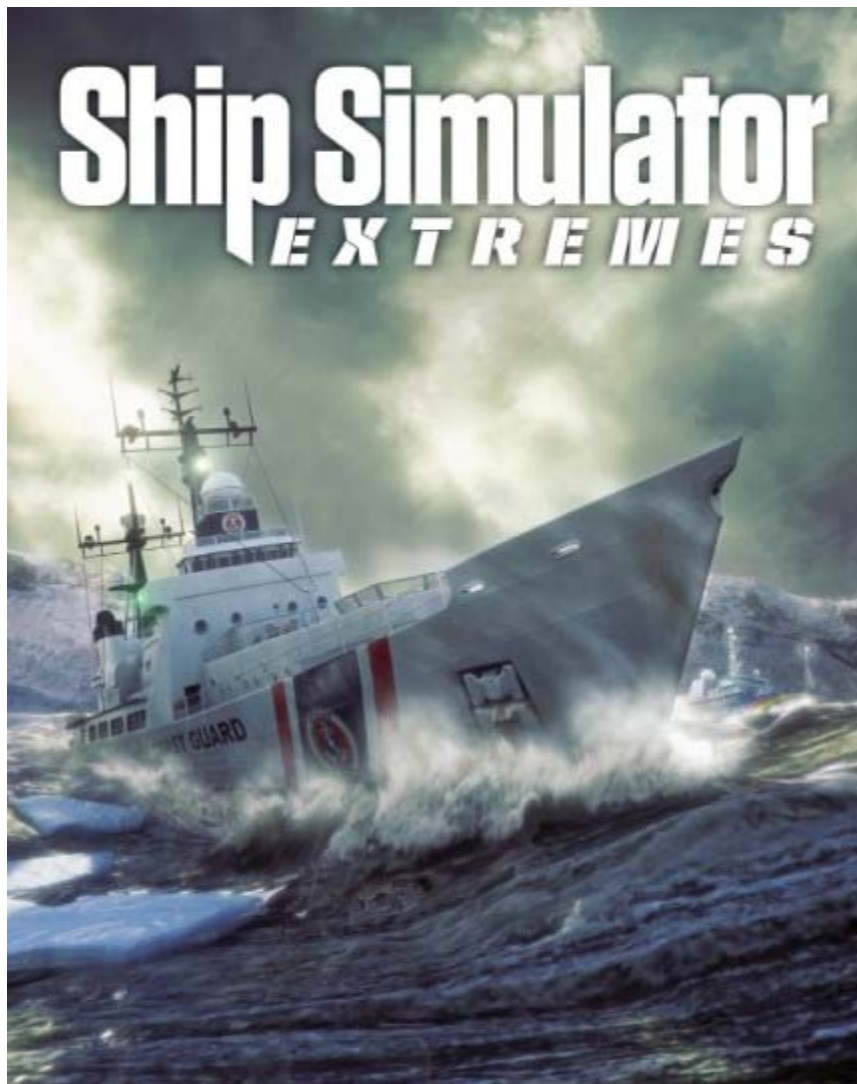


Mission Editor Guide

for



www.shipsim.com

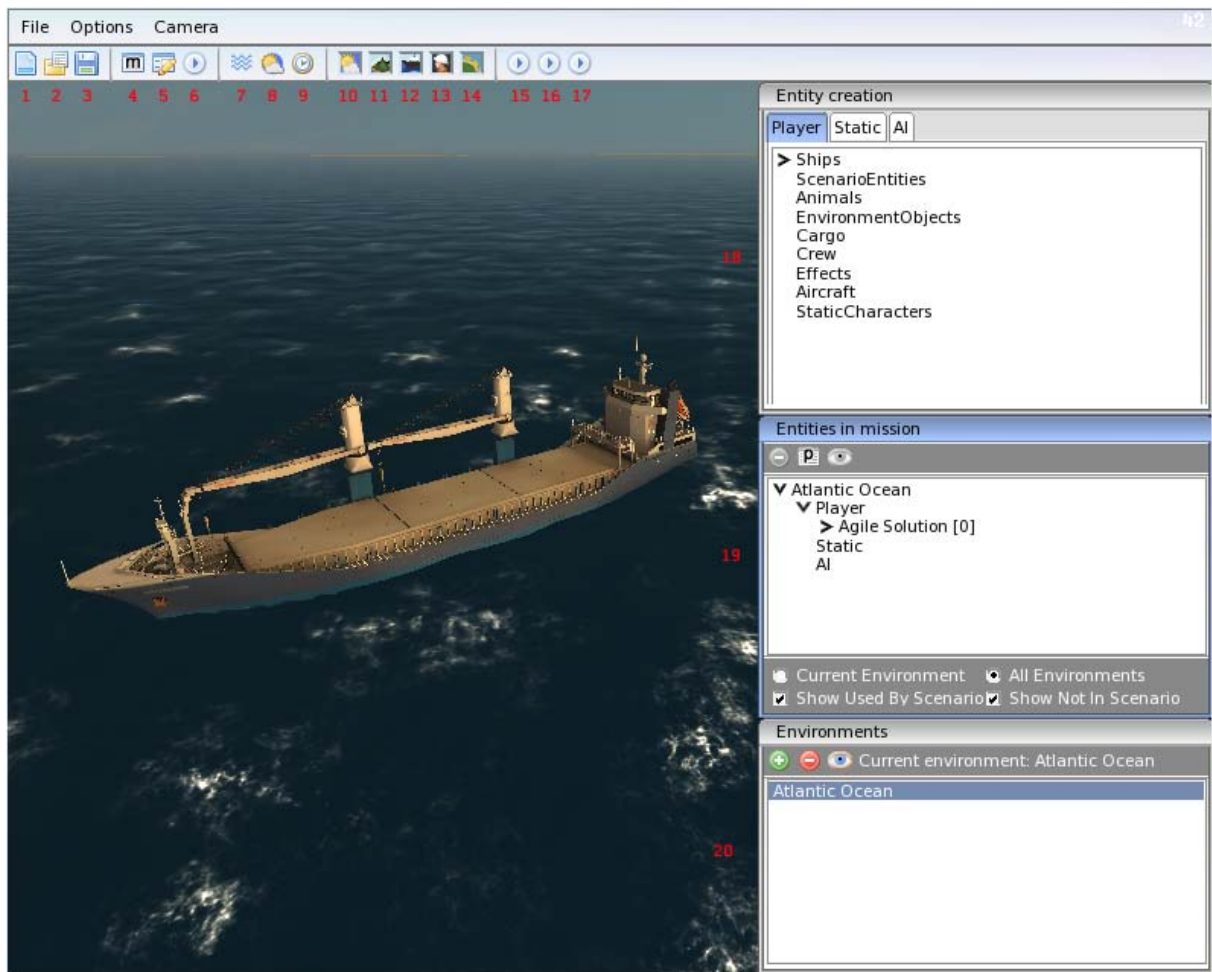


Table of Contents

Mission Editor Interface.....	3
Main Menu	3
Uploading/Sharing a Mission.....	3
Toolbar.....	3
Entity / Environment Section.....	4
Entity creation.....	4
Entities in mission	5
Environments	5
Viewport	5
Logic Editor.....	6
Logic Graph & Nodes	6
Logic Graph	6
Creating and Connecting Nodes.....	6
Selecting and Moving Nodes.....	7
OneTime	7
AND & OR Setups	7
Variables	7
Trigger Properties	8
Conditions.....	8
Actions	8
General	8
Ocean Settings.....	8
Dynamic Settings	8
Ocean Change.....	8
WeatherSim Settings	9
WeatherSim	9
Cloud Preset.....	9
Fog	9
Cloud Description	10
Summary.....	10
Tips.....	10
Weather Change	10
SkySim Settings.....	10
Date & Time	10
Location	10
Implementations	11
Objectives & Waypoints	11
Mooring	12
Towing & Bridge controls	12
Anchoring.....	12
Parenting	13
Fires	13
Timers	13
Convert Controller	14
Deployables	14
Teleport & Environment Switch	15
Photos.....	15

Characters.....	15
AI.....	16
Airplanes & Helicopters.....	16
Whales.....	17
Watercannons.....	17
Cinematic Camera.....	17
Final Note.....	17
Tutorial Mission.....	18

Mission Editor Interface



Main Menu




The main functions are accessible in the main menu, such as loading, saving, test mission, ect.

Uploading/Sharing a Mission

When you have created a mission in the editor, you can upload it for evaluation to the Ship Sim Extremes Mission Repository. You do this by going to **File -> Upload** in the File menu*. Your mission will then be tested by our custom mission managers. When it is playable it will be added to the mission repository and Ship Simulator Extremes players* will automatically notice the mission in their Single Mission menu in the game where it is ready for play.

