Mission Editor Notes v1.3

Maykel Boes has written an excellent manual for the SSE Mission Editor. By studying it and experimenting with the features, you should gain a good grasp of the mechanics of mission creation.

The purpose of these notes is to provide clarification and amplification prompted by questions asked on the Ship Simulator Extremes> Missions board of the Forum.

The first section is in tutorial form with step-by-step exercises for doing many of the basic operations.

The layout of the next sections roughly follows the layout of the VSTEP manual. This will help you in cross-referencing.
Page numbers in Maykel's manual are referenced thus: {5 Environments}

Hyperlinked (Clickable) Table of Contents

Endnotes look like this: note[3]. To follow the note, click on the superscript number.
To return from the endnote, right-click and select **Previous View**, or press **Alt+Left arrow**.

Starting the Mission Editor

Click the button at the bottom of the SSE Main menu.

Caution: Each of these three operations can take some time to complete:
1. Loading the Mission Editor after clicking the button on the main menu: At least 30 seconds.
2. Loading the mission after selecting it in the editor: At least 30 seconds.
3. Loading the Environment after selecting it in the Environments panel: At least 45 seconds.

Proceeding before each one has completed may cause some buttons or selections not to respond.
Be patient.

The ME starts with a blank viewport (environment).
To begin a new mission, add an environment.
To open an existing mission, click the second button from the left on the main toolbar.

Table of Contents

Getting started with a new mission:

Click the leftmost button on the Toolbar (below File) to begin a new mission.

Initially you have an empty Environment filling most of the screen. On the right side of the screen are three panels: Entity creation, Entities in mission, and Environments.

Click the green **(+)** in the Environments panel to get a list of environments. Select one and then click OK. The selected environment is added to the Environments panel. To make an environment the current one, click it and then click the blue eyeball (third toolbar button in the Environments panel).

It may take a while for the environment to load completely.

Hold the right-hand mouse button down to rotate the environment to get a top-down view, zooming with the scroll wheel. *The mouse pointer must be in the environment,* ***not*** *in a panel.*
Pan around the environment with the **Arrow Keys**.

Sea, Weather, and Sky (Date/Time)

You can set the initial sea state and weather differently for each environment in the mission by clicking the third set of three buttons on the main toolbar. You can also make some changes during the mission by means of a trigger.
You can set the Date & Local Time that will be *the same in all environments*. The time shown above the chart—the mission time—is the **local time** in the current environment.

In SSE, the **local time** is the same in all environments—when it is Noon in NY, it is noon in San Francisco, in Marseille, and in Bora Bora. This differs from the situation in real life, where each of those environments is in a different time zone.
It is possible to advance the local time when you teleport to a different environment in order to simulate the travel time. In practice, you would adjust the advancement to arrive at a particular local time—to leave NY in the morning and arrive in Rotterdam at night—rather than to attempt a realistic travel time.

Right now, you should set the initial time to early in the day, because it might be hard to see what you are doing at night:

Click the ninth button from the left on the main toolbar (Sky/Time Settings).

The time zone, longitude, and latitude are preset for the current environment.
The Atlantic Ocean is also set to a specific 7 nm square of ocean. Even if you change the Lat/Lon sliders to put it somewhere else, it will go back to the default when your mission is run. This is **very** unfortunate.[1]

Set the time to mid-morning. Latitude determines how strongly the month influences the times of dawn & dusk, as you would expect.

Weather is set by clicking the eighth button from the left on the main toolbar (Weather Settings).
The sea state is set by clicking the seventh button from the left on the toolbar (Ocean Settings).
They are discussed later. There's no need to touch them now unless it's too gloomy or rough to work.

Contents

Placing a ship in the environment

Click the **Player** tab in the Entity creation panel. Click the **>** next to **Ships** to get a list of player ships.
Click to highlight the ship[2] that you want to be the Start Ship. It can take a few seconds to respond.


Gizmo

As you move the mouse pointer into the water environment, the ship will follow the pointer.
Left-click when it is near where you want it. A Gizmo will be over the ship.
To center the view on the ship, click to highlight it in the Entities in mission panel and then click the eyeball button.
The ship will be centered *only* after you move the mouse into the environment.

Use the Gizmo to position the ship: Drag anywhere on the red arrows to move the ship around the environment. If you try to drag the gizmo where the ship can't go, the ship will stay behind in a red box.
To rotate the ship, click on the gizmo's green circle. The green pointer will rotate to match the mouse. You can then drag the mouse to point the ship in the right direction.
As in SS 2008, you can drag the mouse away from the gizmo to line up with some object.

Now you should set the **properties** for that ship: Highlight it in the Entities in mission panel and then click the **P** button (left of the eyeball on the panel toolbar).

In the Properties for Entity panel, change the **Name** to be meaningful. Be sure that **Visible** is checked. You can then click the red X to hide the panel, or double-click its title bar to collapse it.

Now that you have a player ship, you can set the **mission properties**:

Click the **m** button (fourth from the left on the main toolbar) to get the MissionProperties panel.
Enter the mission title and your forum name. Select a **Difficulty** and a **Start Ship** from the drop-down lists. *You can't save the properties if you have not picked a Start Ship* .

Leave Damage and Locked unchecked. Select the language that you will work in,[3] and any tags.[4]
You can enter **briefing notes** that will be displayed when you select the mission on the Single mission > Select Mission menu. You must put at least one character in the box in order to save the properties.

Saving your mission

Finally, you should save the mission by clicking **File > Save as...**
In the Save mission as... panel, click the **[User Missions]** button.
Enter the name of your mission in **File name:** and click the Save button.
Caution: Be sure that the mission name does not contain any characters that are not allowed in file names.
Warning: Do **not** click the File > Upload button unless you have a complete and tested mission that you intend to submit for testing and possible addition to those missions available for in-game download.
It is not necessary to upload a mission in order for you to play it.

Now you are ready to proceed with completing your mission. You should save often as you proceed, in case you need to undo some mistake.
After the initial save, you can click the **Save** button (third from the left) on the main toolbar.

The mission, as it stands, is playable as a free roam. You should find it in SSE **Play > Single mission**. You could customize it with mooring bollards, AI, even additional player ships. Contents

Loading your mission into the Editor

1. Click **File>Open...** or Click the second button from the left on the main toolbar.
   Click the **[User Missions]** button to access the missions you have saved in:
   [yourDocuments]>ShipSimExtremes UserData>Missions.
2. Click to highlight the mission and then click the **Open** button.
3. Select your language from the drop-down list and then click the OK button.
4. After a few seconds, the list of environments will be displayed in the Environments panel[5].
5. Select an environment—usually the starting environment—and then click the Eyeball button on the panel toolbar to make it the current environment. Wait for it to be displayed completely.

Deleting entities from the mission

Selecting entities in the mission

You can select larger objects by clicking them in the environment, but it is often easier and safer to select them in the Entities in mission panel. You need to **deselect** any currently selected entity before trying to select from the Entities in mission panel.[6] (V1.3)

You can then remove them by either clicking the red (-) button or pressing the Delete key.

The list is arranged by environment and then by type—Player, Static, and AI.

Sections of the list can be expanded or condensed by clicking the **>** symbol.

A **>** in front of a ship means that it carries one or more deployable boats. Click the **>** to see them.

Caution: Two or more entities may occupy the same space in the environment—a **SphereAreaEntity** on a ship, for instance. If you try to select one in the environment, you might get the wrong one. It is safer to select the entity from the list.

When an entity is selected, you can view and edit its properties by clicking the **P** button on the panel toolbar. You can center the entity in the environment by clicking the blue eyeball button on the panel toolbar. The centering happens when you move the mouse into the environment.

When setting certain parameters for a condition or action in the Logic Editor, you will use the **Select from scene...** button. You can select from the list rather than clicking the entity in the environment.

At the bottom of the panel, you can select what entities are listed: Normally you would select the second radio button and both check boxes to be sure that you can see every entity in the mission.

One common mistake in SS08 was failure to give Area objects unique names; trusting to the Editor to figure out which one was meant.
In SSE, each entity within a type is given a unique ID number in brackets. Look at it if in doubt as to which entity you want to select.
Better still: Give each entity a meaningful name when you place it.

