевых связей быть не может
Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
XPos = 2152.26; // координата по оси X
YPos = 115.08; // координата по оси Y
ZPos = 4254.13; // координата по оси Z
XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
}

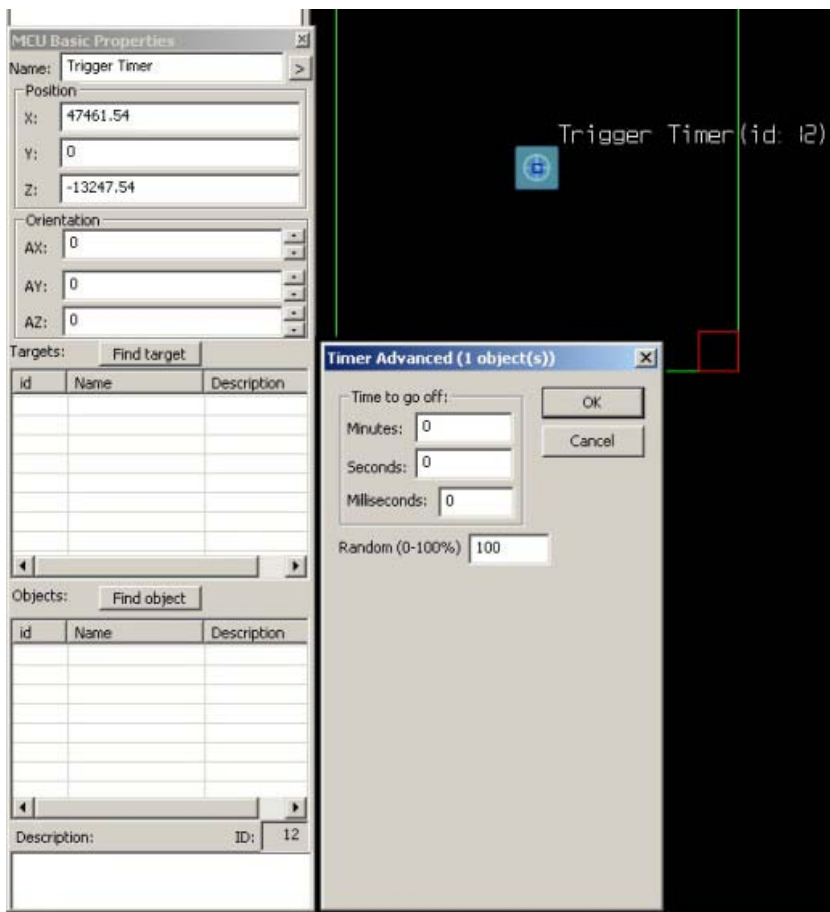
```

7.8.8. Trigger:Timer



На приведенном примере:

- Trigger: Timer – триггер, который срабатывает по заданному времени;
- IN - любое исходящее из миссии событие, запускающее триггер Timer.
- OUT – любое событие миссии, которое запускается при срабатывании триггера Timer.



Триггер, срабатывающий по истечении заданного времени.

Даблклик по иконке триггера открывает меню Timer Advanced Properties.

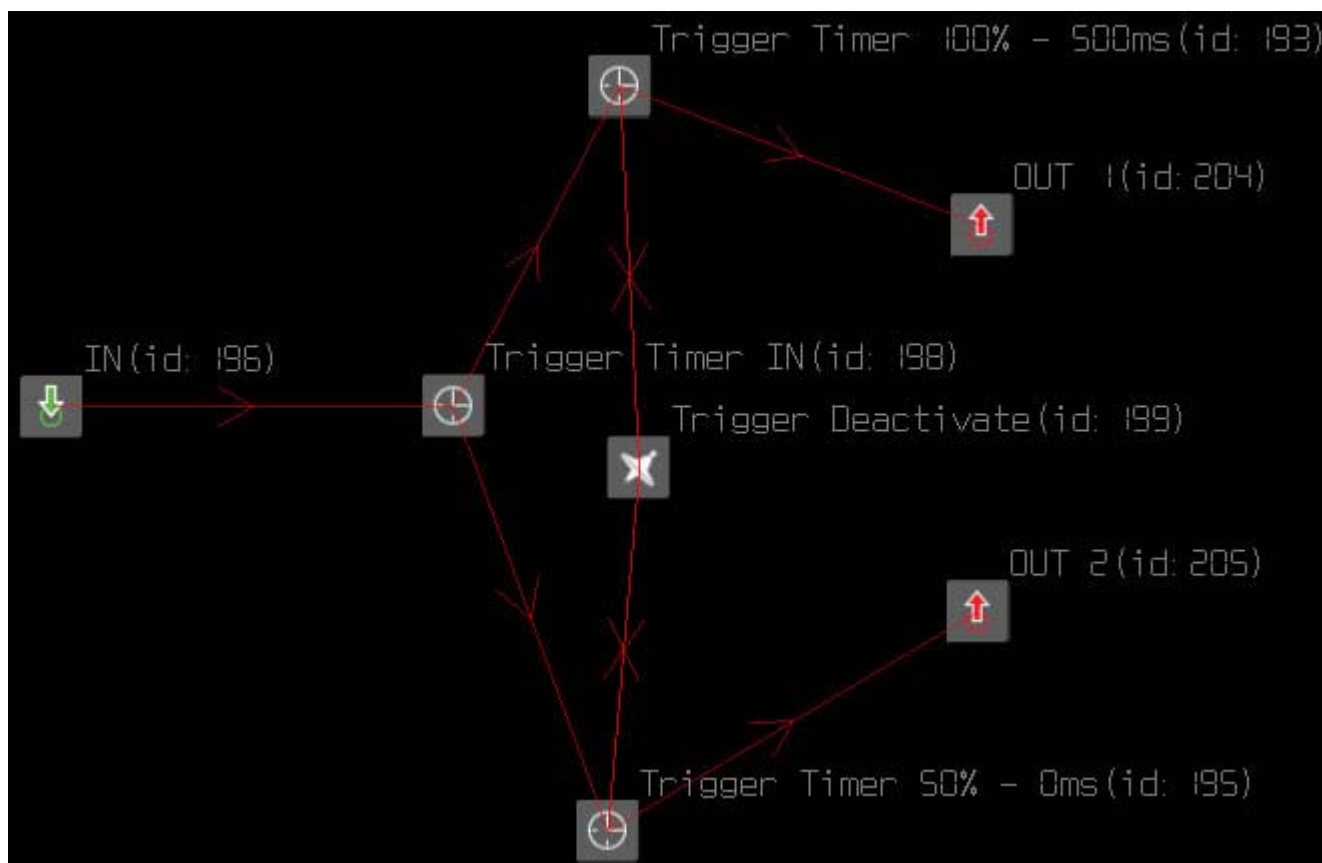
Advanced Properties:

Поля Time to go off: Minutes, Seconds, Milliseconds – устанавливают время в минутах, секундах и миллисекундах, по истечению которого сработает триггер.

Поле Random (0-100%) – устанавливает вероятность срабатывания триггера в процентах.

7.8.9. Схема Random Timer.

Данная схема при помощи функционала Trigger:Timer позволяет запускать в миссии любые случайные события в зависимости от выставленных вероятностей:



На приведенном примере – схема random 50/50:

- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи запускает триггер Timer IN.
- Timer IN – буферный таймер (фактически пустая проставка, просто для удобства схемы);
- Trigger:Timer 100% - 500ms – таймер с установкой Random=100% и временем в 500ms;
- Trigger:Timer 50% - 0ms – таймер с установкой Random=50% и временем в 0ms;
- Trigger:Deactivate – триггер деактивирующий, либо “Trigger:Timer 100% - 500ms”, либо “Trigger:Timer 50% - 0ms” – в зависимости, какой из таймеров первым «дернет» деактиватор;

- OUT 1 – любое событие миссии, которое запускается при срабатывании триггера “Timer 100% - 500ms.”;
- OUT 2 – любое событие миссии, которое запускается при срабатывании триггера “Timer 50% - 0ms.”.