**Only players that have registered their license key on the Ship Simulator community forum at forum.shipsim.com can upload and play shared custom missions.*

Toolbar

-  **New Mission (1)**
Start a new empty mission file
-  **Open (2)**
Load a saved mission
-  **Save (3)**
Save the current mission

The functions 1, 2 and 3 are also available from the main menu under **File**. If you want to save the mission under another name you can go to **File -> Save as...**



Mission Properties (4)

Here you can fill in the mission properties. Some of these will be visible in the game menu when selecting a mission such as the Title, Creator, Difficulty, Tags and Description.



Logic Editor (5)

The mission logic can be created in the logic editor. This will be further explained in the Logic Editor section.



Test Mission (6)

An important function for mission creation is testing a mission to check if your logic works. If the test mission button is pressed the nodes in the logic editor will be active. The node with a white circle around it is the currently active node. The test mission button will change into a stop testing button. All changes that are made while testing will be reversed when the stop testing button is pressed.

The functions 4, 5 and 6 are also available from the main menu under **Options**.



Ocean Settings (7)

The settings of the ocean can be changed here. This will be further explained in the Ocean Settings section.



WeatherSim Settings (8)

The settings of the weather can be changed here. This will be further explained in the WeatherSim Settings section.



SkySim Settings (9)

The settings of the date, time and location can be changed here. This will be further explained in the SkySim Settings section.



Toggle Sky (10)

Toggles the visibility of the sky and weather.



Toggle Terrain (11)

Toggles the visibility of the terrain.



Toggle Reflections (12)

Toggles the visibility of reflections in the ocean. This can improve performance when turned off.



Toggle Crew (13)

Toggles the visibility of characters. This can improve performance when turned off.



Toggle Terrain Borders (14)

Toggles the visibility of the terrain borders.



Moor (15)

A mooring line can be initialized at the start of a mission. Select a ship that is currently in the mission and then press this button to view the mooring points. Click with the **Left-Mouse Button (LMB)** on a mooring point and then click on any other mooring point that becomes visible. A mooring line will be established. To remove a mooring line, select the ship, press this button and then click with the **LMB** on the mooring point to which the mooring line is connected.



Tow (16)

A towing line can be initialized at the start of a mission. Select a ship that is currently in the mission and then press this button to view the towing points. Click with the **LMB** on a towing point and then click on any other towing point that becomes visible. A towing line will be established. To remove a towing line, select the ship, press this button and then click with the **LMB** on the towing point which the towing line is connected to.



Anchor (17)

An anchor can be initialized at the start of a mission. Select a ship that is currently in the mission and then press this button to view the anchor points. Click with the **LMB** on an anchor point. The ship will be anchored to the bottom of the ocean. To remove an anchor, select the ship, press this button and then click with the **LMB** on the anchor point which has an anchor line.

Entity / Environment Section

Entity creation (18)

The Entity creation section contains all entities that are available to place in a mission. These are listed under three tabs: **Player**, **Static** and **AI**. The arrow (>) indicates which entities are available for each tab. You can click on the arrow to expand the list.


- Player entities can be controlled by the player.
- Static entities can only be placed and not be controlled in any way.
- AI entities can be controlled by the AI system.





To place an entity, click on the desired entity in the list and then place it in the viewport by clicking on the desired location. (Note: an environment must be loaded to place an entity).

Once the entity is placed, a gizmo (Figure 1.1) will appear around it. You can move the entity by clicking and dragging the red cross. You can rotate the entity by clicking and dragging the green circle. The green triangle on the circle indicates the direction the entity will be facing. Ships are always placed at a height of zero. You can place other entities (for example fires) directly onto the surface of ships by holding the **M** key while placing the entity. Ships can only be placed inside the environment borders (orange lines in the environment), these borders will differ for each ship.

Entities in mission (19)

All entities that are placed in the mission are listed in this section. They are grouped per environment and type (Player, Static or AI). These groups can be collapsed or expanded by clicking on the arrow (>). The number between square brackets behind an entity is just an ID number to make the entity unique. If an entity has an arrow in front of it, it means that this entity has a deployable entity. An entity can be deleted by selecting it from the list and pressing the  button or the **Delete** key.

Every entity has a number of properties. To edit these, press the  button. A property window will pop up (Figure 1.2). Only a few of these properties are important. The location, rotation and scale of an entity can be manually changed in this window. Press the  button to expose and edit the values. (Note: If you want to use the gizmo after changing values manually to change the position/rotation of an entity then first deselect and reselect the entity, otherwise the position/rotation will be reset to its previous values.) Visible can be unchecked if you want the entity to be invisible at the start of the mission. The name of the entity can be changed in the textbox.

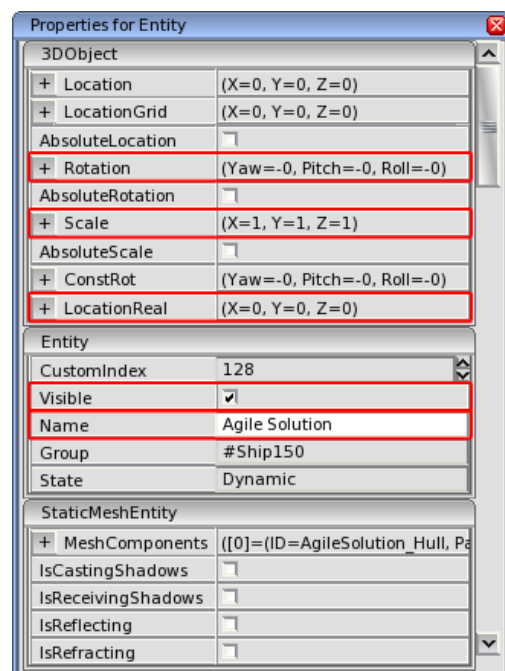



Figure 1.2: Properties for Entity Window

The camera can be focused on an entity by selecting it from the list and then pressing the  button or double clicking it entity in the list. Move the mouse cursor into the viewport. This function is also available from the main menu under **Camera -> Center on current entity**.

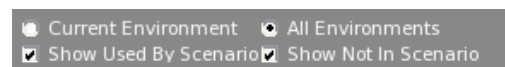





Figure 1.3: Settings for entity list

In missions with a lot of entities and environments it is easier to show only the ones you are currently working on. This can be configured by changing the settings at the bottom of the Entities in mission section (Figure 1.3).

Environments (20)

The environments section contains a list with all environments that are in the mission. An environment can be added by pressing the  button and selecting the environment from the window that will pop up. An environment can be deleted by selecting it from the list and pressing the  button. To start working with an environment it has to be loaded in the viewport. This can be done by selecting the environment from the list and pressing the  button.

Viewport

The viewport shows all the entities and environments visually. You can navigate around with the camera.

- **Move Camera:** **Up Arrow Key** = Move forward

Down Arrow Key = Move backward

Left Arrow Key = Move sideways to the left

Right Arrow Key = Move sideways to the right

- **Orbit Camera:** Orbit the camera around its focus point by holding the **Right-Mouse-Button (RMB)** and drag.
- **Zoom Camera:** Zoom the camera in and out by scrolling the **mouse wheel**.
- **Camera Focus Point:** The focus point of the camera can be changed by moving the camera or selecting one of the options from the main menu under **Camera**.
- **Select Entities:** Select entities by clicking with the **LMB** on an entity.
- **Minimize Windows:** Any window can be minimized by double-clicking with the **LMB** on the title bar. Double-click again to restore the window.

Logic Editor

The logic editor contains five sections (Figure 2.1). You can create the mission logic with these tools.