Contents

Mooring a ship in the Editor

First move the ship into position against the quay. Be sure that the bounding box is still green.
A red bounding box indicates that you are too close, or that the depth is too shallow.

Next, light up the mooring points on the ship to guide you in placing the bollards:

Select the ship by clicking on it or by selecting it in the Entities in mission panel. Then click on the **Moor** button — the 15th button on the main toolbar (bollard). This will light the yellow mooring points on the ship. You can now hide the gizmo by clicking somewhere away from the ship.

Placing mooring bollards & other objects on shore

Now to place the bollards: Click the **Static** tab on the Entity creation panel. Then click the **>** next to **EnvironmentObjects** to get a list of bollards. Click a type to select it and move the mouse pointer over the quay to near where you want the bollard.
Left-click to get the gizmo. If you don't see the gizmo, push the M key.
Drag the gizmo, if necessary, and then click it to add the bollard to the Entities in mission list.

Notice that the bollard is not visible on the quay. Objects are initially placed with a height (Y) of zero—at sea level. Quays are 3 meters above the water. You need to raise the bollard to the correct height:

With the bollard selected in the Entities in mission panel, click the eyeball on the panel toolbar and move the mouse into the environment to center on the gizmo.

Click the **P** button on the panel toolbar to get the Properties for entity panel.

Click the **+** next to **LocationReal** to edit the values. Backspace over the **Y** value and set it to 3. This should put the bollard on the Quay. You may need to make fine adjustments with the little U/D arrows or enter a different value in some locations. **Do not move the gizmo during the adjustments.**

After the height is correct, click near to, but not on the gizmo to make it go away. Then you can re-select the bollard in the Entities in mission panel. This routine is necessary to get the gizmo properly attached to the bollard.

Now you can use the gizmo to move the bollard into position.

While the Properties for entity panel is open, you can give the bollard a meaningful name.

To tie the ship up to your bollards, you need to select it and then click the **Moor** button on the main toolbar to light the yellow points.

Click one of those points to light up the shore points. Click a shore point to connect the line.

Notice that the ship point is now red to indicate that it has a line. You can click a red point in the editor or in the game to drop the line.

Contents

Placing Waypoints and Triggers

The terms *waypoint* and *trigger* are used almost interchangeably in SSE.

In SS08 there was a clear distinction between the two: Active waypoints were visible on the chart and as ugly green circles on the environment. Triggers were not visible on the chart or in the environment. Visible waypoint icons on the stack had notes to tell the player what to do on the way to, and at the waypoint; in SSE that function is performed by *Objectives*.

In SSE, with few exceptions, triggers are marked on the chart and the CTW & DTW [7] are shown below the conning circle. They are also waypoints because one is expected to navigate toward them.

Whatever you call them, you need something to hang them on. You can use ships and other visible objects, or you can choose from a set of 3 invisible entities: **SphericalAreaEntity**, **BoxAreaEntity**, and **PointEntity**.

The one you will probably use most is the sphere. You can set its radius and detect when something is inside it. The box is similar, except that you can control all three dimensions. The point has no dimensions.
You place them as you would a ship: Click the **Static** tab and double-click the **>** next to **ScenarioEntities**. You display their properties with the **P** button and give them a name and appropriate dimensions.
You must also set the visibility — whether it can be seen on the chart. Usually they start out invisible and are made visible in the mission logic (the Scenario) when it's their turn.

Example waypoint:

Assuming you have done the Getting Started exercise and have a mission with one player ship, we will now tell it where to go:
Pick a spot on the water in the environment some distance from the player ship, and place a Spherical Area there.
Give it a name like WPT1 and set the radius to 100.
Set **visibility** false by un-checking the box.

Now you have a waypoint in the mission, but you have some work to do before you can see it in SSE:

Contents Continue

The Logic Editor (Scenario editor)

There are two parts to mission creation, and two editors: The one that you have been using to place objects in the environment, and a **logic** editor (also called the scenario editor), where you build the finite state machine that controls the running of the mission. {6 Scenario Editor}
Open the Logic Editor by clicking **Options > Scenario Editor** on the top menu bar[8].
The first thing we need is an **Objective** to tell the player what to do. Objectives are one type of variable that you can create in the lower right panel of the logic editor, titled Variables.
Click the green + button to get a panel in the upper left with fields for **Variable Name** and **Variable Type**. Click the down arrow to get a list of variable types. Click to select **Objective**. Give it a name like OBJ1. In **Short Description**, type a one-line note like Sail to WPT1. This will be shown as soon as the objective becomes active. The **Long Description** is shown if the player clicks the white arrow under the short description. You must type something in both fields before the OK button works.

Now you can start to build the logic diagram using the toolbar on the center right Nodes panel: Click the red triangle on the left end and drag it to the top center of the blue panel. Double-click to anchor it. The large black circle means it is selected. Its name appears in the Node: field of the top right Node Properties panel. You can backspace and type a better name. This is where your mission will start.

Next you need a **trigger** to initialize and start the mission with the first objective:
Click the blue trigger tool and drag the trigger below the red triangle. Double-click to anchor and select it. You can name it Initialize in the Node Properties panel. You can/should type comments in the **Description** field to remind yourself why you did what you did.

You need to connect the triangle to this trigger: Move the mouse into the small circle on the triangle until the circle turns red. Then drag down with the left button into the small circle on the trigger to connect the two with a black arrow.

One Time below the trigger means that the trigger will only act once rather than continually checking conditions.

If necessary, click on the outer circle of the trigger to select it. When the trigger is selected, there will be a third, black, circle. The properties for the selected trigger are in the top left panel.

That toolbar has green buttons to add **conditions (c)** and **actions (a)** to the trigger.

The trigger checks the conditions until they are **all** met, and then does the actions in order.

For this trigger there are no conditions and two actions: Start the first objective and make WPT1 visible on the chart.

Click the green action button.

For **Property Type**, click the down arrow to the right of the field to get a list of actions. Scroll down and double-click to select Objective-Start.

Click the **Objective** button and select OBJ1 from the drop-down list. Click OK and OK again.

Click the green action button.

For Property Type, click the down arrow to the right of the field to get a list of actions. Scroll down and double-click to select Entity – Set Visibility.

Click the **Entity** button and then click the **Select from scene** button. The logic panels get out of your way so that you can click on the WPT1 sphere in the environment or in the Entities in Mission list. Click OK. Click the **Value** button and enter **1** in the top field to make WPT1 visible on the chart. Click OK.

If you were to save and run the mission at this point, it would start with the short objective displayed and WPT1 visible on the chart if you are close enough. The CTW & DTW are shown below the conning circle and the open circle waypoint marker on the conning circle is visible if you have the center button active.

Note: If you run out of space for triggers, etc., see Enlarging/Scrolling the Logic Editor in the Tips section.

Clearing the waypoint upon arrival

The next step is to detect when the player ship reaches the trigger radius of WPT1, and then make WPT1 invisible and clear objective 1.
To do that, you need to add another trigger, but there must be a **State** between the triggers.
Click the white **State** tool on the Nodes panel and anchor it under the trigger.
You could change its name to "Enroute to WPT1" because that is the state that the mission is in.
Connect the trigger down to the state as you did before.

Place a trigger below the state and connect the state to it. Select the trigger and give it a name in Node Properties.
The new trigger will have one condition, the arrival of the player ship: Select the trigger and click the green Condition button in Trigger Properties.

Select Area – Contains Player Ship from the **Property Type** list.
Click the Area Entity button and the **Select from scene...** button.

Then click on the WPT1 sphere. Click OK.

You need to add two actions: Make WPT1 invisible and clear the objective OBJ1.

You make WPT1 invisible the same way you made it visible earlier except that you set the variable to 0. You clear OBJ1 by selecting Objective – Clear and then clicking **Objective** and selecting OBJ1 from the list.

Click the **Message** button and type whatever is appropriate.