Схема работает следующим образом:

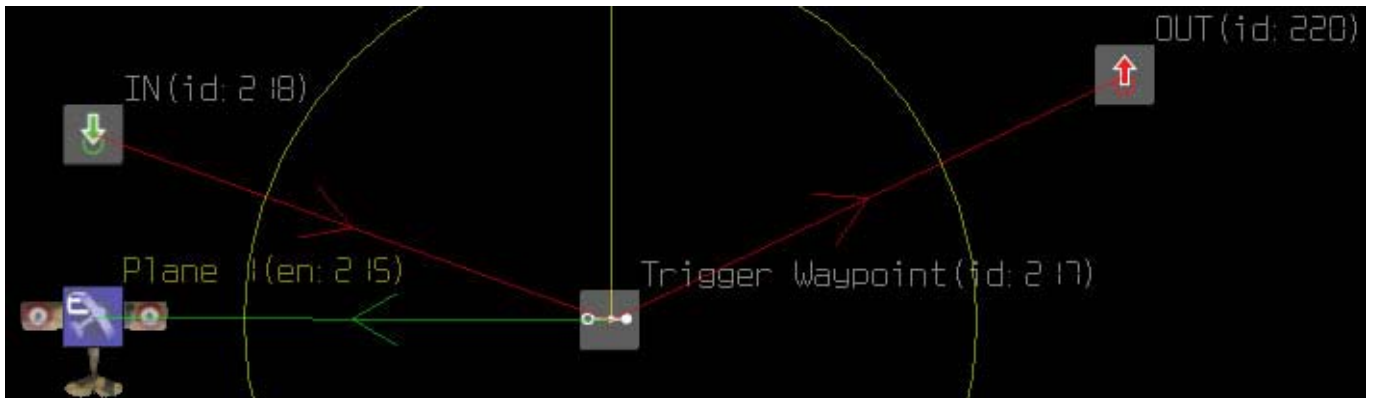
- если сработает “Timer 50% - 0ms.” (то есть 50% рандом таймер сработает раньше, чем 100% таймер), то он деактивирует “Trigger:Timer 100% - 500ms” и сработает (дернется) событие миссии по ветке “OUT 2”;
- если сработает “Trigger:Timer 100% - 500ms”, то он деактивирует “Trigger:Timer 50% - 0ms” и сработает (дернется) событие миссии по ветке “OUT 1”.

Таким образом, можно «балансировать» условие срабатывания в зависимости от установок Random в таймере.

Текстовый формат триггера:

```
MCU_Timer //тип MCU
{
  Index = 32; //ID MCU
  Name = "Trigger Timer"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер, у данного
триггер исходящих объектных связей быть не может
  XPos = 2164.17; // координата по оси X
  YPos = 115.08; // координата по оси Y
  ZPos = 4280.93; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Time = 0; //установка значения времени
  Random = 100; //вероятность срабатывания триггера в процентах
}
```

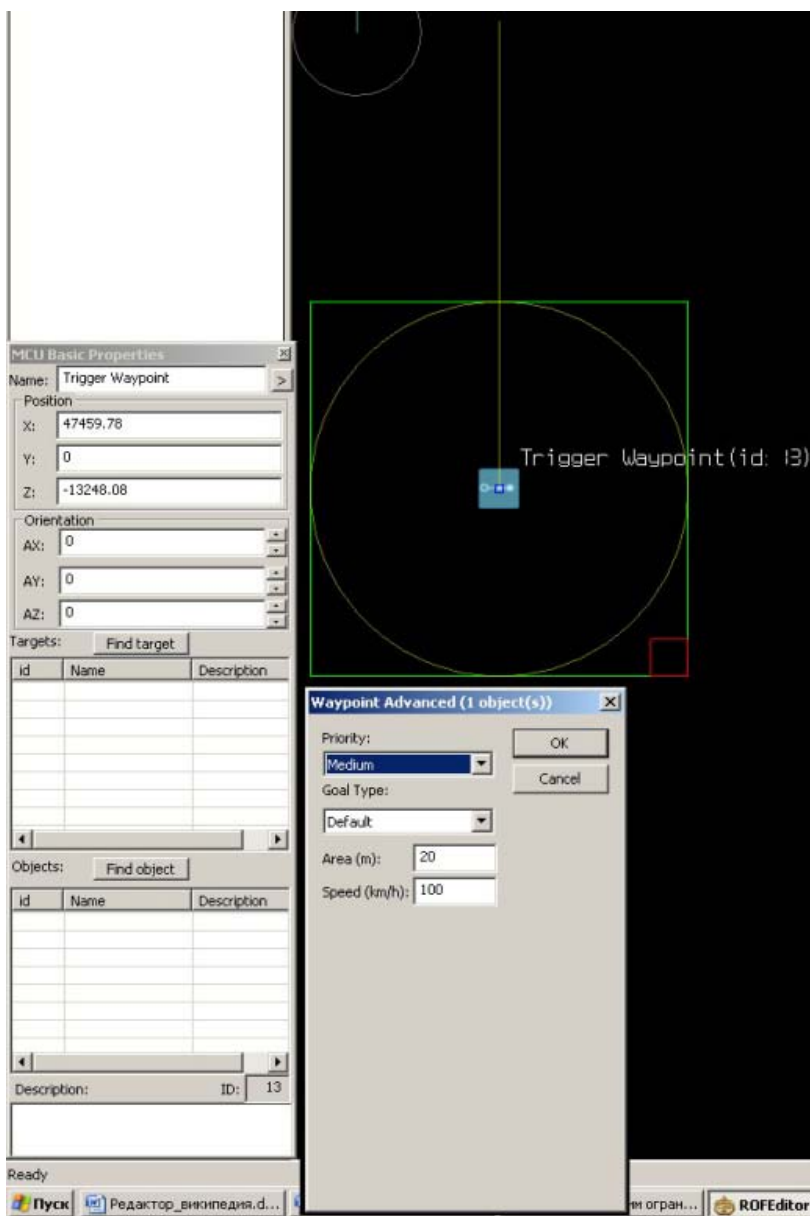
7.8.10. Trigger:Waypoint



На приведенном примере:

- Trigger: Waypoint – триггер, который задает команду объекту двигаться к этому триггеру и срабатывает по данному объекту, т.е. **фактически Waypoint одновременно является так же и командой для движущихся объектов;**
- IN - любое входящее событие из миссии, которое при помощи целевой (или Message) связи активирует триггер Waypoint.
- OUT – любое событие миссии, которое запускается при срабатывании триггера Waypoint.

Данный триггер **ОБЯЗАТЕЛЬНО** должен иметь связанный объект (или объекты) т.е. должен быть связан с ним объектной связью.



Даблклик по иконке триггера открывает меню Waypoint Advanced Properties.

Advanced Properties:

Поле Priority: High, Medium, Low – уровень приоритета выполнения движения в эту точку для AI.


Low – AI во время полета к точке с таким приоритетом, будет производить поиск любых доступных целей и по возможности будет атаковать их;


Medium – AI во время полета к точке с таким приоритетом будет отклоняться от маршрута или предпринимать какие-либо действия, если AI определит угрозу для себя (атакующий сзади самолет – AI будет обороняться);

High – AI будет лететь к точке вне зависимости от исходящих угроз до тех пор, пока он не достигнет заданной точки (не выполнит команду).

Поле Goal Type – Default, Primary, Secondary. Три вида визуализации иконки контрольной точки маршрута. Отображение на полетной карте и в игре. Сделано для удобства визуального распознавания контрольных точек маршрута и никак не влияет на результат миссии.