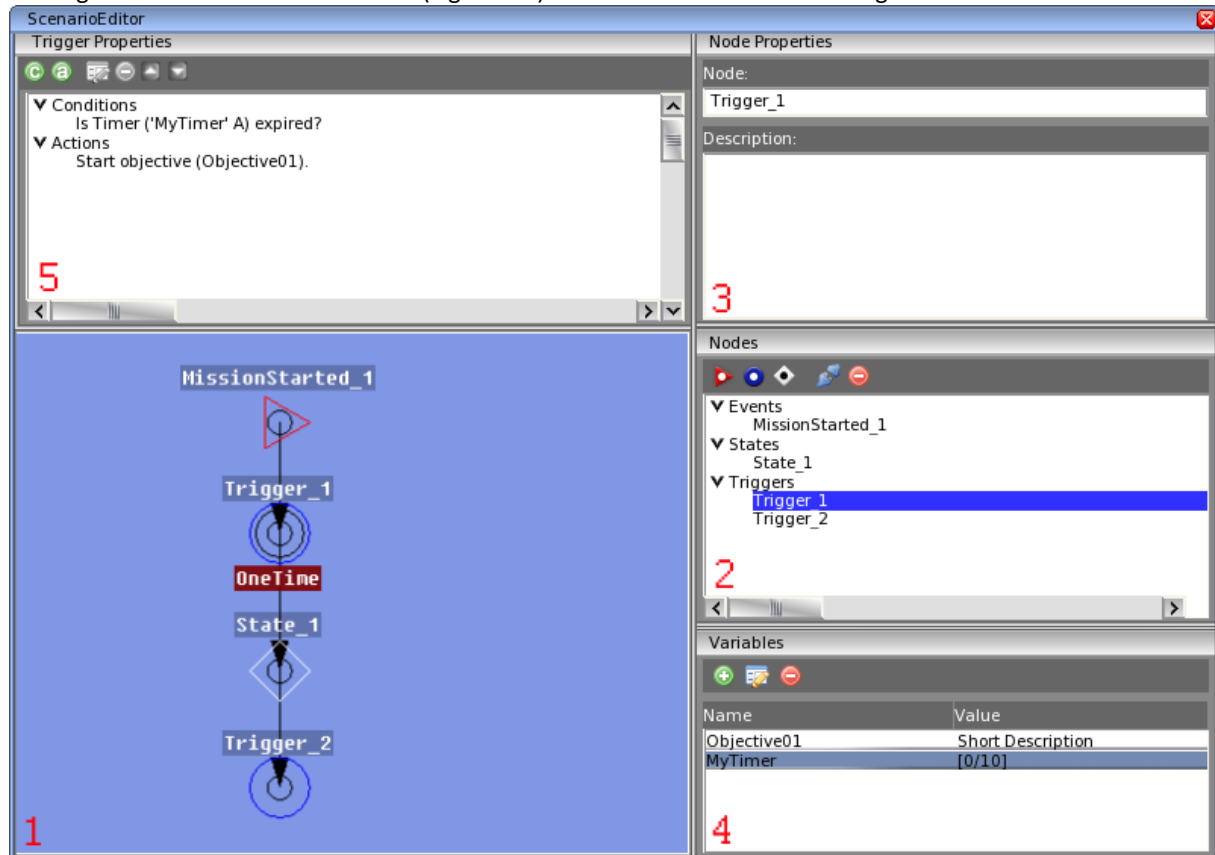


Figure 2.1: The logic editor

Logic Graph & Nodes (1 & 2)

Logic Graph

You can pan and zoom to navigate around in the logic graph.

- **Panning:** Press and hold the **RMB** then drag the mouse around, your view of the graph will move around. You can center the graph by pressing the **C** key.
- **Zooming:** You can zoom in and out using the **scroll wheel** of your mouse.

Creating and Connecting Nodes

You can create nodes in the logic graph to set-up a structure for a mission. Nodes can be **events**, **states** or **triggers**.

- ▶ **Events:** Events are the start of a node structure. Always start with an event, followed by a trigger.
- **States:** States will keep checking the conditions of the trigger they point to until the conditions are true.

- **Triggers:** Triggers are the most important. They contain conditions and actions that will define the mission.


Nodes can be created in two ways:

- Click with the **LMB** on the desired node button on the Nodes toolbar and place it by clicking in the logic graph.
- Hold a key (**E** for Events, **S** for States, **T** for Triggers) and click with the **LMB** in the logic graph to place it.

Nodes can be connected in two ways:

- Press and hold the **LMB** in the middle of a node (it will turn red when you hover over it with you mouse cursor), drag it to the desired node and release the **LMB**.
- Select a node, press and hold the **L** key and then select the desired node you want to connect to.

Nodes can be disconnected in two ways:

- Select the node you want to disconnect and press the  button on the Nodes toolbar.
- Select the node you want to disconnect and press the **D** key.

Selecting and Moving Nodes

Clicking a node using the **LMB** will select it. Combining this with the **Shift** key will also select all nodes that the selected node is connected too. You can add nodes to your selection by holding **Ctrl**. Another way of selecting multiple nodes is dragging the mouse using the **LMB**, this will create a selection rectangle and all nodes within it will be selected.

When you have one or more nodes selected, press and hold the **LMB** within (but outside the inner circle of) one of the selected nodes. Dragging the mouse will now move all selected.

OneTime

OneTime appears under a trigger when it is connected to an event or directly to another trigger. This means that the engine will check the conditions only once. If the conditions are not met when the engine checks them, the logic will stop at that trigger.

AND & OR Setups

You can split up a node structure. It is important to pay attention to the fact that you can create a split in such a way that it continues with a single or double chain (Figure 2.2). With the AND setup the engine will keep checking the conditions of both triggers. If one of the triggers conditions is met it keeps checking the other one. With the OR setup the engine will stop checking the other triggers conditions when one triggers conditions are met.

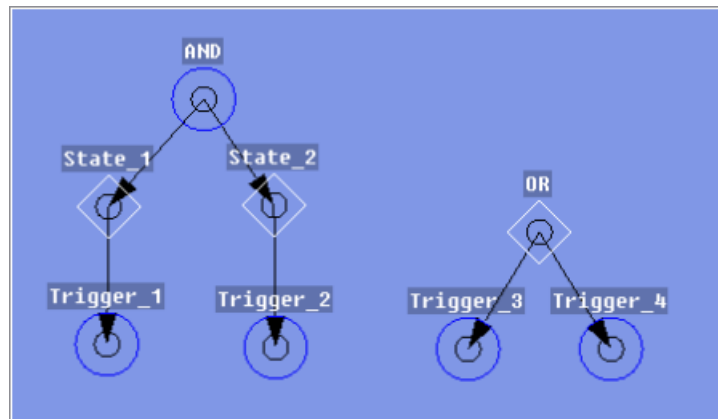



Figure 2.2: Node structure for AND & OR setup

Node Properties (3)

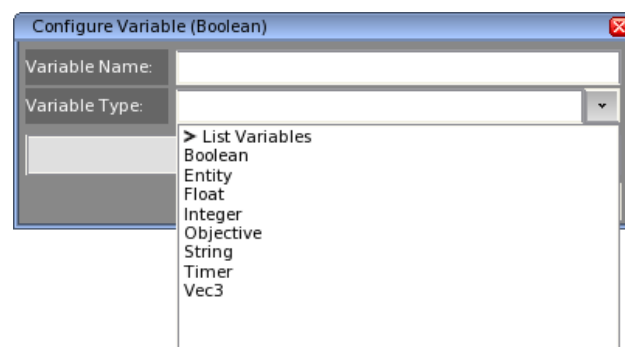
In this section you can rename the selected node and/or give it a description.

Variables (4)

In the variables section you can add, edit and remove variables from the logic. You can add a variable by pressing the  button. A window will pop up where you can give the variable a name and select a variable type.



List variables can only take a specific value depending on the type.

- **Boolean:** True or False variable.




- **Entity:** Define an entity as variable.
- **Float:** Number with decimals. Can be positive or negative.
- **Integer:** Whole number. Can be positive or negative.
- **Objective:** Objective variable to give players information what to do.
- **String:** Text variable.
- **Timer:** Timer variable.
- **Vec3:** Can store three float numbers

Figure 2.3: Configure Variable window with drop down list


You can edit a variable by selecting it from the list and pressing the  button. To delete a variable select it from the list and press the  button.

Trigger Properties (5)





Conditions

Conditions can be added to a trigger by pressing the  button. A window will pop up where you can select a condition from the drop down menu. Every condition will explain itself in the description. Most conditions offer the choice to select an entity from the scene or a variable that you defined in the variables section. Conditions that operate with values offer the choice to select a variable or give a direct numeric input.

Actions

Actions can be added to a trigger by pressing the  button. A window will pop up where you can select an action from the drop down menu. Every action will explain itself in the description. Actions have all kinds of inputs but they will point to the correct list to make it easier to select the correct one.

General

Conditions and actions can be edited by selecting the condition/action and pressing the  button. Conditions and actions can be deleted by selecting the condition/action pressing the  button. The order of conditions/actions can be changed by selecting the condition/action and pressing the  or  button. The order of conditions does not matter so they don't need to be changed. Actions however will be performed in the order they are listed so it is important to place them in the order you want them to be performed.

Ocean Settings

Dynamic Settings

The ocean can be customized by changing the sliders in the **Dynamic Settings** tab (Figure 3). The small, middle and large waves can be changed by height and speed. The **WaveScale** changes the overall scale of the ocean.

Ocean Change

The ocean settings can be changed during the mission by using an action. This action is called **Ocean – Blend waveheights**. You can set the values of the height for each of the waves and blend to these values in the amount of time you set in this action.