Now you are done with the first objective. You can start the second objective in this trigger as you started the first objective in the previous trigger, using an objective variable that you create for OBJ2 as you did for OBJ1:

Contents

Rescue a swimmer

Following the tradition of SS 2008, for our next objective we will rescue a drowning person. This involves much more work than it did in '08:

1. First we need a person in the environment, it does not matter where. Frank has volunteered.
2. Then we fit Frank with a radius of 30 meters, so we can tell when we are close enough to him.
3. Then we toss Frank overboard at a point area that we have placed.

We will do 1-3 in that first initializing trigger:

In the Entity creation panel, click the AI tab and then double-click the > next to **Crew**.

Select Civilian 03 and double-click on the water near the player ship. Center the top-down view on him. In his properties panel, name him Frank.

In the Entity creation panel, click the Static tab and then double-click the > next to **ScenarioEntities**. Select **SphereAreaEntity** and center it as accurately as possible over Frank.

If necessary click the **P** button to get the properties panel for the sphere. Set the radius to 30 meters. Give it a name like **FrankSphere** and uncheck the **Visible** box.

Now you need to set a **PointEntity** somewhere past the last waypoint as the place to toss Frank overboard. Name it FOB.
In the Entity creation panel, click the Static tab and then double-click the > next to **ScenarioEntities**.
Select **PointEntity** and place it where you want to toss Frank.
Open the Scenario Editor and select the initializing trigger. Add these two **Actions**:
Entity – Attach to Entity **ChildEntity** is the 30m sphere area, **ParentEntity** is Frank.
Character – Overboard **Character** is Frank, **Ship** is the Player ship, **PointEntity** is FOB.

Now you need another objective variable to rescue the swimmer:

In the Variables panel, click the green + button to get a panel in the upper left with fields for **Variable Name** and **Variable Type**.

Click the down arrow to get a list of variable types. Click to select **Objective**. Give it a name like OBJ2. In **Short Description**, type a one-line note like Rescue the Swimmer.

The **Long Description** should say: Approach within 30 meters of the swimmer and keep the speed under 2 knots for 20 seconds.

Continue in the last trigger that ended Objective 1 by adding an action to start the next objective, OBJ2:

Click the green action button.

For Property Type, click the down arrow to the right of the field to get a list of actions. Scroll down and double-click to select Objective-Start.

Click the **Objective** button and select OBJ2 from the drop-down list. Click OK and OK again.
Click the green action button.

For Property Type, click the down arrow to the right of the field to get a list of actions. Scroll down and double-click to select Entity – Set Visibility.

Click the **Entity** button and then click the **Select from scene** button. The logic panels get out of your way so that you can click on the **FrankSphere** sphere. Click the OK button.

Click the **Value** button and enter **1** in the top field to make **FrankSphere** visible on the chart. Click OK.

We have now started the second objective, to rescue the swimmer, and have made that waypoint visible on the chart. We are now in the "rescue swimmer" state, so we need to add a **State** below the trigger and connect the trigger down to the state as we did before.

Then we need to add a trigger below the state to determine when we have completed the rescue:
Connect the State down to the new trigger.
Click the green condition button and select Uber – Is Entity in Area with Speed and Duration.
Entity A is the player ship, **Area B** is **FrankSphere, Comparator C** is "less than or equal", **Speed D** is 2, **Seconds E** is 20, and **Objective F** is OBJ2.

The displayed condition should make sense when you are done.
You may recognize this Uber condition from the rescue, taxi, etc. waypoints in SS 2008.
This condition adds advisory messages to the objective note to stay within radius, maintain speed, etc. It also displays a countdown timer.

The first action should be to take the swimmer aboard:

Click the green Action button and select Character – Transfer To Ship.
Character is Frank, **Ship** is the Player ship.

You need to add two actions: Make **FrankSphere** invisible and clear the objective OBJ2.

You make **FrankSphere** invisible the same way you made it visible earlier except that you set the variable to 0.
You clear OBJ2 by selecting Objective – Clear and then clicking **Objective** and selecting OBJ2 from the list. Click the **Message** button and type whatever is appropriate.

Save and test the mission. Notice that Frank swims toward the boat. Drowning people are more helpful in SSE. The **Ship** parameter in Character – Overboard tells the swimmer to follow that ship if he is close enough.

Notice, also, that the swimmer now appears in the boat. AI characters wander about the ship, and even climb ladders.

Ending the mission

You should end the mission logic with a message that the mission has either Won, or Failed:
Mission – Clear/Win!
Mission – Lose/Fail!
Message — Type something appropriate.

Contents

Editing Triggers

To examine or edit a trigger in the Logic Editor, first click on the edge of that trigger to select it. The Trigger Properties panel in the upper left will show the conditions and actions for that trigger. To examine or edit one of those items, double-click to select it.

Then click the third toolbar button (Edit Action/Condition) to get the Configure Condition or Configure Action panel. You can then change any of the parameters.

Click the **Cancel** button to leave the panel without changing anything. Contents

Taxi waypoints

The two character actions do not help much for taxiing:
You can use Character – Transfer To Ship to move people into the boat from shore, and to transfer them to or from a ship. But you can't put them ashore with Character – Overboard because they end up submerged.
The big drawback is that you must use AI characters, which are expensive in memory use, impact the FPS, and won't sit down in the boat.

One way to move people around is to place three copies of the same person: One at the departure point, one at the destination, and optionally one sitting in the boat. You simply change the visibility of the copies to move them to or from the boat.

Use Static characters. There are three versions of each: Standing, leaning, and sitting. With patience you can place a seated one on the Vstp7. If you do place a character on the boat, remember to attach it to the boat with the Entity – Attach to Entity action.

Remember: If you are building a taxi waypoint on a ship, you **must** attach the Area Entity to the ship by using the Entity – Attach to Entity action in an initializing trigger.

Contents

Mooring waypoints

There are no "mooring waypoints" like those in SS08. Instead there are four **conditions:**

Ship - Moored The ship must be tied up to at least one bollard.

Ship - Moored2 The ship must be tied up to at least two bollards.

Ship - Moored with entity The ship must be tied up to a specific thing, usually another ship or a bollard.

Ship - Not Moored The ship must not be tied up to anything.

If you use either of the first two, the ship need not be tied to the bollard that the waypoint was built on, as was the case in SS08. This avoids a common problem in SS08 missions where the player chose the wrong bollard.

So, for a do-it-yourself mooring waypoint, you need to make some bollards, or pick a ship.
Then you need to place a **SphereAreaEntity** to provide a visible waypoint on the chart and to check that the ship is in position to tie up. The sphere doesn't really need to be visible—the player could be told "moor at pier 14". The sphere is still necessary to check that he doesn't tie up someplace else if you are using the first two conditions.
Remember: If you are building a Mooring waypoint on a ship, you **must** attach the **SphereAreaEntity** to the ship by using the Entity – Attach to Entity action in an initializing trigger.

There are also two actions that you can use to moor or unmoor a ship without troubling the crew:

Ship - Connect To Moorage Point Connect closest ship point to a specified point. Used when the ship is not under the player's control, as a static ship being towed.

Ship – Disconnect All Mooring Lines and tow lines. Used when a ship becomes AI.

Anchoring

To anchor a ship in the editor, select it and then click the 17th button on the main toolbar—the anchor. This will light the blue anchor points on the ship. To drop the anchor, click a blue blob. To raise an anchor, click a yellow blob.

You can control anchoring in the mission logic with these conditions and actions:
Ship – Anchored: Checks if the specified ship is anchored
Ship – Not Anchored: Checks if the specified ship is not anchored.
Ship – Drop Anchor: Drops all anchors
Ship – Hoist Anchor: Hoists all anchors
You would use the last two actions when towing a static ship from one anchorage to another.

Yes, you can anchor AI ships, but be careful to weigh anchor before you give then a command to move. Remember that AI ships don't stop on a dime, you need to wait a while before you drop anchor after a command to stop. {12 Anchoring}

Contents

Towing & Disconnect waypoints

These, also, are do-it-yourself projects in SSE. To cause the player to connect a towline to a ship, you create an Objective variable to tell him what to do, and optionally, a **SphereAreaEntity** to guide him into position with a visible waypoint and to check that he is in position. Or, you could just tell him to tow the ship and leave it up to him to position his ship. The conditions that you can use in the completion trigger are:

Ship – Towing ...any ship; so you should use the sphere to be sure he has the correct ship.
Ship – Towing Ship Checks that a specific ship is being towed.
Remember: If you are building a Towing waypoint on a ship, you **must** attach the **SphereAreaEntity** to the ship by using the Entity – Attach to Entity action in an initializing trigger.