Default – иконка выглядит 

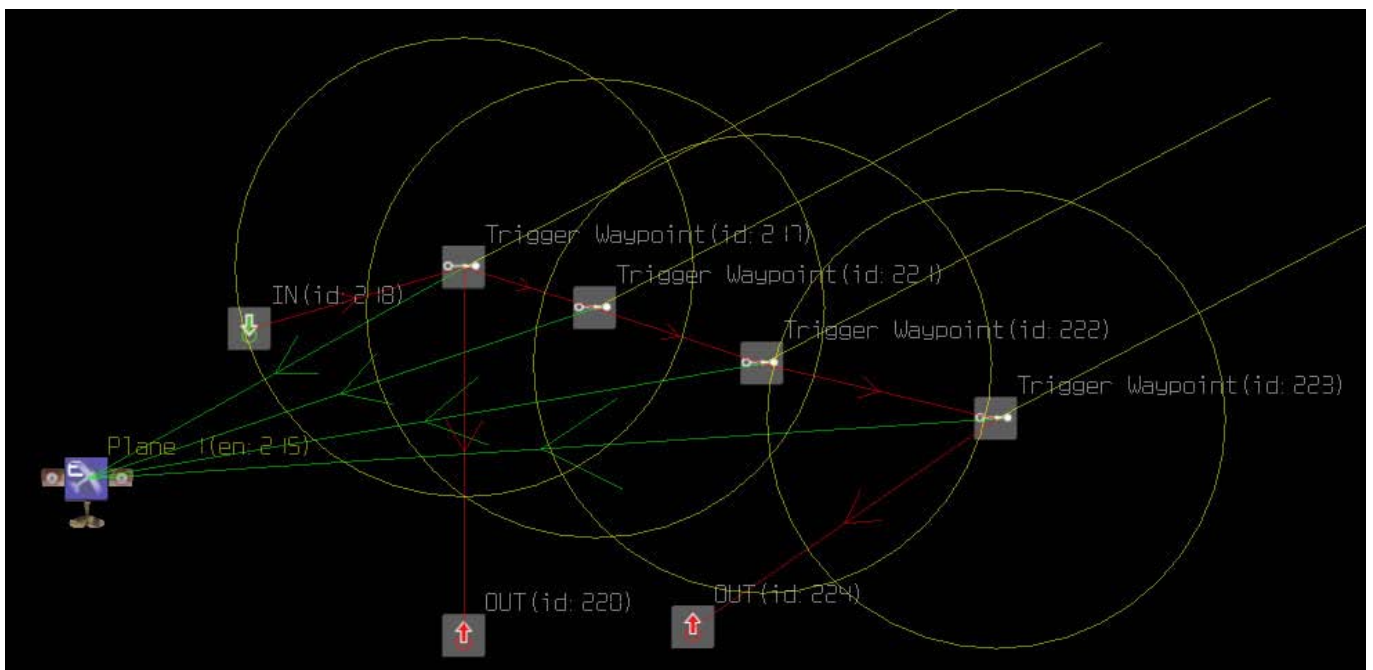
Primary – иконка выглядит 

Secondary – иконка выглядит 

Поле Area (m): - размер (радиус) области срабатывания триггера, по факту область Waypoint – бесконечный по высоте цилиндр;

Поле Speed (km/h): - указывает с какой скоростью (в километрах/час) объект должен перемещаться к этому триггеру.

При расстановке Waypoint для выделенного объекта существует возможность автоматизированной расстановки Waypoint при использовании зажатой клавиши Left Shift+ЛКМ все объектные и целевые связи по маршруту устанавливаются автоматически.



Соответственно, т.к. Waypoint функционально – триггер, то при срабатывании он может «дергать» как следующие по цепи Waypoint, так и другие события в миссии.

При установке радиуса Waypoint (Area) старайтесь не ставить очень маленький радиус, потому что AI может промахнуться мимо Waypoint и будет повторять заходы до тех пор, пока не попадет в заданную область.

Не забывайте устанавливать высоту (YPos) положения маршрутной точки, потому как AI будет лететь именно на заданной в этой точке высоте.

Если разным объектам (объекты не объединены в группу) назначен один и тот же Waypoint, то триггер сработает только для первого вошедшего в Waypoint Area, второй объект при пролете (проезде) этого Waypoint останется над ним (в нем), если, далее, этому объекту не назначено другой команды.

Существует возможность «перенацеливания» объекта на другой маршрут. Например, объект движется в заданную точку и получает другую маршрутную точку – при этом он направится в новую полученную точку вне зависимости от приоритета новой маршрутной точки. В этом случае все неиспользуемые (не сработавшие) Waypoint обязательно должны быть деактивированы триггером Deactivate, иначе, если данный объект случайно пролетит через не сработавший Waypoint – он сработает, т.к. находится в буфере команд AI.

ВНИМАНИЕ! – если при выполнении AI любой текущей команды поступила команда Waypoint – то команда Waypoint «перебьет» все другие текущие команды вне зависимости от приоритета и типа выполняемых команд, поэтому всегда выдавайте команду ПОСЛЕ (например с проставкой через таймер) выдачи команды Waypoint.

Текстовый формат триггера:

MCU_Waypoint //тип MCU

```
{
  Index = 33; //ID MCU
  Name = "Trigger Waypoint"; //имя MCU (не отображается в GUI)
  Desc = ""; //описание MCU (не отображается в GUI)
  Targets = []; // ID объектов, на которые целевой связью ссылается данный триггер
  Objects = []; // ID объектов, на которые объектной связью ссылается данный триггер
  XPos = 2092.68; // координата по оси X
  YPos = 111.939; // координата по оси Y, фактически это высота положения маршрутной точки
  ZPos = 4319.66; // координата по оси Z
  XOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  YOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  ZOri = 0; //ориентация в пространстве (не несет функционала для миссии)
  Area = 20; //радиус срабатывания команды
  Speed = 100; //скорость полета для AI
  Priority = 1; //приоритет выполнения команды для AI
  GoalType = 0; //влияет на вид иконки при отображении в GUI игры
}
```

8. Создание кооперативной миссии.

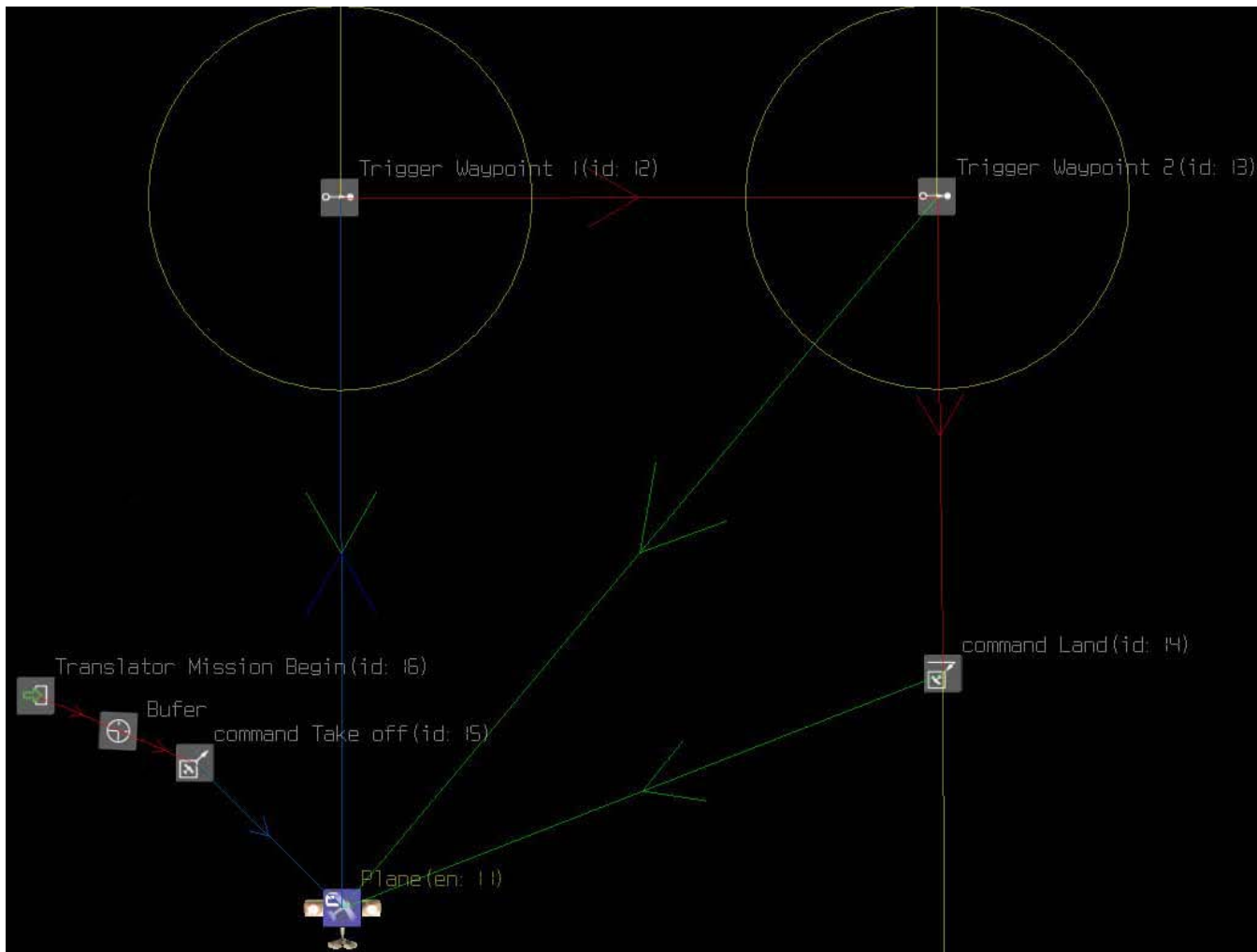
По сути кооперативная миссия не отличается от одиночной миссии за исключением следующих условий:

- тип миссии (см.п.5.1.) должен быть “cooperative”;
- для того, чтобы кооперативная миссия корректно работала, нужно сохранять её в каталог \data\Multiplayer\Cooperative\, при этом вместе с миссией, обязательно должен лежать её лист-файл (*.list);