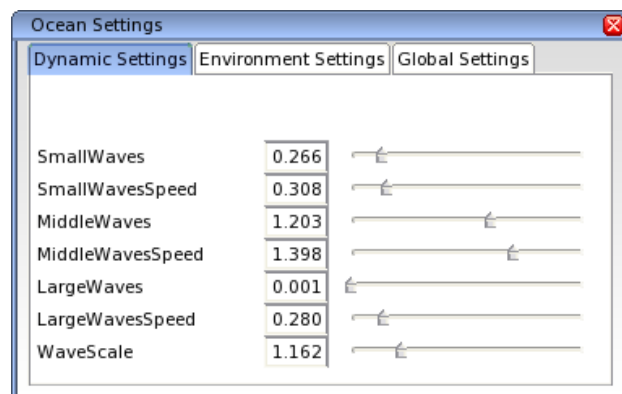


Figure 3: Ocean Settings window, Dynamic Settings tab

WeatherSim Settings

WeatherSim

The WeatherSim tab contains basic weather settings. Any weather condition can be set using the sliders under **Weather Settings** (Figure 4.1). The WeatherSim automatically finds the type of precipitation that belongs to the current temperature and precipitation level. **Cloud Generation Settings** determines what kind of cloud will be generated on the edges of the environment. There are 3 heights, and for each height the ratio between the two cloud types on that height as well as the cover you want can be set. Note that these Cloud Generation Settings are not directly visible as it only handles automatic spawning of new clouds. Also note that for any precipitation you will have to add clouds that can emit this precipitation! For example, rain can only be emitted from middle and low clouds.

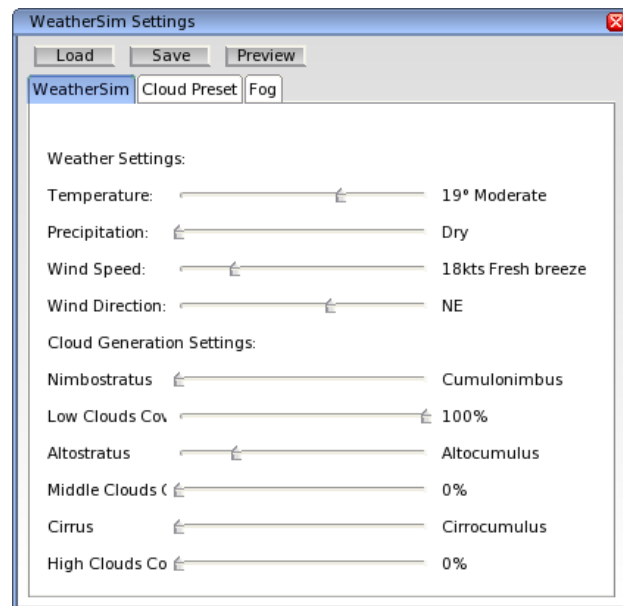


Figure 4.1 WeatherSim Settings window, WeatherSim tab

Cloud Preset

The Cloud Preset tab can be used for setting up the initial clouds. First select the layer (height), then the type you want to add. The letter between brackets behind the type is the color, R for Red, G for Green. The cloud will show up in that color on the preview window right (Figure 4.2). The yellow color indicates overlapping of the clouds. The settings are explained at the bottom of the window. The buttons:

- **Add:** Adds the clouds with the defined settings.
- **Clear:** Clears all clouds of the selected type.
- **Relax:** Spread or merges the clouds, depending on the type.
- **Update:** Updates the clouds to 5 minutes, also including wind.

Preview will speed up time (x10) and use the cloud generation settings to update the clouds. If you stop the preview the weather will be restored to the point where you started the preview. (Note: all changes made during preview will be lost, also preview does not give perfect results, just an idea how the clouds will evolve).

Load/Save serializes both the Weather settings as initial clouds. This is not required for the mission, but it can be useful.

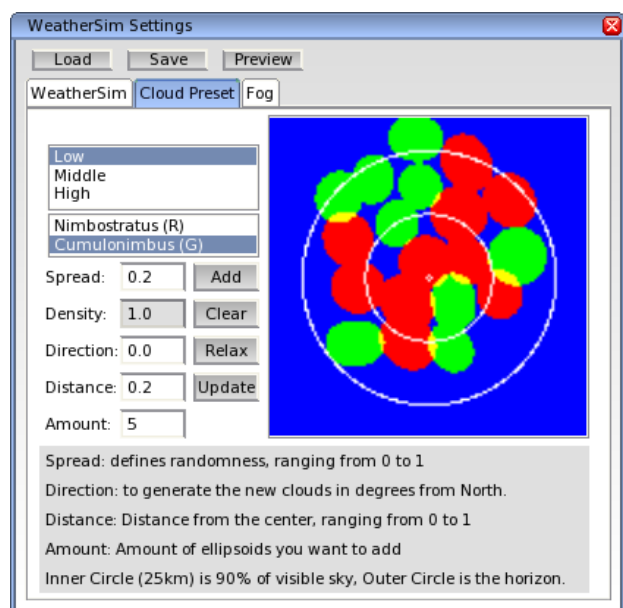


Figure 4.2: WeatherSim Settings window, Cloud Preset tab

Fog

The Fog tab can be used for adding fog to the environment. There are two modes which can be toggled by pressing the **Switch Mode** button. Default it is on automatic, with only one slider which controls the fog density. You can switch to Manual mode which gives you more control over the fog like different settings between ground and sky fog, fog blending and fog color.

Implementations

Objectives & Waypoints

Objectives give the player information about what to do. Without objectives there is no mission. Objectives can be added in the Variable section as described earlier in this document. An objective variable has three inputs.

- **Name:** Give the objective a simple name like "Objective01".
- **Short Description:** Describe in a short sentence what the player has to do. This sentence will be visible to the player in the game.
- **Long Description:** Describe in more detail what the player has to do. You can also create some sort of storyline here. This description will be visible when the player expands the objective information.

Objectives only use actions to work in the game. The following actions are available for objectives:

- **Objective – Start:** Start the specified objective.
- **Objective – InfoMessage:** Show given message for specified objective.
- **Objective – Complete:** Clear specified objective with given message.
- **Objective – Fail:** Fail specified objective with given message.

The function of Start, Complete and Fail are obvious. InfoMessage gives the player a short message during an objective. If the specified objective is not active the message will not be shown.

To give the player a visual representation of where they need to go a waypoint on the chart can be set, these are green stars. This can be done by changing the visibility of a SphereAreaEntity or a BoxAreaEntity. These can be found in the **Entity creation** section under the **Static** tab in the **LogicEntities** list. The visibility of an entity can be controlled by the action **Entity – Set Visibility**. Remember to uncheck the visible property of the waypoint otherwise all waypoints will be visible at the start of mission.

Example shown in Figure 5.1

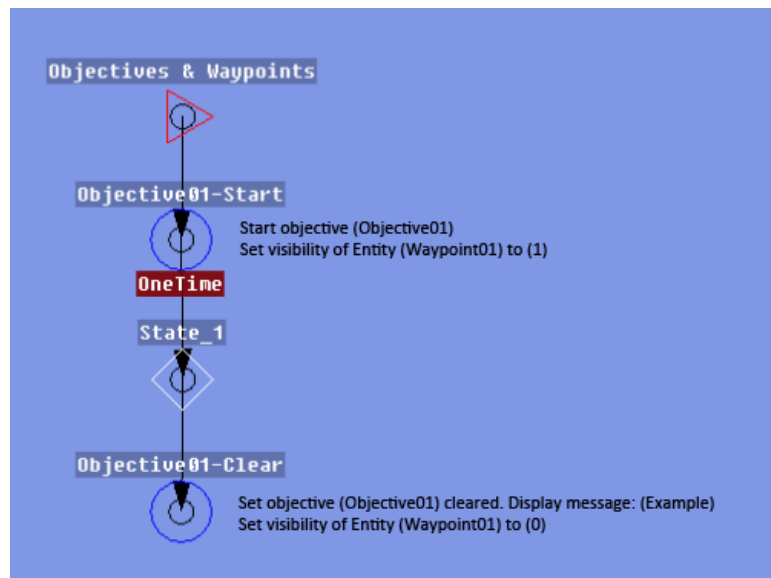


Figure 5.1: Example of an objective and waypoint

There are also actions to complete or fail a mission. They are needed for every mission so that the player can actually finish the mission.

- **Mission – Clear/Win!:** Clear the current mission.
- **Mission – Lose/Fail!:** Fail the current mission.

Mission – Clear/Win! is always placed at the end of a node structure.

Mission – Lose/Fail! can be placed anywhere in the mission where you think the player should fail the entire mission.

Mooring

Often mooring will be done by the player in-game but there are a few situations where you want to force mooring or unmooring. A ship can also be moored from start of a mission which is explained in the **Toolbar** section.