For the equivalent of the SS08 Disconnect waypoint, you can use one of the two conditions above to check that he still has the tow, or you can use the condition Ship – Not Towing to fail the mission at that point.

If he is in the disconnect area and still has the tow, issue a message to tell him to disconnect and use Ship – Not Towing to tell when he has disconnected.

There is more work that you need to do when towing:

If you are towing a player ship that, according to the mission story, is disabled, you can disable steering, engines, etc. so that the player has limited ability to assist the tow.

Contents

Disabling Ship Controls

You can use the action Ship – BridgeComponents – SetEnabled to disable(0) or enable(1) any bridge control. {12 Towing & Bridge Controls}
There are three properties for this action; **Entity, Componentindex,** and **Enabled**.
Entity is the ship, selected from the scene.
Enabled is either 1 (true), or 0 (false).
Componentindex is the index, or ID, number of the controller that you want to disable/enable.
To get that number, you need to look into the bridge or cockpit of the vessel, where little magenta numbers are shown on the controls if you press **Shift+Ctrl+A**.
Getting a good view of the controls inside the bridge is the tricky part: You have to do it in the Editor without a bridge camera. For large ships, like Agile Solution, approaching from astern with the Up-arrow key works best. The zoom with the mouse wheel is too course, but combined with the arrow keys you can position yourself in the wheelhouse and see the controls.
The ship must be selected in order to see the numbers, and you must also have enabled them by pressing **Shift+Ctrl+A**.
Fortunately the numbers show through the roof and walls of the wheelhouse, so you can tell when you have them turned on. Also, the numbers are always the same size (tiny) regardless of the zoom level.
The numbers for the main controls are not the same for all ships—for some the wheel & throttle are 1 & 2, but for Agile Solution the wheel is 24, the throttle 22, and the bow thruster 23.

This table may help, but you should verify by testing.

Contents

Barges —Herkules Atlas

To connect Atlas to a barge in the editor, select her and then click the **Tow** button —16th from the left on the main toolbar.

Be sure to connect to the stern of the barge to push. The bow of the barge has the tall steering cross. You should center Atlas carefully and get her as close to the barge as possible before connecting.

Lash-ups

Unfortunately, there are not enough mooring points on the barges to allow a good lash-up and also a pushboat connection that gives good steering control for an abreast lash-up.

Nor are there enough mooring points for towing on the hip.

There is also the question of whether the barge dynamics might fight each other. In SS 2008, pre-built lash-ups that had their own dynamics were available.

Contents

Teleporting — Changing location or Environment

You can rapidly move the player ship to a different location in the current environment, or to a specific location in a different environment.

You can also teleport to the current or new environment and advance the time in that environment by a number of hours. For example: you can moor in New York at 3 PM and take a nap until 9 PM.

The three teleporting commands are:

Entity – Teleport: Move somewhere else in the current environment. **Entity** is the player ship. **DestinationEntity** is a PointEntity that you have placed in the current environment.

Entity – Move to Environment: Move to a specified location in a different environment.

Entity is the Player ship. **Environment** is the name of the new environment, typed very carefully.
Point is a PointEntity that you have placed in the new environment.
Note: This does not appear to operate in V1.3.
You can use Environment – Switch to different Environment instead. Set **TravelTime** to zero.

Environment – Switch to different Environment: Move to a specified location in the current or new environment and advance the time in that environment by a number of hours.
Environment is the name of the new or current environment, typed very carefully.
TravelTime is the number of hours that you want the mission time to advance. (an integer)
Ship is the Player ship. **Point** is a PointEntity that you have placed in the new environment.

For example: To leave NY at 9:00 AM and arrive in Marseilles after dark at 9:00 PM, you would use Environment – Switch to different Environment with **TravelTime** set to 12 hours. You would arrive on the same date that you left NY. If you want to simulate taking a week for the transit, you can set **TravelTime** to (7 * 24)+12 = 180 hours. The date is not shown in the game, so you might not bother. However you could tell from the change in the moon's phase and the delay of ~7 hours in its rise and set times that seven days had passed.

Contents

Timers

You can use timers to produce a delay in doing a certain action, or to test if a certain amount of time has passed.
You create timers in the Variables panel:
Click the green **(+)** button on the panel toolbar and select **Timer** from the **Variable Type** drop-down list.
Give it a name like Timer_1.
Set the **Target Time** to the number of seconds you want the timer to run before it expires. New Timers always start at 0, but you can set a different initial value with an action: Timer– Set Current Value.

Conditions

You have three conditions to use in testing the status of the timer:
Timer– Expiration Check: Checks if the specified timer is expired.
Timer A — select from list.
Timer – Time Passed Check: Check if the specified timer has passed the specified time.
Timer A — select from list. **Comparator X** — select from list (usually "Equal or Greater Than").
Float B — specified time in seconds.
Timer – Time Remaining Check: Checks if the specified timer has the specified time left.
Timer A — select from list. **Comparator X** — select from list (usually "Equal or Greater Than").
Float B — specified remaining time in seconds.

Actions

There are five actions for controlling the timer:
Timer–Start: Starts the specified timer.
Timer – Stop: Stops the specified timer; the current value becomes 0.
Timer – Pause: Pauses the specified timer. You can continue with Timer – Start.
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.

Two actions allow you to display the timer to the right of the Objective notes:
Timer – Start Displaying Timer: **Timer A** is the timer selected from the Select Variable drop-down list.
Objective B is the objective selected from the Select Variable drop-down list.
Objective B must be active—displayed—in order for the timer to be displayed.
Timer – Stop Displaying Timer: **Timer A** is the timer selected from the Select Variable drop-down list.

Elapsed Time

If the Target Time is not set to zero, the displayed time starts at **(Target Time**) and counts down to zero. Then the displayed time counts up. If you want the time to be displayed starting at zero, set **Target Time** to zero.

Contents

Fires & FiFi

Fires are **static** entities of type **Effects**. They are placed as you place bollards, adjusting the **Y** value to the height of the surface. You can also adjust the **X & Z Scale** values to spread the fire out.

The fire entity **must be attached to a parent entity**, such as a ship.

You light the fire in the logic editor with the action: Fire – Start.

If you want just smoke without flames, use the action: Fire – Smoke Only.
Set **SmokeOnly** to 1 to get only smoke. Set it to 0 to get the flames.
This action can come either before or after Fire – Start.

Example:

Place a static *Latitude* and a player *Sherpa* in the environment.

We'll start a fire on the helipad:

1. In the Entity creation panel, click the Static tab and the **>** next to **Effects**. Select **Fire**.
2. Move the mouse pointer over the helipad and left-click to get the gizmo. If necessary, press the **M** key
3. Center the view on the gizmo and zoom in. The white cube that represents the fire was placed at Y=0, and needs to be raised to sit on the deck.
4. With the fire selected in Entities in mission, click the **P** on the toolbar to get the **Properties** panel.
5. Click the **+** next to **LocationReal** and adjust Y to rest the cube on deck. (about 9.4 meters)
6. Click somewhere else to de-select the fire, and then select it again in the Entities in mission list to adjust the gizmo.

In the Properties panel, you can click the **+** next to **Scale** and increase X & Z to spread the fire out.
You will need to run the mission in order to see the effect.

To light the fire, you need to put the action Fire – Start in a trigger in the logic graph.
When you run the mission, you can douse the fire with *Sherpa*'s monitors.

You have two conditions that you can use in the logic to check the condition of the fire:
Fire – Alive: Checks if the specified fire is still burning.
Fire – Extinguished: Checks if the specified fire is extinguished.

There is an action to stop the fire: Fire – Stop. This does not *extinguish* the fire. You can only do that by putting a lot of water on it. The condition Fire – Extinguished will **not** be met.