- в кооперативной миссии может «сработать» (будет засчитана как успешно выполненная) только одна основная цель (см.п. 7.7.5.) с установкой Primary Tasks/Order coalition =>Success, т.к. средствами миссии, в кооперативном режиме, можно задать только одного победителя – Order Coalition, соответственно, если в течении миссии НЕ СРАБАТЫВАЕТ Translator:Mission Objective/ Order coalition =>Success, то автоматически побеждает Opposite Coalition. Для вторичных целей, ограничения такие же как и для одиночной миссии, поэтому будьте внимательны – если в миссии есть несколько Mission Objective/Order coalition =>Success, то миссия засчитает только один – последний сработавший;
- если после выполнения Order coalition своей цели, при завершении миссии на стороне Order coalition не останется ни одного живого игрока, то для Order coalition будет автоматически засчитано поражение;
- если в миссии не осталось ни одного живого игрока, то будет засчитана ничья;
- для того, чтобы самолет был доступен для выбора в лобби кооперативной миссии, в свойствах самолета (см.п.7.3.1.) должна быть установлена галочка cooperative start. Таким образом можно задавать как старт на земле, так и старт в воздухе;
- будьте внимательны при установке Default Camera Operator для коалиций (см.п. 7.7.2), не забывайте их устанавливать - каждую для своей коалиции, для того, чтобы игроки, которые будут участвовать в миссии как зрители (spectators) появлялись в нужном месте карты при загрузке миссии;
- не забывайте правильно выставлять свойства Translator:Icon (см.п. 7.7.3.);
- в кооперативной миссии не может быть самолета с AI=Player (см.п.7.3.1.);
- если в кооперативной миссии, самолет не занят ни одним из игроков, то после старта миссии данный самолет полетит как AI с установленным в миссии уровнем AI

В остальном, весь логический механизм работы миссии (работа MCU, действия AI и п.р.), будет работать так же, как и в одиночной миссии.

9. Часто используемые блоки миссий, примеры и приемы построения.

9.1. Взлет, полет по маршруту и посадка.



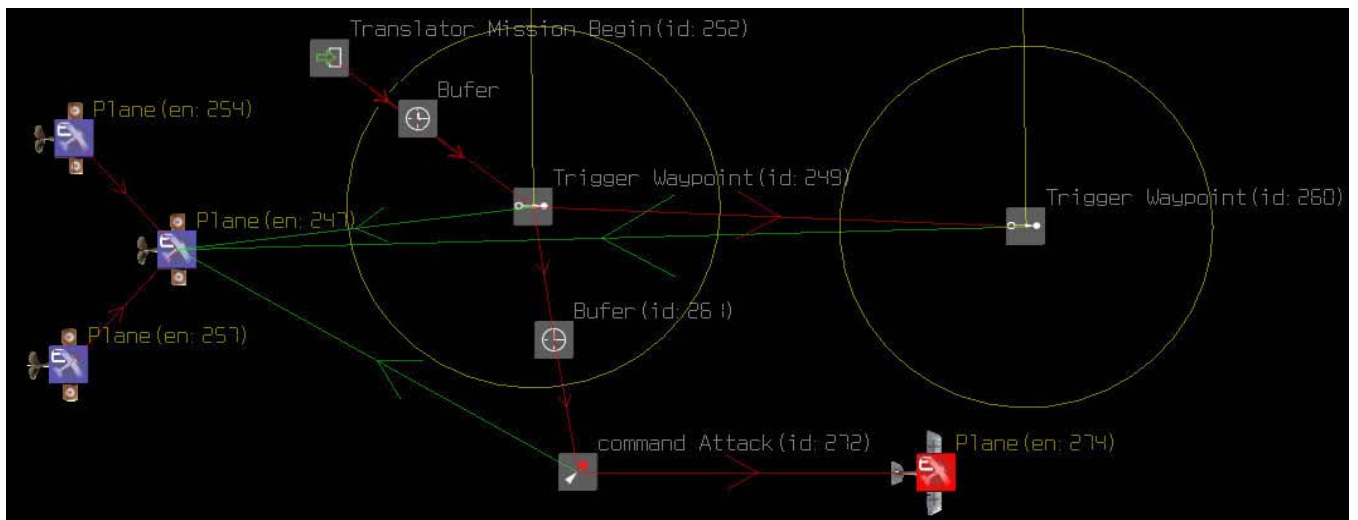
На приведенном примере:

- команда Take Off (см.п.7.6.13) объектной связью связана с самолетом и запускается при помощи Translator:Mission begin, то есть, самолет, который «знает» что эта команда ссылается на него и при активации данной команды – выполняет её;
- Waypoint (см.п.7.8.10.) 1 и 2 – маршрутные точки самолета, которые, в свою очередь, так же ссылаются на самолет объектной связью. Первый Waypoint активируется при помощи On Take off Message самолета (см.п.2.3.) и целевой связью «дергает» следующий по ходу самолета Waypoint. То есть, самолет при выполнении команды взлета (фактически – взлетит) «дернет» первый Waypoint. В Advanced Properties самолета в разделе On Messages table (см.п.7.3.1.) будет следующая картина: [Message Type: Take Off; Command ID:15; Target ID 12];
- команда Land, ссылается на самолет объектной связью и активируется от крайней маршрутной точки самолета;

Таким образом, в приведенном примере самолет взлетит. Пролетит по двум маршрутным точкам и выполнит посадку.

- три самолета объединены в звено, все команды через объектную связь привязаны к ведущему (ID:225) звена (см.п.6.5.);
- звено летит по маршрутным точкам, при этом маршрутные точки «дергают» команды на перестроение (см.п.7.6.8.) через проставку в виде таймера, во избежание одновременного поступления нескольких команд одному (в данном случае waypoint и formation) объекту - ведущему группы;
- команды Formation объектной связью ссылаются на ведущего группы, тем самым транслируя ему команду на перестроение группы

9.3. Команда Attack на маршруте.