Conditions

- **Ship – Moored:** Checks if the specified ship is moored to a moorage point.
- **Ship – Moored2:** Checks if the specified ship is moored to two moorage points.
- **Ship – Moored with entity:** Checks if the specified ship is moored with the specified entity.
- **Ship – Not Moored:** Checks if the specified ship is not moored to anything.

Actions

- **Ship – Connect To Moorage Point:** Connects the closest moorage point on the specified ship to the specified moorage point.
- **Ship – Disconnect All Mooring Lines:** Disconnects all mooring/towing lines from the specified ship.

There are situations where you want to automatically moor a ship to the dock. For example, when the player has towed another ship to its destination and the towed ship has to be moored to the dock.

An example for use of disconnecting all lines automatically is when a ship is moored to the playership and this turns into AI and sails away from the playership. Since the player can't control AI ships, you want to force disconnection of the mooring lines.

Towing & Bridge controls

Often towing will be done by the player in-game but you can initialize towing lines from the start of mission which is explained in the **Toolbar** section.

Conditions

- **Ship – Towing:** Checks if the specified ship is towing any ship.
- **Ship – Towing Ship:** Checks if the specified ship is towing a specified ship.
- **Ship – Not Towing:** Checks if the specified ships is not towing anything.

Actions

- **Ship – Disconnect All Mooring Lines:** Disconnects all mooring/towing lines from the specified ship.

Towed ships are sometimes broken or need to be towed out of the harbor. In both cases they don't use their engines. With the action **Ship – BridgeComponents – SetEnabled** you can disable any control by specifying the ship, the component index of the control and an enable value (0=disabled, 1=enabled). The component index of the controls can be made visible by pressing **Ctrl + Shift + A**. The component indexes will be visible on the ship that is selected. (Note: Only Player and Static ships can be towed)

Anchoring

Often anchoring will be done by the player in-game but you can initialize anchors from the start of mission which is explained in the **Toolbar** section.

Conditions

- **Ship – Anchored:** Checks if the specified ship is anchored
- **Ship – Not Anchored:** Checks if the specified ship is not anchored.

Actions

- **Ship – Drop Anchor:** Drops all anchors
- **Ship – Hoist Anchor:** Hoists all anchors

An example for use of forced hoisting of anchors is when a broken ship is out on sea and needs to be towed. Once the player connects the towing line you can force the anchors of the towed ship to hoist.

Parenting

Parenting attaches two entities to each other. The child always follows the movement of the parent. Parenting can be used for attaching areas or fires to ships. There are two actions which are related to parenting:

- **Entity – Attach to entity:** Attaches the specified child entity to the specified parent entity.
- **Entity – Detach:** Detaches the specified entity from the parent entity.

Fires

Fires can be found in the **Entity creation** section under the **Static** tab in the **Effects** list. Fires need to be parented to another entity in order to make them extinguishable. The default scale of a fire is 1. To spread the fire over a larger area, the scale over X and Z can be edited in the properties of the fire. How larger the area how longer it takes to extinguish a fire. An example is shown in Figure 5.2.

Conditions

- **Fire – Alive:** Checks if the specified fire is still burning.
- **Fire – Extinguished:** Checks if the specified fire is extinguished.

Actions

- **Fire – Start:** Starts the specified fire.
- **Fire – Stop:** Stops the specified fire (Note: will not be marked as extinguished).
- **Fire – Smoke Only:** Starts the specified fire with only smoke particles.

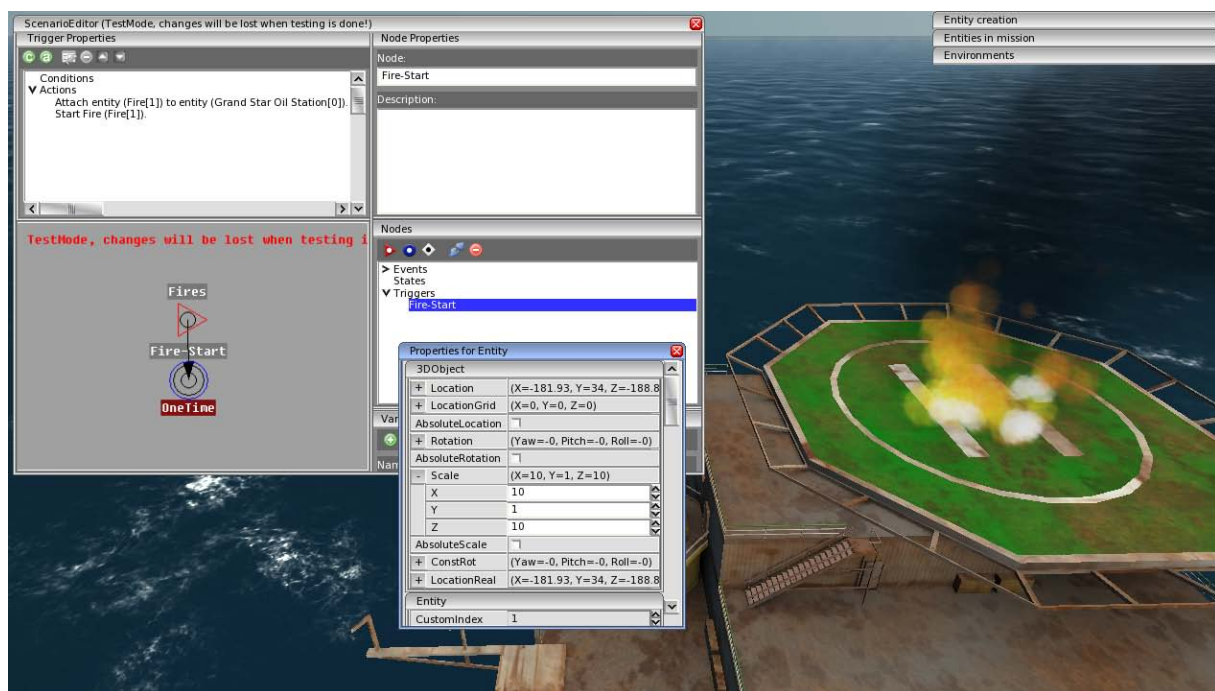


Figure 5.2: Example of used actions and properties for a fire

To make a fire spreading you can start fires by using a timer. To make it even more complex you can have a fire spreading and let it intercept by the player so that it spreads not all the way by checking if the previous fire is still burning. It's up to you.

Timers

Timers can be used for all kinds of things. For example, an objective needs to be completed within a time limit, or the delay between fires, etc. A timer can be added as a variable in the **Variables** section. The value you give the timer is the target value. The current value of a new timer is always 0 and starts counting when the **Timer – Start** action is used.

Conditions

Timer – Expiration Check: Checks if the specified timer is expired.

Timer – Time Passed Check: Check if the specified timer has passed the specified time.

Timer – Time Remaining Check: Checks if the specified timer has the specified time left.

Actions

Timer – Start: Starts the specified timer.

Timer – Stop: Stops the specified timer, current value becomes 0.

Timer – Pause: Pauses the specified timer, continue with start action.

Timer – Set Current Value: Sets the current value of the specified timer to the specified time.

Timer – Set Target Value: Sets the target value of the specified timer to the specified target value.

Convert Controller

Entities can be added as **Player**, **Static** or **AI** from the **Entity creation** section. However this can be changed during the mission. The entities can be converted to one another. This we call Convert Controller. The action **Entity – Convert Controller** makes it happen. It can convert any type of controller to another by specifying the entity and the controller type to convert to.

Deployables

Often the player needs to deploy vessels in-game but this can also be done automatically if necessary. When a player deploys a vessel him-/herself the deployed vessel will become a playership automatically. However if you let a vessel deploy by an action, it needs to be converted manually to a Player entity. When boarding a vessel with an action, it needs to be converted manually to a Static entity. The **Convert Controller** section describes how to do this.

Conditions

Ship – Vessel boarded: Checks if the specified vessel is boarded on the specified ship

Ship – Vessel deployed: Checks if the specified vessel is deployed from the specified ship.

Actions

Ship – Board Vessel: Boards the vessel with the given ShipID on the specified ship.

Ship – Deploy Vessel: Deploys the vessel with the given ShipID from the specified ship.

The ShipID starts at 0 for every vessel on a ship. If there is only one deployable the ShipID is always 0. Test the action with a ShipID to determine which vessel has which ShipID. An example is shown in Figure 5.3.