Contents

Water Cannon

You can control the water cannon in the mission logic, and can use an AI ship for FiFi:
Ship – Watercannon – Enable: Enables or disables the water cannon.
Ship – Watercannon – Set Target: Sets the target of the water cannon.
Ship – Watercannon – Set Power: Sets the power of the water cannon.
All three require **Entity** = the ship or deployable, and **Watercanon Index** = 0 if there is only one. If there are more on the vessel, you need to guess & test to find the index number.
Enabled = 1 to enable, or 0 to disable—Start/stop squirting water.
Power is a number that's usually between 40 – 50. You can experiment to find what works best.
Target Entity is either a PointEntity or a ship.

The water cannon can affect other entities, such as a ship, if it puts enough water on it.
To determine if enough water has hit an entity, use this condition:
Entity – Affected by Water Cannon **Entity** Is usually a ship, selected in scene.

Just what "Affected by" means is up to you. One not-particularly-elegant possibility:

Entity – Destroy

Entity – Destroy **Entitiy** is one that you want to remove. In the case of a ship, it just disappears; no dramatic sinking. If it is a player ship, it leaves behind its icon on the left of the screen, but clicking it has no effect.
Whether it also destroys your mission is something you need to test.

In-game, an indication that the water is hitting the target is the spray of large bubbles from the target.

Contents

AI Ships

The AI in SSE does not require creating paths, as it did in SS08.
AI ships find their own way to a **PointEntity** goal, avoiding any obstacles as best they can.

To make the equivalent of the SS08 AI:

1. In the Entity creation panel, click the AI tab and double-click the **>** next to Ships.
   Select the AI ship from the list and place it at the starting position.
   Rotate it to face in the direction that it should start moving.
   Select it in the Entities in mission panel and click the **P** tool for the Properties for Entity panel. If the ship should start at the beginning of the mission, check the **Visible** box.
   If you will start that ship later in the mission, uncheck the **Visible** box.
2. In the Entity creation panel, click the Static tab and double-click the **>** next to ScenarioEntities.
   Select **PointEntity** from the list and place it at the end of the AI ship's route.
   If necessary, click the **P** to display the Properties for Entity panel for the point.
   Give it a name like "EndAI1", and uncheck Visible.
3. You should start a separate branch of the logic diagram for starting AI. This will avoid clutter in the main initializing trigger: {7 AND & OR Setups}
   Place a new State to the left of the State that is just below the main initializing trigger.
   Name the new state AI, and connect the main initializing trigger to it.
   Place a trigger below the AI state and connect them. Name the trigger "Initial AI".
   All AI that starts when the mission starts should be started in this trigger.
   To start an AI ship later, place a State and a Trigger with a condition below this trigger.
4. To start an AI ship at the start of the mission, select the **Initial AI** trigger and the action AIShip – Go to goal and respawn. Set the **Ship** and the **Goal**. This will behave like an AI ship in SS08, but is less likely to bonk into something.
5. To start an AI ship later when something happens in the mission, place a State/Trigger pair below the **Initial AI** trigger.
   Set an appropriate condition.
   The first action should be Entity – Set Visibilty to make the AI ship visible.
   The second action is AIShip – Go to goal and respawn as in 4. above.

Exercise: In the NY environment, place a player Apollo in the East River north of the Williamsburg Bridge, heading south. Make Apollo the Start Ship.
Place an AI Loire just south of Apollo, heading south.
Place the goal for Loire in the North River (Hudson) above Hoboken.
You can follow Loire as she finds her way around the tip of Manhattan.

Contents

Controlling AI ships

Unlike SS08, where you could not do much to modify the behavior of an AI ship once it was spawned, in Extremes you have many ways to modify the behavior of an AI ship.

Speed

You can change the speed of an AI ship at any point on its route with these commands:

Note: All speed commands are in meters/sec. There are 1852 meters in a nautical mile.

One knot equals 1852/3600 or 0.514 meters per second.

AIShip – Set preferred speed: Sets a speed in m/sec that the ship can modify if necessary to avoid collision. You should use this if the AI will encounter traffic.

AIShip – Set Accelerate from Start: Accelerate up to the preferred speed if **DoAccelerate** is 1. If **DoAccelerate** is 0, start with the preferred speed.

AIShip – Apply speed: Commands the ship to travel at a specified speed in meters/second.

The ship will not slow for traffic. Use this if you need the ship to hold a specific speed for a transfer objective.

AIShip – Apply full speed: The ship will run at the maximum speed for her type. She may not slow for traffic.

AIShip – Stop: Commands specified AI ship to stop sailing. She decelerates to a quick stop.

AIShip – Set Decelerate to Final Goal: The ship will decelerate before reaching its goal if **DoAccelerate** is 1. Normally the AI ship will not decelerate to its goal.

Setting or modifying the goal

You can redirect the AI ship to a different goal while it is on its way to the original goal:

AIShip – Go to goal: The ship will go to a specified goal and stop sailing when the goal is reached.

AIShip – Follow object: The AI ship will follow another ship—player or AI—keeping a safe distance from that ship.

AIShip – Approach object: The AI ship will get close to the object—usually another Player or AI ship, and then move away. It will repeatedly approach the object until it is told to go elsewhere. In most cases, a trigger is set to sense when it is within a certain distance from the object. Some missions use this to simulate a pilot transfer, etc.

AIShip – Abandon Object and go to Goal: This can be used to secure from following or approaching and send the ship to another goal.

AIShip – Ignore object: The ship will not try to avoid the object. You might use this if you need to get close to her with the player ship without deflecting her from her heading.

Contents

Convert Controller—to Player, AI, or Static

Any ship can have its controller type changed to **Player, AI** or **Static.**
Of course, unless the ship exists in the program as a player ship you can't take control of it by changing its controller type to Player.
An example is approaching a waypoint in Sydney and suddenly finding yourself aboard the pilot boat, having left your ship running ahead full toward land. When you catch up to her, you find her sailing blissfully along. You board her and take command.
While you were gone, her controller type was changed to **AI** with a safe goal. When you board her from the pilot boat, her controller type is changed back to **Player**, and the Pilot boat becomes **AI** and goes home.

Entity – Convert Controller makes it happen. **Entity** is the ship; **Controller Type** is selected from the top list: Player, AI, or Static.

Example:

1. Start a mission with Apollo as the Start Ship and a second player ship i.e. Arie Visser behind her and heading in the same direction.
2. Place a visible SphereArea some distance ahead of Apollo. Name it ConvertTrigger.
3. Place a PointEntity a longer distance from the sphere at about 45 degrees from the first heading. Name it AIGoal.
4. Add a trigger to the logic diagram with:
   Condition: Area-Contains Player Ship **Area Entity** is the ConvertTrigger sphere.
   Action: Entity-Convert Controller **Entity** is Apollo; **Controller Type** is AI.
   Action: AIShip – Go to goal and respawn **AIShip** is Apollo; **GOAL** is AIGoal.

Save and run. Drive Apollo to the sphere. Apollo should become an AI ship and head off toward the Point Entity. What happens to me? My player ship is gone!
You end up in command of the second Player ship **if** there are just two in the mission.

But what if there is no other player ship? You become a passenger on the AI boat. You can go to the helm view because Apollo was a Player ship, but the controls don't work.

If there are three or more Player ships, you stay on the AI ship until you choose a player ship from the icons on the left side of the screen.

If you have ever wondered what it would be like to ride an AI ship, this is how you do it.

Checking Controller Type

You can test if a ship is of a specific type —Player, Static, or AI—with this condition:
Entity – Controller Comparison **Entity** is a ship; **Comparator** is either Equal To, or Unequal To;
Controller Type is selected from the drop-down list.

Contents

Deployable craft

The deployable craft can be controlled in the mission logic by these conditions and actions:
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 deployable craft on a ship are identified by an ID number. If there is only one craft on the ship, its ID is 0. For multiple craft, you can guess and check.

There is a **>** next to ships with deployables in the Entities in mission panel. Click the **>** to list the deployables. Usually the [ID] numbers in the list are in the same order as the craft numbers:
To select craft[0] click the first craft in the list.
To select craft[1] click the second craft in the list.

One other chore you must attend to is changing the control type when you deploy or recover the craft:
Aboard the ship, the craft are Static. When a player deploys a craft, it is automatically a Player craft. However, when deployed by the mission logic it can also be an AI craft.