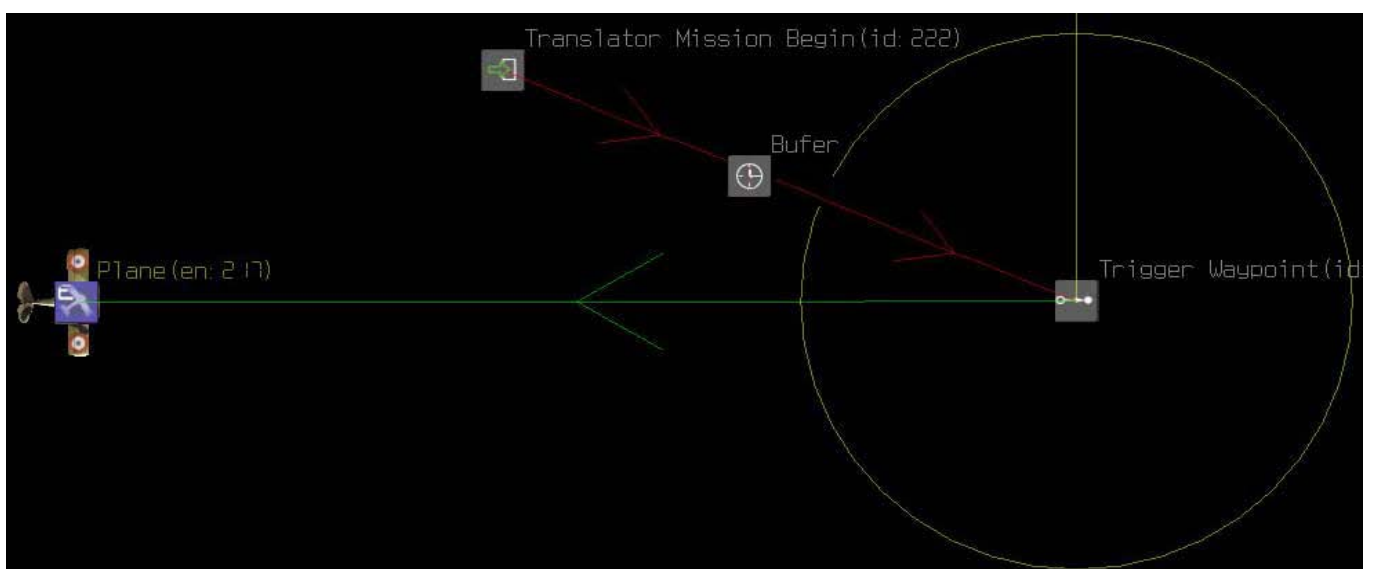


На приведенном примере:

- три самолета объединены в звено, все команды через объектную связь привязаны к ведущему (ID:247) звена (см.п.6.5.);
- звено летит по маршрутным точкам, при этом, через проставку в виде таймера дергается команда Attack, которая объектной связью ссылается на ведущего группы, а целевой связью указывает на объект атаки.

Соответственно, в данном примере группа получит две команды последовательно – сначала следующую маршрутную точку, а через время указанное в таймере – команду на атаку, группа выполнит команду атаки (например - уничтожит цель) и после её выполнения полетит на маршрутную точку, которая осталась в «памяти» AI.

9.4. Старт в воздухе.

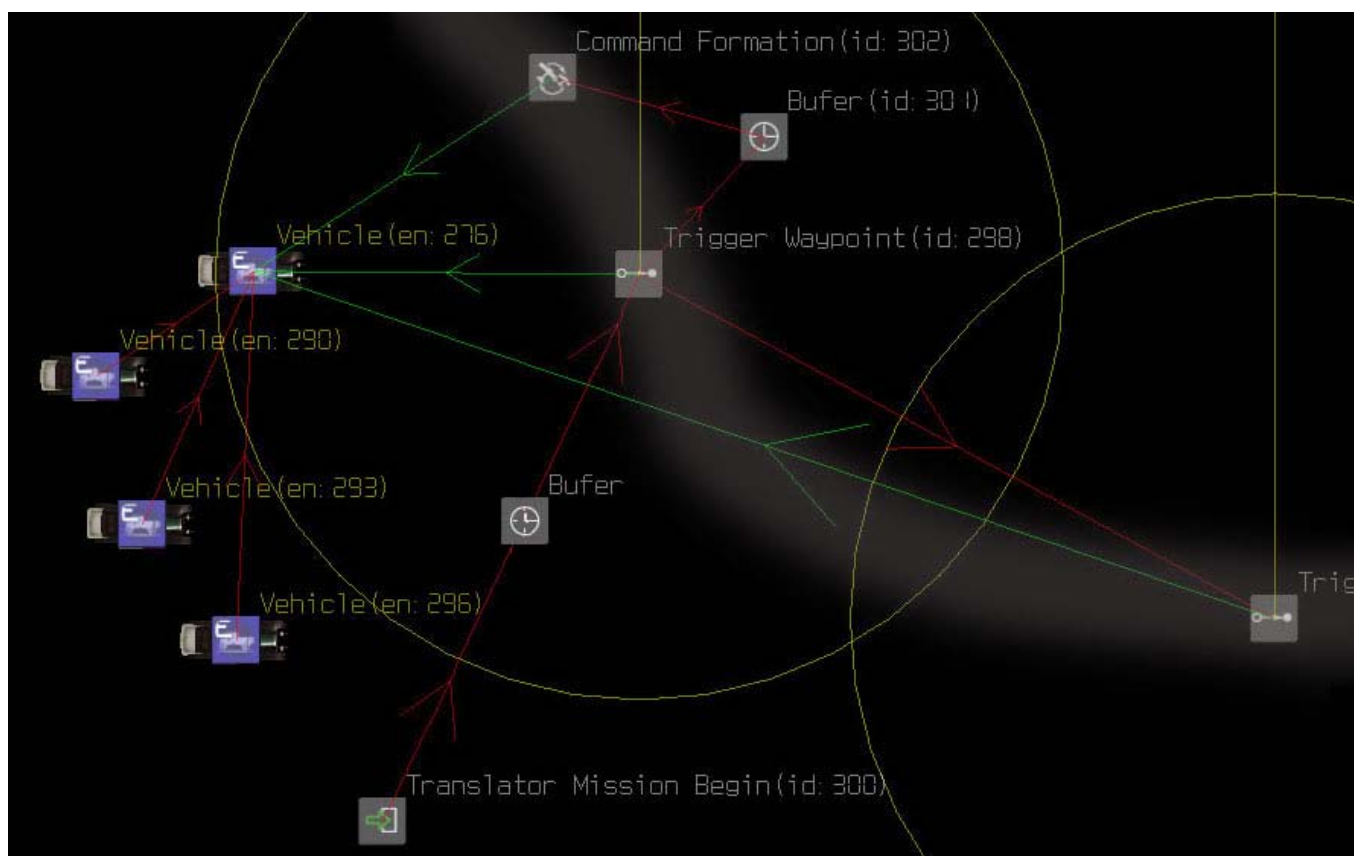


На приведенном примере самолет после появления в воздухе получит команду на движение к маршрутной точке (Waypoint).

Для старта самолета в воздухе фактически достаточно в его свойствах установить галочку “Start in the Air” (см.п.7.3.1.). При этом, не забывайте установить высоту (Position: Y), на которой он

должен появиться (см.п.7.2.1.). То же самое касается и создания кооперативных миссий со стартом в воздухе.

9.5. Колонна автомобилей на дороге.



На приведенном примере:

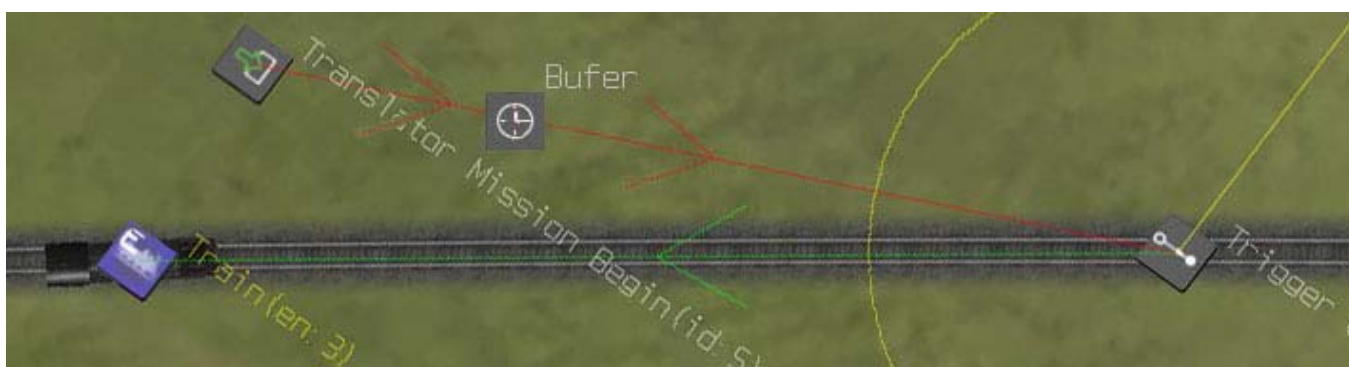
- автомобили объединены в группу и начнут движение после поступления команды на движение к маршрутной точке (Waypoint);
- по достижении маршрутной точки, через проставку в виде таймера, группа получит команду на построение (см.п.7.6.8.), и если в свойствах Formation будет стоять “Vehicle: On Road Column”, то автомобили найдут ближайшую дорогу (если дизайнер миссии не поставил их сразу на дорогу) и выстроившись колонной, поедут по ней до заданной следующей маршрутной точки.