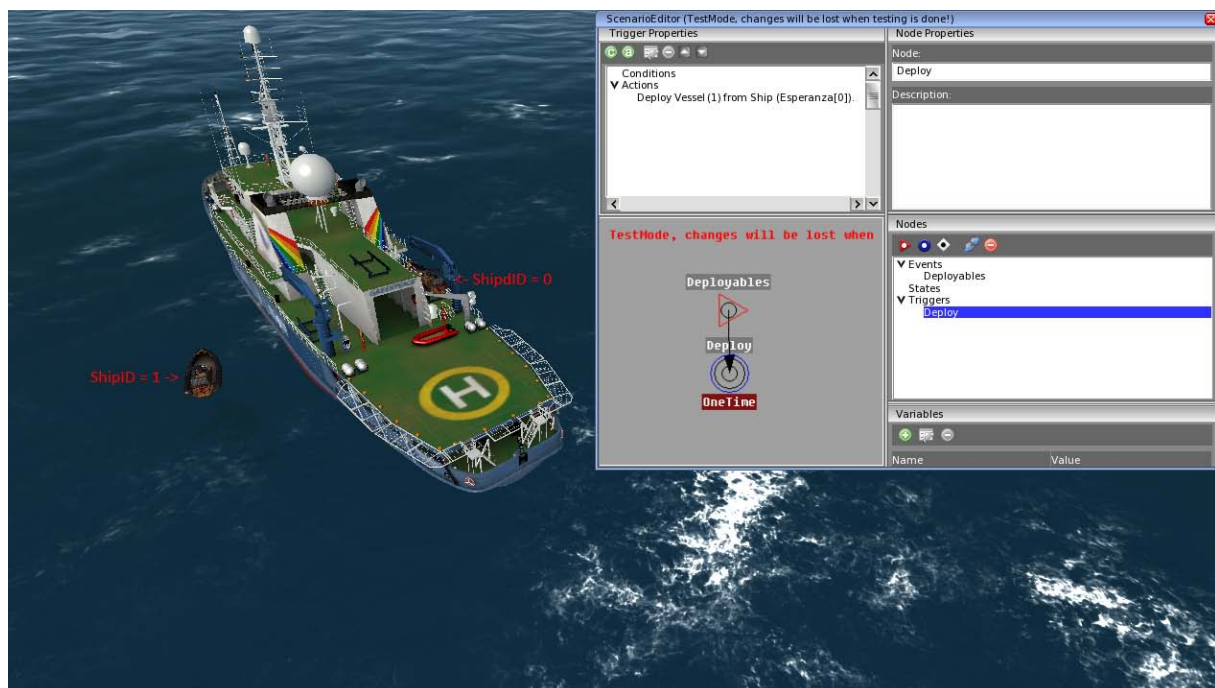


Figure 5.3: Example of used action with the ShipID

Teleport & Environment Switch

It is possible to teleport ships to another location in the same environment or teleport to a location in a different environment. There are three actions which can perform this.

- **Entity – Teleport:** Teleports the specified ship to the specified location
- **Entity – Move to Environment:** Teleports the specified ship to the specified location in the given environment.
- **Environment – Switch to different Environment:** Teleports the specified ship to the specified location in the given environment with a given travel time (this affects the actual daytime).

The **Environment – Switch to different Environment** action is used to switch the game to a different environment. The ship that has to be specified must be player ship in order to work.

Locations are set with a PointEntity which can be found in the **Entity creation** section under the **Static** tab in the **LogicEntities** list. The direction of the PointEntity determines the direction the ship is facing when teleported.

Photos

Players can take photos during the mission. These photos will be saved to the hard drive. There are a three conditions for photos that can be used.

- **Photo – Taken:** Checks if a photo has been taken.
- **Photo – Taken Photo Contains Entity:** Checks if a photo has been taken with the specified entity on it.
- **Photo – Taken Photo Does Not Contain Entity:** Checks if a photo has been taken without the specified entity on it.

If you want for example a building that needs to be photographed than you can place a PointEntity inside the building and use that in the condition.

Characters

Character can be placed on a ship and fall overboard. Thereafter they need to be rescued. Characters can be found in the **Entity creation** section under the **AI** tab in the **Crew** list. To use a character for a mission they need to be placed in the environment first. It doesn't really matter where because one of the character actions needs to be used in the first trigger of a node structure. (Note: Don't place too many characters as they take up a lot of memory). All the characters in the Crew list have animations so these are heavier for the engine than the characters found in the **StaticCharacters** list under the **Static** tab. These characters don't have animations but they have different poses, so can be placed on ships as a kind of decoration.

Actions

- **Character – Overboard:** Throws a character overboard on the specified location. The character will follow the specified ship when that ship is close enough.
- **Character – Transfer To Ship:** Transfers the specified character to the specified ship.

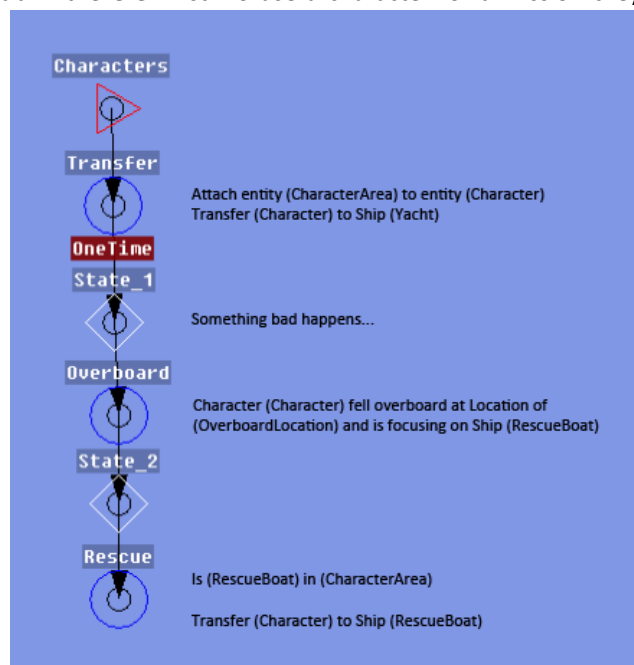


Figure 5.4: Example of a character going overboard and rescued

In the first trigger of node structure the characters needs to be placed on a ship with the **Character – Transfer To Ship** action or thrown overboard with the **Character – Overboard** action. This is because the characters will not do anything if they are just placed in the environment. Example is shown in Figure 5.4.

AI

AI ships can be added from the **Entity creation** section under the **AI** tab in the **Ships** list. AI ships need actions to do actually something. These actions are:

- **AIShip – Abandon Object and go to Goal:** Commands specified AI ship to first sail away from the specified object and then head to the specified goal.
- **AIShip – Apply full speed:** Commands specified AI ship to sail at full speed
- **AIShip – Apply speed:** Commands specified AI ship to sail at specified speed.
- **AIShip – Approach object:** Commands specified AI ship to approach the specified object.
- **AIShip – Follow object:** Commands specified AI ship to follow specified object.
- **AIShip – Go to goal:** Commands specified AI ship to go to the specified goal. AI ship will stop sailing when the goal is reached.
- **AIShip – Go to goal and respawn:** Commands specified AI ship to go to the specified goal and respawn from its start position when the goal is reached. The AI ship will keep heading for the goal.
- **AIShip – Ignore object:** Commands specified AI ship to ignore specified object. Entities may collide when using this action.
- **AIShip – Set Accelerate from Start:** Commands the specified AI ship to have the preferred speed as the initial speed (Set DoAccelerate to a value of 0).
- **AIShip – Set Decelerate to Final Goal:** Commands specified AI ship to decelerate before reaching its goal. Normally the AI ship will not decelerate to its goal.
- **AIShip – Set preferred speed:** Commands specified AI ship to try and sail the specified speed. If something is in the way it will change speed according to the situation.
- **AIShip – Stop:** Commands specified AI ship to stop sailing.

PointEntities are used to define goals. An object can be a PointEntity as well as a ship.

Airplanes & Helicopters

Airplanes and Helicopters are using an interpolation system to fly from one point to another. They can be found in the **Entity creation** section under the **Static** tab in the **Aircraft** list. To set up a flight route you have to use the following conditions and actions.

Conditions

- **Entity – Other Entity Within Range:** Checks if the specified entity is inside the specified radius of the other specified entity.

Actions

- **Entity – Interpolate – Set Target:** Sets a specified target for the specified entity.
- **Entity – Interpolate – Start:** The specified entity starts moving towards its target. If there is not a target yet nothing will happen and this action needs to be used again after a target is set.

PointEntities are used to define targets. The speed of an airplane/helicopter depends on distance between targets. How bigger the distance between targets, how faster the entity will move. An example is shown in Figure 5.5. The interpolation system can also be used for other entities.