Contents

Weather

Weather in SSE is about clouds & wind. Precipitation—rain, snow, hail—comes from clouds. To get the precipitation you want, you must have the right clouds.

The weather—clouds—moves across the earth because the air mass moves from regions of high pressure to regions of low pressure. That pressure differential is represented in SSE by wind— its velocity and direction. By controlling those parameters of wind, you can have traveling weather systems in SSE. {19 WeatherSim Settings}

Initial weather

The starting weather in each environment in the mission is set independently with the eighth button from the left on the main toolbar. WEATHER SETTINGS Click the **WeatherSim** tab.

There are sliders for temperature and precipitation amount. The temperature determines the kind of precipitation—rain, snow, or hail.

The sliders for wind speed and direction control the movement of the weather. The speed is in knots. The wind blows **from** the indicated direction in V1.3. Note that this is the common convention that the "North Wind" blows out of the north.

Clouds are in three altitude regions: Low region clouds can be Nimbostratus, Cumulonimbus, or a mixture of the two. These types can produce heavy precipitation—progressing from rain or snow to hail and thunderstorms as you move the slider right toward Cumulonimbus (thunderheads).

Middle altitude clouds—Altostratus and Altocumulus—produce lighter precipitation, and are not as dark as the lower clouds. As you move the slider right, the precipitation goes from large clouds with light snow or rain, to small clumps that produce drizzle.

High altitude clouds—Cirrus and Cirrocumulus—are very light and wispy "mare's tails" composed of ice crystals. They produce no precipitation.

Each layer has a slider to set the percentage of cloud cover.

These settings determine the clouds that are generated on the edge of the environment, not directly overhead. The clouds are moved over the environment by the wind.

You can test the settings by clicking the **Preview** button. The weather will develop and move at 10X normal speed. Press the button—now **Stop**—to stop and reset the generation.

The **Load** & **Save** buttons are for making presets. You don't need to use them. Your final settings will be saved in the environment when you click the **Save** button on the main toolbar.

These settings control only the generation of weather that will move in. To set the weather that is overhead when the environment initializes, you must use the **Cloud Preset** tab.

Contents

Cloud Preset

Use the **Cloud Preset** tab to set the clouds that will be overhead when the environment initializes. Only the position of the various cloud types is set here. Their subsequent movement and the precipitation they produce is governed by the same wind and temperature that you have set in the **WeatherSim** tab.

To add initial overhead clouds, first select the altitude region from the top list. Then select one of the two cloud types in the second list.

Both types in a region can be seen together on the plot, distinguished by color—yellow where they overlap.

Only the clouds in a single altitude region can be displayed on the plot at one time—selected from the top list.

You set the parameters for a cloud type by entering numbers in the four boxes and then clicking the **Add** button:

Amount is the number of circles to add. Each circle represents the location of a certain amount (density) of that cloud type. The circle does not represent the shape or size of the cloud.

Distance is the decimal fraction of the distance from directly overhead to the horizon—0.0 is overhead, 0.8 is low on the horizon.

Direction is the azimuth in degrees of the circles— 0 is north, 90 is east, etc.+

Spread determines how close a group of circles will be clustered when they are added—0 puts the circles nearly on top of each other, 0.2 will spread them out by a random amount that depends upon the cloud type.

Each time you click the **Add** button, another group is added.

Click the **Clear** button to remove all of the clouds *of the currently selected type.*

Clicking the **Relax** button randomly alters the positions of all circles *in the current altitude region.*

Clicking the **Update** button advances time by 5 minutes each click so that you can check how the initial clouds will move out. Important: First use the **Save** button to save your setting so that you can restore it with the **Load** button. Click the **[Weather Presets]** button in the Save preset as window.

You will need to experiment with the weather settings to get an understanding of their behavior.

Contents

Fog

The **Fog** tab in the WeatherSim Settings panel allows you to set the type and amount of fog by two methods: Manual or Auto Fog. You toggle between methods with the **Switch Mode** button.
{9 Fog}

Auto Fog produces a uniform fog over the environment. The slider increases the density of the fog, effectively setting the visibility distance.

The Manual mode allows you to produce any kind of fog you want, including orange fog, useful if we ever get a Los Angeles environment.

You can control ground fog separately from sky fog.

Changing Weather during a Mission

You can change the weather by means of a trigger during a mission.

In a trigger at the point in the mission logic where you want the weather to change, you can use these **actions**:

Weather – Change Wind Direction This changes where the weather comes from and leaves.
The wind blows **from** the indicated direction.

Weather – Change Wind Speed This changes how fast the new weather moves in and the old weather exits.

Weather – Set Low Cloud You set the ratio between the two types and the fraction of cloud cover.
Weather–Set Middle Cloud

Weather–Set High Cloud
These three actions change the ***generation*** *of weather below the horizon* over the number of seconds set in the blendtime. The effect will not be apparent until the weather moves in—determined by the wind speed.

Weather–Set Precipitation Changes the precipitation over a period of time.
The **Precipitation** button offers five choices: Dry, Drizzle, Rain, Heavy Rain, and Cloudburst.
The **blendtime** is the time in seconds over which the precipitation changes.
Remember: If you want anything but "Dry", you must have clouds overhead that can give it to you; otherwise, you must wait until they roll in. Also, if you want snow or hail, the temperature must be set low enough. If you want rain, the temperature must be high enough.

Weather– Set Temperature Changes the temperature in degrees Celsius over **blendtime** seconds.
You can use this to change rain into snow, or vice versa. Go to "Extremes"—+20 to -20 degrees.

Contents

Ocean Settings — Sea State

The seventh button from the left on the main toolbar (blue waves) opens the Ocean Settings panel. The Dynamic Settings tab lets you adjust the waves in your mission. The Environment & Global settings are above your pay grade. {8 Ocean Settings}

There are three groups of waves: Small, Middle, and Large. The relative heights of each group can be set independently over a range from 0 to 2.000. Usually you'd set one group to 1.000 as the standard.

The speed of each group can also be set independently, which determines their wave length.

There is a WaveScale slider that sets the max height after you've adjusted the other sliders for the kind of sea you want.

You need to just play with the settings to get a feel for how they affect the sea state.

After you fiddle with the settings, the panel will be reset to their defaults when next you re-start SSE.

The Wind Direction—set in WeatherSim—determines in what direction the sea runs. The wind velocity does not seem to affect the sea state.

The direction of the waves is somewhat randomized to avoid the checkerboard appearance of SS08.

You want to get the urge to try humongous waves out of your system before you attempt serious missions. The behavior of ships in an extreme sea is not all that pretty. Small boats react in a realistic manner—they sink. But before they go down, they do some weird things.

If you want to gauge the height of your waves, use the BoxAreaEntity. You can set one or more with various Y dimensions. They sit at mean sea level, so Y is the peak-to-trough distance.

Ocean Change during mission

You have one **action** to change the sea state during the mission: Ocean – Blend waveheights.

You must set heights for all three groups, and the number of seconds over which the change takes place.

While you are setting this action, you can have the Ocean Settings panel open for reference. You can drag it out of the way by the title bar.

You cannot change the wavelengths during the mission.

Contents

Lights

You can control the navigation and mast lights in the mission logic, and you can test the state of the lights:

Controlling the lights

The action: Ship – Set Lights State turns the mast and navigation lights on/off.
Ship is the ship whose lights are to be set. **Lights State** is selected from a drop-down list.

You can set the lights for every ship in the current environment with this action:
Mission – Set Lights State of Ships in Current Environment to
Lights State — Selected from the drop-down list.

Checking the lights state

You can check that the lights are in a specified state with the condition:
Ship – Check Lights State At present, a bug prevents setting the state.

Contents

Cinematic Camera

You can use this camera to insert a **Cut Scene** into the mission—leaving the current camera on the player ship and switching to a motion-controlled camera for a certain number of seconds, and then returning to the original camera on the player ship.

You place a Cinematic Camera in its starting position by selecting **CinematicCamera** from **Static > ScenarioEntities** (or LogicEntities) in the Entity creation panel.
Rotate the gizmo to face in the initial direction.