Фактически, для того, чтобы заставить двигаться любой мобильный объект типа Vehicle или Train, нужно «дернуть» его маршрутную точку.



На приведенном выше примере Автомобиль (или поезд) движется по дороге по нескольким маршрутным точкам. По достижении первой контрольной точки (id:208) автомобиль получает команду на движение по дороге (id:213) (в случае поезда – данная команда не требуется) и направляется к следующему Waypoint и так далее по цепочке.

9.6. Движение поезда.

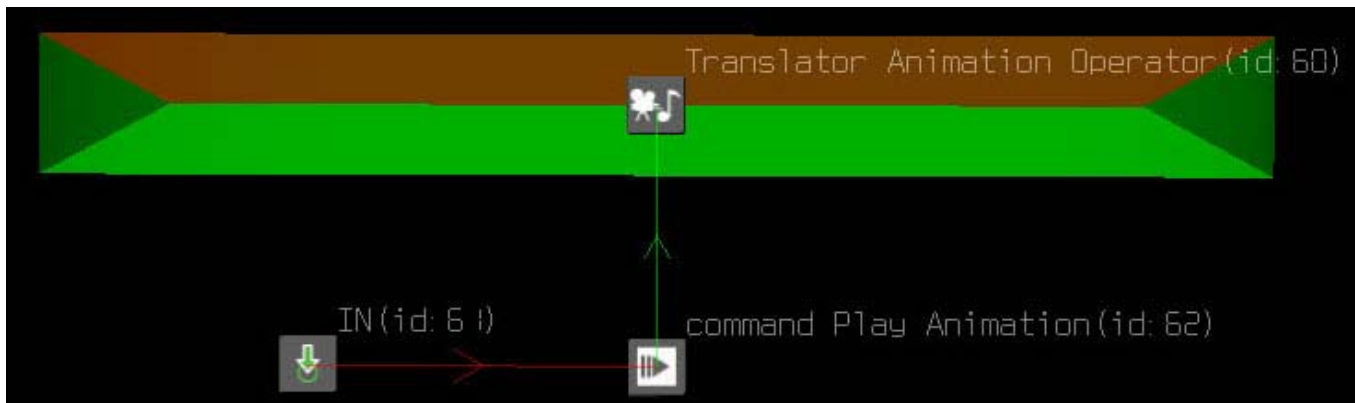


Для того, чтобы «запустить» поезд (см.п.7.2.2.), достаточно поставить его на рельсы и дернуть его первую маршрутную точку.

ОБЯЗАТЕЛЬНО – ставьте поезд и его маршрутные точки на рельсы.

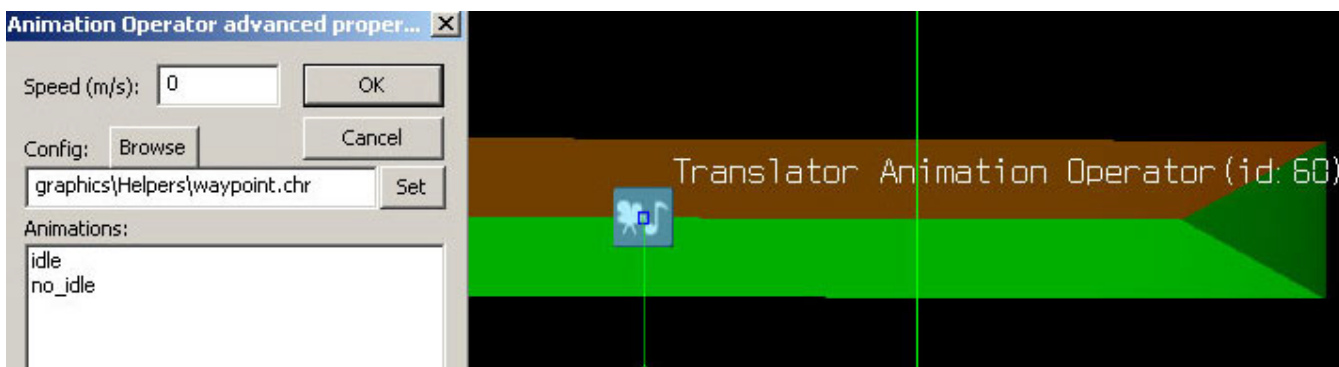
9.7. Создание анимационной сцены на примере хелпера (фигура в воздухе).

9.7.1. Объект анимации виден сразу после запуска миссии.



На приведенном примере:

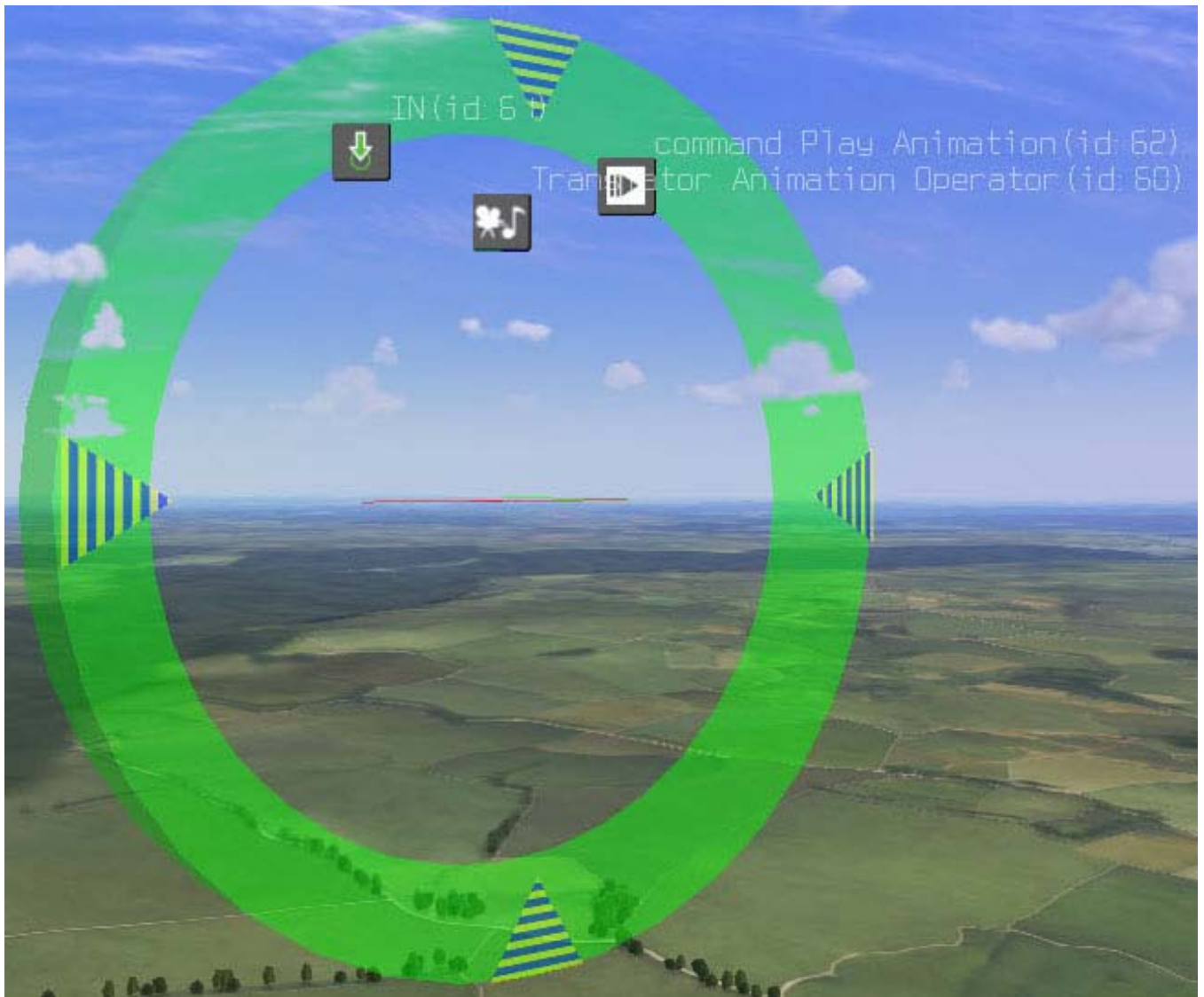
- устанавливаем Translator:Animation Operator (см.п.7.7.1.) и в его свойствах выбираем нужную нам сцену (в данном случае waypoint.chr или любой другой *.chr файл).