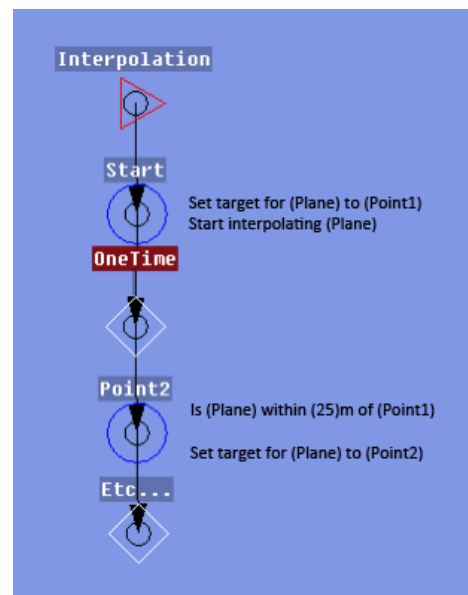


Figure 5.5: Example of the interpolation system

Whales

Whales use their own interpolation system. It basically works the same as the airplanes/helicopters interpolation but with a few more actions and a minor difference, which is the speed. In this interpolation system the speed of the whale is controllable by an action. The whale can be found in the **Entity creation** section under the **AI** tab in the **Animals** list.

Conditions

- **Entity – Other Entity Within Range:** Checks if the specified entity is inside the specified radius of the other specified entity.

Actions

- **Whale – Add target:** Sets a specified target for the specified whale.
- **Whale – Apply speed:** Applies the specified speed to the specified whale.
- **Whale – Dive:** Plays the dive animation on the specified whale.
- **Whale – Set Whaler:** All whales will flee from the specified whaler.
- **Whale – Start:** The specified whale starts moving towards its target. If there is not a target yet nothing will happen and this action needs to be used again after a target is set.

Watercannons

Watercannons will be often controlled by the player in-game, but if you want AI ships (for example whalers) to spray water at certain spots or ships, you need to use two actions to make that happen.

- **Ship – Watercannon – Enable:** Enables (1) or disables (0) the watercannon with the specified index on the specified ship.
- **Ship – Watercannon – Set Target:** Sets the target of the watercannon with the specified index on the specified ship to the specified entity.
- **Ship – Watercannon – Set Power:** Sets the power of the watercannon with the specified index on the specified ship to the specified power. A realistic power is between 30 and 40, it depends on the ship. Test it yourself to find out which power you need.

The target needs to be a PointEntity or a ship. It does not matter if you set the target first or if you enable the watercannon first. The watercannon index starts at 0 at every ship, so if there is only one watercannon the index will always be 0. If there are more watercannons on a ship you have to test which watercannon belongs to which index.

Cinematic Camera

A cinematic camera can be used to make a short cut scene of an important event (for example a ship that catches fire). The cinematic camera can be found in the **Entity creation** section under the **Static** tab in the **LogicEntities** list. The cinematic camera can be controlled by several of actions.

- **CinematicCam – Face Location:** Set the specified camera to face the location of the specified entity at a specified height over a specified time.
- **CinematicCam – Set Height:** Set the specified camera to a specified height over a specified time.
- **CinematicCam – Switch Back to Previous Cam:** Switches view back to the previous camera.
- **CinematicCam – Switch To Cinematic Cam:** Switches view to the specified camera.
- **CinematicCam – Transport Cinematic Camera:** Transports the specified camera to a specified destination at a specified height over a specified time. The camera inherits the rotation of the destination point.
- **CinematicCam – Transport and face location Cinematic Camera:** Transports the specified camera to a specified destination at a specified height facing a specified location over a specified time.

The location and destination points must be PointEntities. Use a timer to switch back to the player view with the **CinematicCam – Switch Back to Previous Cam**.

Final Note

There are a lot more conditions and actions available than described in this guide. The description of all those conditions and actions are available in the mission editor itself and it gives a clear idea of what they do. There are several ways to do the same thing. Create the structure you think works the best and test it regularly.

Tutorial Mission

This step by step guide is written assuming you have read the **Mission Editor Interface** and **Logic Editor** section so you basically know how to navigate around and add entities.

- Add and load the **Hamburg** environment.
- Add the **Bugsier 2** from the **Player** tab **Ships** list and place it at the location shown in the image below.



- Open the **Mission Properties** and fill in a **Title** and **Creator**. Choose the **Bugsier 2** as the **Start Ship**.
- **Save** the mission. (Remember to regularly save your mission)
- Open the **Logic Editor** and add the following **Objective** variables:

Variable Name: Objective01

Short description: Sail to the Ocean Prince

Long Description: Unmoor your ship from the dock and sail towards the Ocean Prince.

Variable Name: Objective02

Short description: Tow the Ocean Prince inside the harbour

Long Description: Connect a towing line to the Ocean Prince and tow it to the designated location inside the harbour.

Variable Name: Objective03

Short description: Return to the Tug Station

Long Description: Disconnect the towing line and sail back to the Tug Station. Moor at the dock.

Variable Name: Objective04

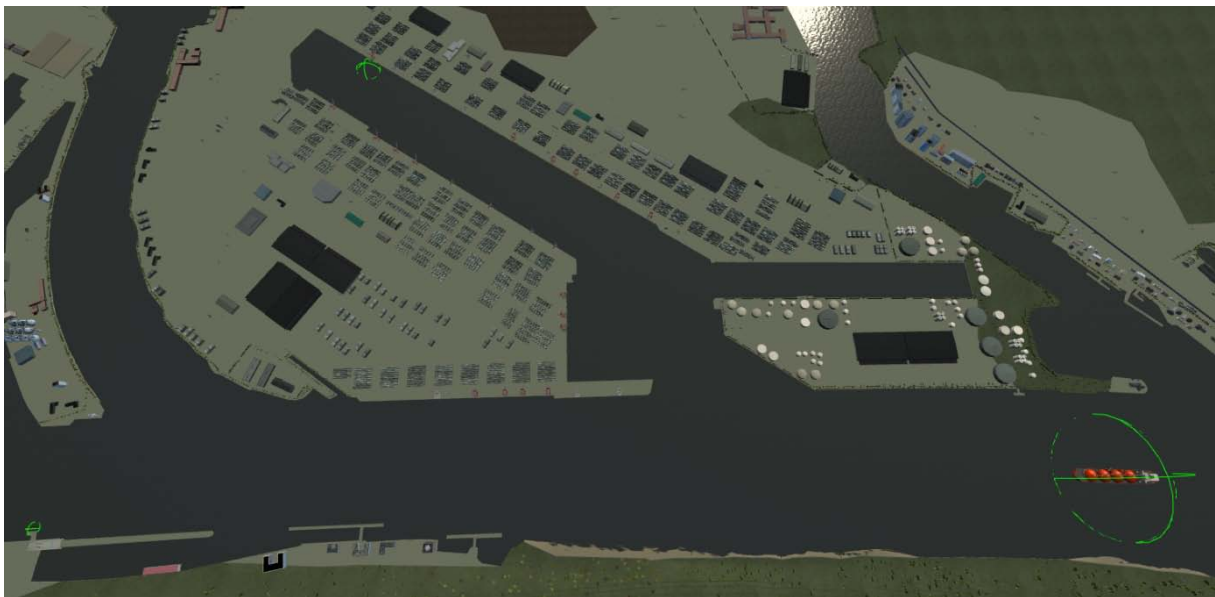
Short description: Pick up the pilot

Long Description: Sail towards the Ocean Prince and pick up the pilot.

- Place the other ships that are needed for this mission. Select the **Ocean Prince** and the **Apollo** from the **Static** tab **Ships** list and place them at the locations shown in the image below.



- We now need to add the waypoints for the objectives. Select the **SphereAreaEntity** from the **Static** tab **ScenarioEntities** list. Place the first waypoint in the middle of the **Ocean Prince**. Go to the **Properties** of the **SphereAreaEntity**, give it a **Radius** of **200**, rename it to **Waypoint1&4** and uncheck the **Visible** property. Place the second waypoint inside the harbour (see image below), give it a **Radius** of **50**, rename it to **Waypoint2** and uncheck the **Visible** property. Place the third waypoint in the middle of the **Bugsier 2**, give it a **Radius** of **20**, rename it to **Waypoint3** and uncheck the **Visible** property. It should now look like the image below.



- For **Objective01** the **Bugsier 2** needs to be moored to the dock since the player needs to unmoor the ship. Add a **Bollard02** from the **Static** tab **EnvironmentObjects** list. Go to the **Properties** of **Bollard02** and give the **LocationReal Y** a value of **3**. Do this three times, then move and rotate the bollards to the positions shown in the image below.



- Select the **Bugsier 2** and press the **Mooring** button on the **Toolbar**. Select a mooring point and connect it to bollard. Do this three times, connect the mooring lines as shown in the image below.