Place a **PointEntity** on the object or spot that you want to be in the center of the field of view. Adjust it to be at the correct height and name it something like: AimCC1.

Use these actions in a trigger to start and control the movements of the camera:
CinematicCam – Face Location: Set the specified camera to face the location of the specified entity at a specified height over a specified time.
Camera is the CC that you placed; **Entity** is AimCC1, the PointEntity you placed to mark the center of focus; **CameraHeight** is the initial height in meters; **BlendTime** is the number of seconds to take moving into position.
CinematicCam – Switch To Cinematic Cam: Switches view to the **CinematicCamera.**
You can do this either before or after Face Location, depending upon whether you want to pan to that location, or start there.
CinematicCam – Transport and face location Cinematic Camera: Transports **CinematicCamera** to the PointEntity in **DestinationPoint** at a height of **CameraHeight** meters, staying focused on **Lookat entity** and taking **BlendTime** seconds to do it.
Note that **Lookat entity** can be the same point as before to keep that point in focus during the move, or it can be a different point.

To return to the camera on the player ship, you use this action:
CinematicCam – Switch Back to Previous Cam: Switches view back to the previous camera.

Important: You cannot add this action in the same trigger that has the actions listed above. Those actions merely set up the action that they describe. They complete immediately rather than after a period implied by their BlendTimes, so the switch back to the player camera would happen immediately.
You need to create a timer in the Variables panel with a **Target Time** equal to the number of seconds you expect the Cut Scene to last.
Place a Timer–Start action either before or after the actions for the cut scene.
Follow that trigger with a **State** and another **Trigger** to determine when the timer expires:
Condition: Timer– Expiration Check
Action: CinematicCam – Switch Back to Previous Cam

There are two more actions that you can use to control the cinematic camera:
CinematicCam – Set Height: Changes just the height of the camera over the **BlendTime**. You can use this for vertical panning.
CinematicCam – Transport Cinematic Camera: This moves the camera to a new Point Entity at a designated **CameraHeight** over **BlendTime** seconds. However, instead of pointing the camera at a Point Entity, The vew direction is set by rotating the gizmo on the **DestinationPoint.**
You can do circular panning in place with one or more of this action. (continued)

Note: When you do a Cut Scene, the player ship has no one at the helm during the scene. If there is a chance that it could get into trouble during that time, you might want to make it AI with a goal that will avoid any trouble. When the Cut Scene is done, you can return control to the player.

Oil Trails

To cause a ship to leak an oil trail, use these actions:
Ship – Oil Trail Start **Entity** is the Ship
Ship – Oil Trail Stop **Entity** is the Ship

Photos

There are three conditions to use when an objective tells the player to take a photo:
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.

To use the last two, you must place a **PointEntity** in the scene so that it would be included in any acceptable photo.

The JPG photos can be found in: [your documents]> ShipSimExtremes UserData > Photos.

Contents

Cargo

Cargo Net

The Cargo Net travels a path between two invisible entities such as the **SphereAreaEntity** with a 3-meter radius. The net needs 3 meters of clearance below its suspension point.

Note: While you can make the Cargo Net travel between any two ships, or any two places, the net travels on an invisible line. To provide a visible line, a mooring line is strung between invisible bollards on the ships. At present, only the whaler and the factory ship have these special mooring points.
To use these ships:

1. Place the whaler on the port side of the factory, on the same heading, with their sterns approximately aligned. Make the open space between them about 30 meters. You can use a temporary SphereAreaEntity with a 15m radius as a gauge.
2. Select the factory ship and click the blue Moor button on the main toolbar to light the point near the stern of the factory.
3. Click the point, and then the one on the whaler, to get a line between the two ships.
4. Adjust the position of the factory ship to get the line perpendicular to the ships and stretched taut. (The net won't follow a catenary.) Unfortunately the tension in the line causes the ships to drift together. You need t find a compromise amount of sag.

   A better solution may be found in an update.

Place the cargo net at its starting position in the same way you set a bollard:

1. In the Entity creation panel, click the Static tab and the> next to Cargo.
2. Select Net01 and place it on the water near where you will use it.
3. Adjust its height (Y) as you do for bollards, and then move it into the starting position.

Create a SphereAreaEntity with a 3-meter radius and place it centered on the top part of the cargo net. Give it a name like NetStart, and un-check Visible.
Create a SphereAreaEntity with a 3-meter radius and place it where you want the cargo net to stop. Give it a name like NetEnd, and un-check Visible.

If either of the those spheres is on a ship, you must attach it with an action in the initializing trigger: Select Entity – Attach to Entity as the action. **Child** is the sphere. **Parent** is the ship.

At an appropriate place on the logic diagram to start the animation, branch from the trigger to a new **State** and **Trigger**. You need two actions:

Entity – Interpolate – Set Target **Entity** is the cargo net **Target** is the SphereAreaEntity **NetEnd**.
Entity – Interpolate –Start **Entity** is the cargo net.

IF you want to make a return trip, place another State & Trigger pair below with the condition:
Entity – Other Entity Within Range **Entity A** is the cargo net **Radius** is 1m or less **Entity B** is the SphereAreaEntity **NetEnd**.
This allows you to start the return trip with just one action: Entity – Interpolate – Set Target.

Aircraft

Plane

The four-engine passenger jet is a **Static** entity under **Aircraft**—complete with long contrails.
Place it in the environment facing in its initial direction.
In its **properties**, set **Y** to the correct altitude in meters. Un-check **Visible**.

The plane will fly by interpolation to a Point Entity goal at a speed that is proportional to the initial distance to the goal. To get a slower speed you need to space the goals closer together. You can adjust the first goal by trial to get the desired speed, and then replicate the distance to successive goals.

Place a **PointEntity** for the first goal and set **Y** to the desired altitude. Rotate the gizmo to point in the direction that the plane is flying—or the plane will rotate when it gets to the goal.

In the trigger to set the plane in motion, you need these actions:
Entity – Set Visibility **Entity** is the plane; **Value** is 1 to make the plane visible.
Entity – Interpolate – Set Target: **Entity** is the plane; **Target** is the PointEntity goal.
Entity – Interpolate – Start: **Entity** is the plane. This sets the plane in motion.
If necessary, save and run to adjust the distance for the desired speed.

For the next segment of the plane's path, add a State and another trigger with this condition:
Entity – Other Entity Within Range: **Entity A** is the plane; **Entity B** is the goal; **Radius:** try 20 meters.
This detects when it's time to set the next goal.

If necessary, use a gauge to lay out the next **PointEntity** goal and rotate the gizmo to line up with the path.
In that last trigger, use this action to send the plane down the second segment:
Entity – Interpolate – Set Target: **Entity** is the plane; **Target** is the new PointEntity goal.

It is best not to make a course change with the plane because when it changes heading the long contrails swing around, which looks weird.

Helicopter

The helicopter is flown using the same method that you use for the plane. The main difference is that the chopper can move vertically and make sharp course corrections.

Vertical ascent

If you want the chopper to rise vertically before heading to the next goal, you must place the PointEntity very carefully to avoid having the chopper rotate to a strange heading before it begins to rise:
If the path is not exactly vertical—impossible to achieve—then there will be a horizontal component to the path (the projection of the path onto the ground). The chopper will rotate to align itself with that horizontal vector before it rises.
You should make that vector aligned with the heading of the chopper on the ground to prevent any rotation. You also want that vector to be as short as is practical. One easy way to do that:

1. Initially set the point's Y value to be the same as the chopper's (3m, if on a quay).
2. Place the point slightly ahead of the chopper's nose and zoom close for a top-down view.
3. Rotate the environment so that the chopper is heading straight toward the top of the screen.
4. Move the gizmo so that the point lies on the centerline of the chopper and the orange circle intersects the tail boom just aft of where it joins the body.
5. With the point's properties panel displayed, click in a neutral spot to disconnect the gizmo from the point.
6. Set the **Y** value for the desired altitude, and then select the point in the Entities in mission panel to reconnect the gizmo. Save now, before the point gets accidentally moved.