- command:Play Animation (см.п.7.6.11.) – объектной связью ссылается на Animation Operator, внутри Play Animation стоит либо idle либо no_idle (то есть в данном случае – проиграть/не проиграть).

- IN – любое исходящее из миссии событие, запускающее Play Animation.

Вид данной фигуры в 3D:



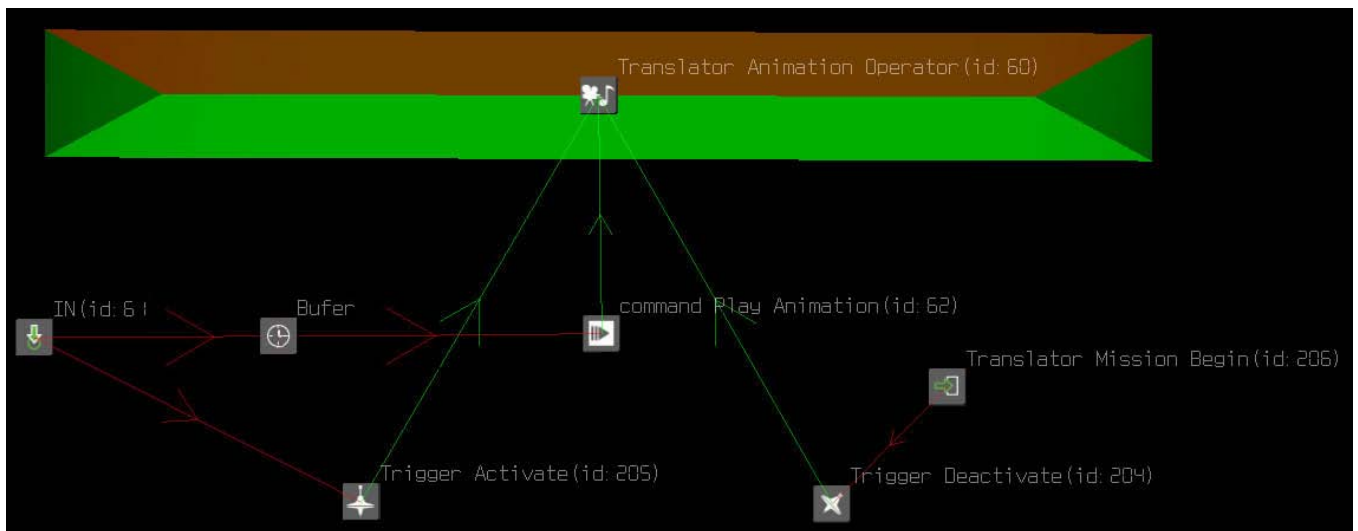
Соответственно, после того, как будет запущена команда Play Animation, анимированные части данной фигуры начнут двигаться (в данном примере это полосатые стрелочки).

Не забывайте выставить высоту Animation Operator и его ориентацию.

Так же, с помощью редактора, можно протестировать запуск анимации, для этого:

- выделите Play Animation, по правой кнопке в меню выберите Select Object Menu =>Play;
- после этого вид автоматически переключится в 3D, а курсор мыши изменит свой вид;
- может случиться так, что взгляд будет направлен не на Animation Operator, если это так, то просто найдите объект на карте (см.п.4.3.).

9.7.2. Объект анимации виден НЕ сразу после запуска миссии.



На приведенном примере:

- Mission Begin деактивирует Animation Operator с самого начала миссии (то есть фигура будет не видна);
- IN – любое входящее событие миссии, по которому активируется Animation Operator и запускается его анимация (через Play Animation).

Соответственно, данный пример отличается от предыдущего, лишь его начальной реализацией.

Так же, повторно можно выключить (деактивировать) и включить (активировать и повторно дернуть Play Animation) любой подобный анимированный объект, установленный через Animation Operator.

9.8. Создание анимационной сцены на примере сцены из тренировочной кампании.

Для создания такого типа сцены нужно:

- *.MGM файл – выгруженная 3D сцена (редактор его не использует, но он косвенно подчинен *.chr файлу);
- *.chr файл – файл, описывающий анимации в сцене;
- *.cam файл – файл, описывающий траекторию «полета» камеры в сцене;
- *.mp3 файл – файл, содержащий синхронизированную со сценой звуковую дорожку.

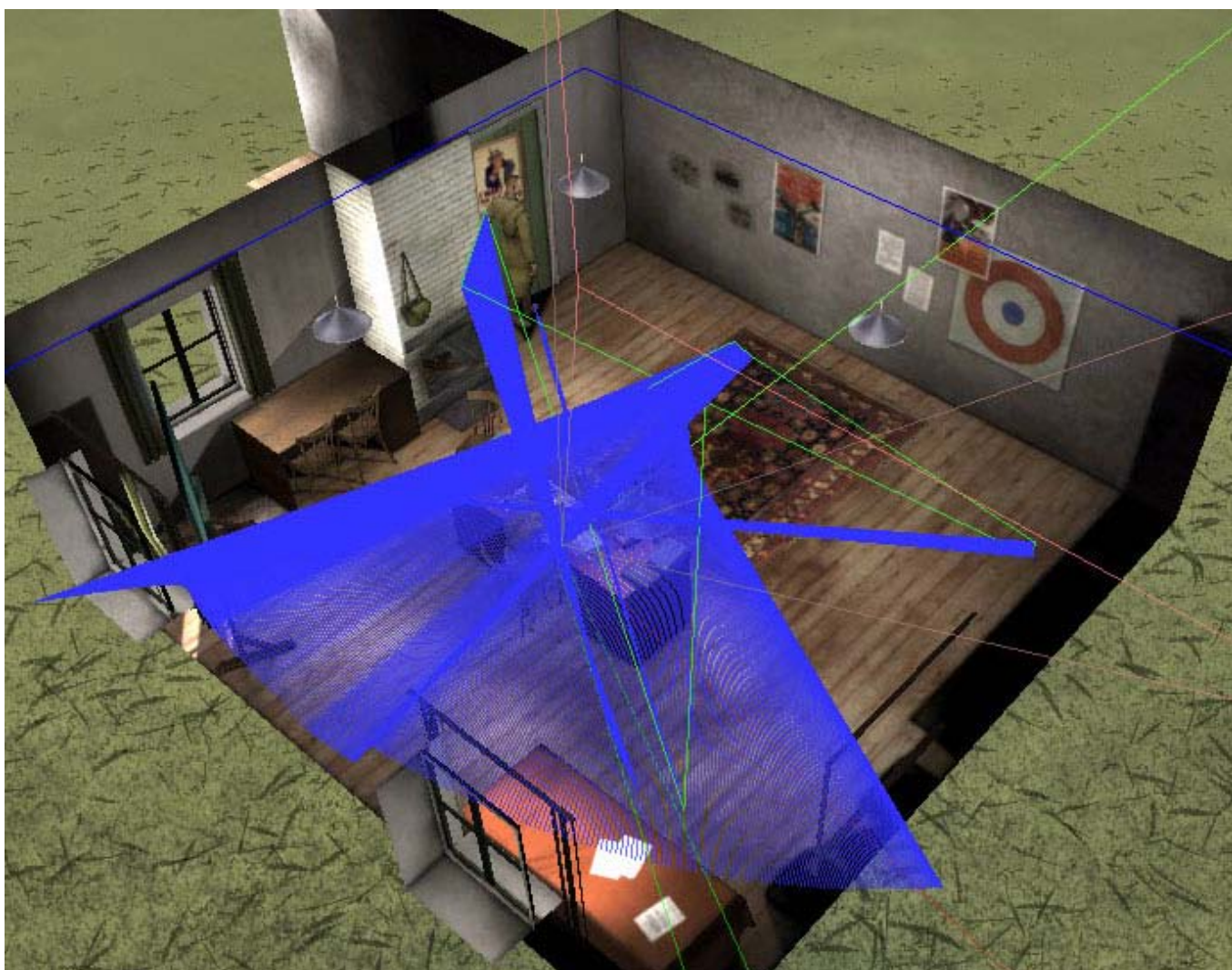
Для построения такой сцены нужно использовать:

- Translator:Animation Operator (см.п.7.7.1.);
- Translator:Camera Operator (см.п.7.7.2.);
- Command: Play Animation (см.п.7.6.11.);
- Command:Play Sound (см.п.7.6.12.).