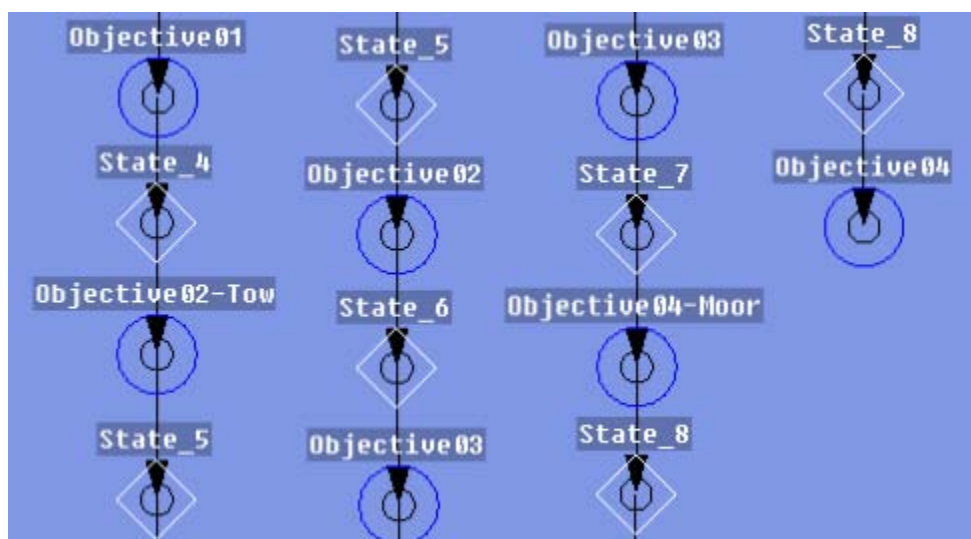
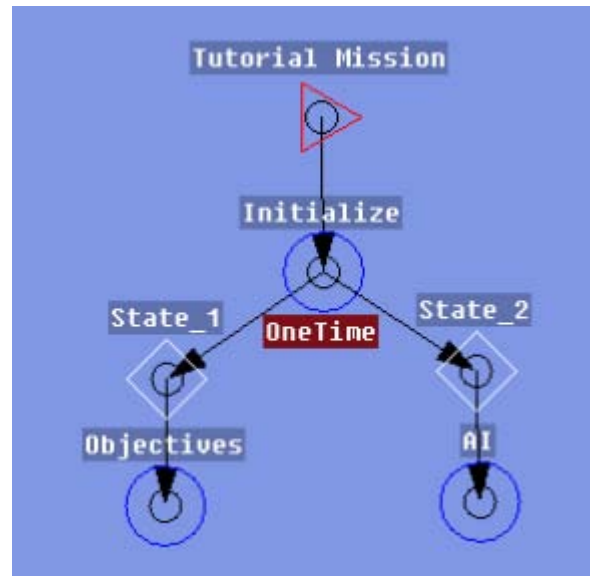


- All entities that are needed for this mission are now placed. It is time to create the mission logic in the **Logic Editor**. Open the **Logic Editor** and place an **Event** node and a **Trigger** node. Rename the **Event** node to your **mission name** and the **Trigger** node to **Initialize**. Connect the **Event** node the **Trigger** node. It should now look similar to the image on the right.
- The **Initialize Trigger** should contain all actions that need to be performed at the start of a mission. For this mission that is only one. Add the action **Entity – Attach to Entity** to the **Initialize Trigger**. Select the **Waypoint1&4** as the **ChildEntity** and the **Ocean Prince** as the **ParentEntity**. You can select those entities either from the viewport or from the **Entities in mission** section. The **Waypoint1&4** will now be attached to the **Ocean Prince** once the mission starts. You can test this by pressing the **Test Mission** button on the **Toolbar**. Do not forget to **Stop Testing**



when you continue working on the mission. To keep the graph organized split up the node tree by adding **two States** and **two Triggers**. Rename to **Triggers** to **Objectives** and **AI**. Connect the **Initialize Trigger** to the **two States**. Connect one **State** to the **Objectives Trigger** and the other one to the **AI Trigger**. It should now look like the image on the right.

- Add an **Objective – Start** action to the **Objectives Trigger** and select **Objective01** as the **Objective**. Make the first waypoint visible by adding the **Entity – Set Visibility** action to the **Objectives Trigger**. Select the **Waypoint1&4** as the **Entity** and set the Value to 1.
- Add a **State** and a **Trigger**, rename the **Trigger** to **Objective01**. Connect the **Objectives Trigger** to the **State** and the **State** to the **Objective01 Trigger**. Add the **Area – Contains Entity** condition to the **Objective01 Trigger**. Select the **Bugsier 2** for **Entity A** and **Waypoint1&4** for **Area B**. You could add the **Ship – Not Moored** condition, but this is not really necessary because the **Bugsier 2** can't sail anyway when it is moored.
- Now we need some actions for the **Objective01 Trigger** to decide what needs to happen if the conditions are met. Add the action **Objective – Cleared** to the **Objective01 Trigger** and select **Objective01** as the **Objective**. For the **Message** type **Great job! You have reached the Ocean Prince**. Add the action **Objective – Start** to start **Objective02**. Add the action **Entity – Set Visibility** two times. One to set **Waypoint1&4** to invisible, the other to set **Waypoint2** to visible. The **Objectives** tree should now look similar to the image on the right.
- You should now have an idea of how the rest of the **Objectives** tree is going to look. It is basically a **State** followed by a **Trigger** every time, sometimes more **Triggers** for one objective, for example if you want to give the player some **InfoMessages** during the objectives. The images below show the rest of the node structure. Recreate it. Then look through the conditions and actions after the image and add them as described.



Objective02-Tow TriggerConditions

Ship – Towing Ship: ParentShip = Bugsier 2, ChildShip = Ocean Prince

Actions

Entity – Convert Controller: Entity = Ocean Prince, Controller Type = Player

Ship – BridgeComponent – SetEnabled: Entity = Ocean Prince, Componentindex = 2, Enabled = 0

Ship – BridgeComponent – SetEnabled: Entity = Ocean Prince, Componentindex = 33, Enabled = 0

Ship – BridgeComponent – SetEnabled: Entity = Ocean Prince, Componentindex = 34, Enabled = 0

The **Componentindex** can be toggled on/off by pressing **Ctrl + Shift + A**. Select the ship you want to see the component indexes of. All controls except for the steering wheel should be disabled on ships that are being towed. See image below.

**Objective02 Trigger**Conditions

Ship – Towing Ship: ParentShip = Bugsier 2, ChildShip = Ocean Prince

Area – Contains Entity: Entity A = Ocean Prince, Area B = Waypoint2

Actions

Objective – Cleared: Objective = Objective02, Message = Great towing!

Entity – Set Visibility: Entity = Waypoint2, Value = 0

Objective – Start: Objective = Objective03

Entity – Set Visibility: Entity = Waypoint3, Value = 1

Entity – Convert Controller: Entity = Ocean Prince, Controller Type = Static

Objective03 TriggerConditions

Ship – Not Towing: Ship = Bugsier 2

Area – Contains Entity: Entity A = Bugsier 2, Area B = Waypoint3

Ship – Moored2: Ship = Bugsier 2

Actions

Objective – Cleared: Objective = Objective03, Message = Excellent!

Entity – Set Visibility: Entity = Waypoint3, Value = 0

Objective – Start: Objective = Objective04

Entity – Set Visibility: Entity = Waypoint1&4, Value = 1

Entity – Convert Controller: Entity = Apollo, Controller Type = Player

Entity – Convert Controller: Entity = Bugsier 2, Controller Type = Static

If there is only one playership available and you convert it to static and another static ship to player the camera will switch automatically to the first playership available.

Objective04-Moor TriggerConditions

Ship – Moored with entity: Ship = Apollo, Entity = Ocean Prince

The Objective04-Moor Trigger has only a condition because this needs to be met first before checking any other conditions.

Objective04 TriggerConditions

Uber – Is Entity in Area with Speed for Duration: Entity A = Apollo, Area B = Waypoint1&4, Comperator C = Less Than, Speed D = 1, Seconds E = 10, Objective F = Objective04

Actions

Objective – Cleared: Objective = Objective04, Message = The pilot has transferred!

Mission – Clear/Win!: Message = You have been a great help today, captain!

The Uber condition gives automatic feedback by **InfoMessages** to be in the area, with the correct speed, and it counts down with a timer.

The mission is finished. Test it to check if it can be played to the end. To make the mission more interesting you can add some AI. Place some AI ships from the **AI** tab **Ships** list in the environment where you like. Also place the same amount of **PointEntities**, these will be the goals for the AI ships. Select the **AI Trigger** in the **Scenario Editor** that you created at the beginning of this tutorial and add for every AI ship the action **AIShip – Go to goal and respawn**. Select the **Ship** and the **PointEntity** for the **Goal** where it has to sail to.