Contents

Whales

Whales are AI entities under Animals.
When you place a whale in the water, it will be at a depth of 7.5 meters (Y=-7.5). Do not change Y.
The whale will be visible from the surface in clear water such as the Bora Bora lagoon.

The normal mode for the whale is to travel toward a **PontEntity** and stop when it gets there.
While traveling, the whale will periodically breach, blow, and slap the water with his flukes.
Technically, the whale does not truly breach; it just surfaces for a short period.

The whale has another mode called "Dive", in which he does not surface again after being frightened by a ship coming too close.

A third mode, fleeing from a whaler is under test.

Whale actions:

Whale – Add target: Click the **Whale** button and select the whale; click the **StaticTargetEntity** button and select a **PointEntity** to head the whale toward it.

When the whale is close to its target, you can repeat that action to head it toward another target. You can thus make the whale travel a defined course.

Use the condition Entity – Other Entity Within Range: to detect when the whale is close to its current target:
Click the **Entity A** button and select the whale.
Click the **Radius** button and enter the distance in meters from the target at which the whale should change course.
Click the **Entity B** button and select the **PointEntity** that is the current target.

Whale – Apply speed: **Speed** is in meters/second. Whales do not obey COLREGS. (Even if they are smarter than the average Third Mate)

Whale – Start: Do this *after* you have set a target.

Whale – Dive: Use the Entity – Other Entity Within Range: condition with a ship, or with a **PointEntity** along the whale's course, to cause it to stay submerged.

Whale – Set Whaler: All whales will flee from the specified whaler. Under test at present.

Contents

Variables

There are various types of variables available for storing numerical values and text strings:

Strings Store strings of alphanumeric characters for use as messages, etc.
Objectives are special arrays of text strings that show the notes for mission objectives.
Integers are whole numbers that can be positive or negative.
Floats are numbers that can have decimal fractions.
Booleans are logical variables that can have only two values: 0 (false) or 1 (true).
Vec3 is an array of three Floats.
Timers are integers that hold seconds and automatically decrement each second until they reach zero, and then increment each second.
Entity The ID of an entity in the mission can be stored and used in parameters for conditions and actions.

To create a variable in the Logic Editor, click the (+) button on the Variables toolbar.
In the Configure Variable panel, select the **Variable Type** from the drop-down list and give it a **Variable Name**.

Setting variables during the mission.

This action allows you to set a *previously defined* variable to either a specified value or to the value of another variable:
Variable – Set {Type} Value where {Type} can be Boolean, Integer, Float, String, or Entity.
A is a previously defined variable selected from the drop-down list.
B can be set either to a value you specify, or to another variable of that type selected from a drop-down list. You choose the method by clicking one of the radio buttons.
For an **Entity** variable, you can use the Select From Scene button.

Manipulating variables

You can use these actions with variables:
Variable – Append to String put **String A** on the end of **String B** (a defined string selected from the drop-down list); **String A** can be another selected string, or can be typed into the box.
Variable – Integer to String converts **Integer B** into **String A** so that you can display it in a message.
Variable – Perform Integer Operation Does arithmetic (+,-. *, /) on **Integer B** and **Integer C** and puts the result in **Integer A** (selected from the drop-down list)**.**
Variable – Perform Float Operation Likewise with Floats

Comparing variables

You can use these conditions to compare two variables of the same type:
Variable – Compare {Type} where {Type} can be Integers, Floats, Booleans, Strings, or Entities.

Contents

Conditions

A list of conditions not explained elsewhere:

Entity – Controller Comparison — **Comparator** is Equal To or Unequal To
Controller Type is Player, AI, or Static

Entity – Speed Check — **Comparator B** Equal Or Greater Than, or Equal Or Less Than
Speed C a Float (i.e. 10.5) or a Variable of type Float

Checks the speed in knots of the vessel. Don't use an exact comparison like Equal To.

Entity – Speed Reached (Knots) — **Speed (knots)** is the threshold speed in knots
Entity – Speed Reached (m/s) — **Speed (m/s)** is the threshold speed in meters/second

Entity – Looked At
Entity – Looked Away From

These check whether, or not, an entity is in the player's field of view. This may not mean that the entity is visible to the player—depending upon the aspect ratio of the display, the zoom level, or whether the entity is obscured by a mast or window frame.

Ship – Is Damaged

Contents

Bridge Components

Vessel	Helm	Engine	Bow Thruster	Stern Thruster	RADAR	GPS	Water cannon
Agile Solution	24	22	23				
Apollo	2	1					
Arie Visser	2	1					
Bangalore	1	2					
Billy Greene	2	1					
Bugsier 2	39&40	39&40					
Cartagena Delight	8	1	17				
Coromuel	1	54	2				
Cutter	1	3	47				
Davit	3	5					
Esperanza	9	10&11	41				
Fairmount Sherpa	43	25	24	23			57
Flambee	8	1					
Fortissimo	9						
Freedom 90	1	17	18	16 Skirt			
Greenpeace RIB	1	2					11
Herkules Atlas	21	14	25				
Inferno	1	2					
Jumbo Javelin	16	6	7	19			
Latitude	1	29	48				
Mamba	3	2 FP					
Mare Australe	1	2	3				
Ocean Prince	35	2	33	34			
Orient Star	1	2	5				
Oriental Blackbird (Sta)	1[9]	2					
P6	1	2	26				
Pride of Rotterdam	1	2	39				
Protector	1	3	55				54
Rainbow Warrior	2	1					
Red Eagle	21&23	22&24					
Red Jet 4	24&23	24&23					
Rowboat - OB	1[10]	1					
RPA12	16	5	18				82
Searay	N/A						
Sigita	41	40	42				
Vermaas	44	1	2	3			
VSTP7	1	2					

Tips:

Enlarging/Scrolling the Logic Editor

To increase the size of the Logic Editor panels, left-drag the side and bottom borders of the editor. To increase the width of the right hand panels, left-drag the left border to the left.

To shift the state diagram in order to see other parts, right-drag on the blue background.

If the mouse pointer leaves the blue background while dragging, you may need to click the blue background to un-stick it from the mouse.

To zoom in or out on the diagram, click on the blue background and then use the mouse wheel.

{6 Scenario Graph & Nodes}

Objectives

InfoMessage

You can display an informative message to the right of the Objective message:
Objective – InfoMessage **Objective** select from drop-down list; **Message** text.

Measured Nautical Mile —Ruler

For laying off accurate distances in the ME, you can use the BoxAreaEntity:
Set the **Z** Scale to 1852 meters to get one nautical mile. **X** can be zero if you just need a line of a certain length.
If you want a nautical-mile course for measuring speed, Set **X** to at least 100 meters. You can trigger start/stop timing with Area – Contains Player Ship and Area – Does not Contain Player Ship.

Contents

Endnotes

[1] There is only one ocean environment in 1.3 that can be used for long ocean voyages—there are three in SS08. Not being able to set a believable Lat/Lon for, say, the Pacific is a handicap. Also, having only the one piece of ocean means it must be used in a reentrant fashion for long ocean voyages. It would be helpful if the base coordinates could be changed by an action in the ME when jumping back for a new traverse.
[2] For working the examples, choose Apollo or RPA12.
[3] This will be the language for Objectives, Messages, etc.
[4] Select tags from the right-hand list and click the Add button. Tags are shown when you select the mission from the **Select Mission** list in SSE.
[5] If the Entities in mission panel is empty after your environments appear in the Environments panel, you have not waited long enough for the mission editor to finish loading. You should **quit to main menu** and restart the editor. Wait longer before you try to load your mission.
[6] In 1.3 you may find that you can't select by clicking in the Entities in mission panel if another entity in the environment is already selected—has the gizmo. Deselect it by clicking a neutral spot in the environment.
[7] CTW is the course to waypoint. More accurately, it is the compass bearing of the waypoint, not necessarily the course to follow.
DTW is the distance in nautical miles to the center of the waypoint. The trigger happens when the ship reaches the trigger radius, which may be hundreds of meters ahead of the center.
[8] You can also open the logic editor by clicking the fifth button on the main toolbar.

[9] Oriental Blackbird has no bridge, so setting her controller type to "Player" will not allow you to control her.
[10] The rowboat has an outboard motor; it is both engine and steering control.