На приведенном примере:

- Animation Operator запускается при помощи Play Animation. В свойствах Animation Operator прописана сцена (*.chr – указываем его в разделе “config”) и звуковая дорожка (*.mp3 – выставляем опцию “config is sample” и в разделе “sound config” указываем путь к звуковому файлу);
- Camera Operator с прописанным конфигурационным файлом камеры (*.cam файл – указываем путь в разделе “3D path”, при этом не забываем выбрать тип камеры как “self”) файл запускается таймером (или любым событием из миссии). При правильном указании пути камеры, появится визуализированная траектория камеры (синим цветом направление камеры, зеленым – её траектория)



На примере выше, Animation Operator и Camera Operator должны иметь одинаковые координаты, выставленные в Basic Properties (см.п.7.4.1.), это нужно сделать так как сцена и камера действуют сообща и имеют общие начальные координаты (в разделе Position одинаковыми должны быть все три координаты, т.е. X, Y и Z).

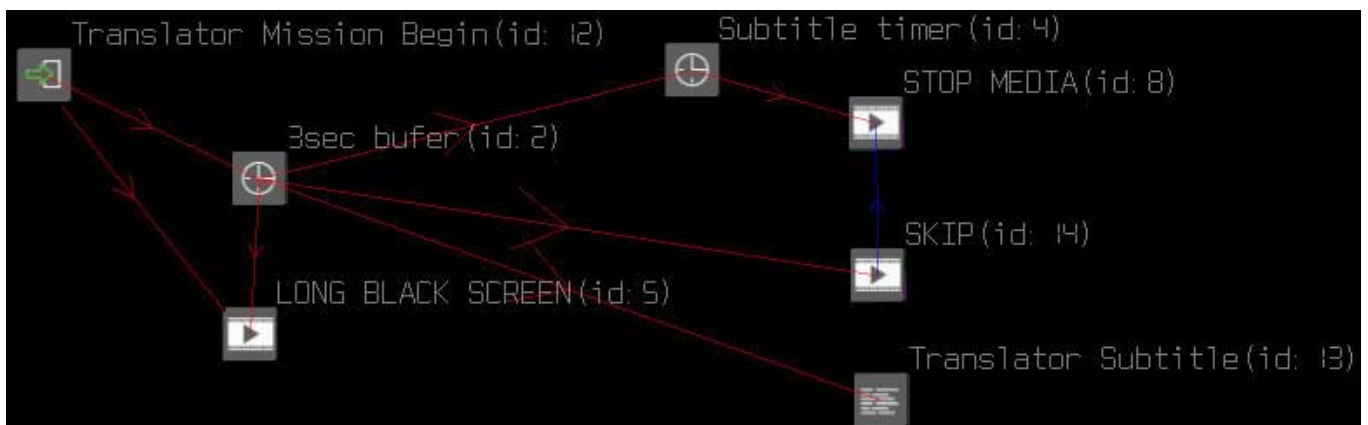
Далее, нужно запустить все элементы сцены одновременно (т.к. в данном примере это требуется по дизайну):

- Play Animation (для запуска анимации) и Play Sound (для запуска звуковой дорожки) запускают Animation Operator (связаны с ним через объектную связь);
- Camera Operator запускается напрямую, через целевую связь.

9.9. Экран (или заставка) при старте миссии.

Задача, после старта миссии, на черном фоне вывести текст, при этом, внизу экрана должно находиться SKIP меню для того, чтобы если игрок не захочет читать текст, он смог бы пропустить его (или любую другую заставку).

Помня об ограничениях, перечисленных в пунктах 7.4.7. и 7.7.8. можно использовать следующую схему:



- 3sec buffer – таймер, который запускает все события в миссии с задержкой, для того, чтобы не запустить Subtitle раньше времени, пока после загрузки миссия не «встанет» на паузу (см.п.7.7.8);
- LONG BLACK SCREEN (Translator Media) с установками Media Type:Image \ [черная картинка размером минимум 1 на 1 пиксель].jpg и “base time” 10000 секунд, с черным задним фоном (Background color RGB) [0;0;0] запускается от самого начала миссии (то есть сразу после загрузки) Mission Begin. То есть, так как мы, из-за ограничений, не можем контролировать время Media вставок в миссию (то есть после загрузки миссия встанет на паузу – а Media нет), от старта миссии мы запускаем очень длинный, черный задний фон;
- Translator Subtitle – субтитры в миссии, которые запускаются через 3 секунды, после старта миссии;
- STOP MEDIA (Translator Media) – с установками Media Type:Stop Media, который запустится через таймер “Subtitle timer”, внутри которого стоит время, равное времени, установленному внутри Subtitle;

- SKIP – (Translator Media) - с установками Media Type:Flash и указанием пути к Flash элементу “Skip” диалога (swf\skip.swf), так же, соответственно, в “base time” нужно установить такое же значение, как и в Subtitle – чтобы синхронизировать их между собой.
- SKIP при помощи On Media Stopped Message (см.п.7.7.4) «деграет» STOP MEDIA, то есть, если во время миссии игрок кликнет на Skip диалог, то SKIP остановит всю Media – LONG BLACK SCREEN и Subtitle и игрок сможет начать игровой процесс;
- если же игрок не нажмет Skip диалог, то по истечении времени внутри Subtitle timer, этот таймер остановит всю Media в миссии.

9.10. Базовая миссия – все города, аэродромы и мосты на карте.

Фактически, все объекты на карте – это объекты редактора, поэтому, для создания такой миссии, вам понадобятся группы файлов (*.Group см.п. 6.6.).

Все объекты карты находятся в следующих *.Group файлах:

- Base-no-trunc.Group - все крупные исторические города и все аэродромы;
- Base-for-trunc.Group – все мелкие города, деревни и мосты.

То есть, для получения такой миссии, вам нужно открыть пустую миссию и последовательно импортировать эти группы, после чего сохранить миссию.

Так же можно импортировать объекты в уже готовую миссию, но будьте внимательны – не импортируйте объекты повторно, чтобы не создать «объект в объекте».

Так как размер карты большой и количество объектов на карте еще больше, если по сюжету миссии игрок не будет летать далеко, то рекомендуем вам вырезать под миссию шаблон, который расставит объекты карты в пределах, заданных миссией. Такое решение позволит сэкономить ресурсы компьютера и значительно снизит время загрузки миссии.

9.11. Файлы, используемые в Media и Animation Operator.

9.11.1. Media файлы в сборке игры:

- Campaigns\Training\Images\BlackFade.jpg – черный Image (*.jpg) 1x1px
- swf\skip_ratio.swf – кнопка SKIP для использования с 3D движка (учитывает разные соотношения сторон экрана);
- swf\skip.swf – кнопка SKIP для использования с видео/флэш вставками;

9.11.2. *.chr файлы для Animation Operator:

- graphics\Helpers\waypoint.chr – круглое кольцо с анимированными стрелочками;
- graphics\Helpers\arrow.chr – большая зеленая 3D стрелка;
- graphics\Helpers\descent30.chr – 3D траектория – снижение 30 градусов
- graphics\Helpers\descent45.chr – 3D траектория – снижение 45 градусов

- graphics\Helpers\loop.chr – 3D траектория – петля
- graphics\Helpers\poi.chr – 3D траектория – анимированная контрольная точка
- graphics\Helpers\turn180.chr – 3D траектория – вираж на 180 градусов
- graphics\Helpers\turn360.chr – 3D траектория – вираж на 360 градусов
- graphics\Helpers\turn90.chr – 3D траектория – вираж на 90 градусов

10. Общая информация

Автор руководства: Сечной Виктор

Разработчик редактора миссий: Михаил Лепахин

Составители: Сечной Виктор, Михаил Левин, Михаил Лепахин, Даниил Тусеев

Редактура руководства: Сечной Виктор, Михаил Лепахин, Михаил Левин, Адамов Арсений, Тусеев Даниил.

Актуальная версия: ROF v1.007



© neoqb LLC, 2008—2009. Все права защищены.