

P0lar_bear's SP Tutorials

Tutorials by: p0lar_bear

Recovery by: Nukeist

BasicSoloMap

Making a Solo Scenario

by p0lar_bear

This tutorial isn't going to be very in-depth; it'll tell you how to get a scenario tag to load as a solo map.

First, in Guerilla, make sure the map's type is set to solo. Then, open it in Sapien, and place ONE player spawn point. Make sure that all four of its types are set to none, and I also think a team index of 0, to be safe. Save, and compile.

Now, it'll load as a solo map when you load it with the map_name command.

NOTE: You don't HAVE to set the map type to solo in order for this to work. But if you want a solo-only map, that won't load thru the multiplier map list, set it as solo.

SoloCutscene

Making a Cutscene for Solo Maps

by p0lar_bear

This is going to be complicated... grab some popcorn, 'cause you're going to be here for a while. Also, make sure you have a valid actor and actor_variant tag. This is no problem if you have the HEK+, or TheGhost's massive tagset.

There are two ways to get bipeds to move around in cutscenes. The first way is by creating custom animations in 3D Studio Max. The second way, which is a bit more

complicated, is by attaching bipeds to an AI encounter, and making them execute a command list. I prefer the AI method for basic cutscenes; they will usually look better than if I got a hold of a cyborg biped rig and tried to animate it. :X

Anyway, to start, pick your characters to use in the cutscene. For this one, we'll make the Master Chief run into the red base of Blood Gulch, look at another cyborg, then kill it.

First, make a copy of the cyborg.biped tag, and rename it to something like

"cinecyborg.biped." Now, open your new tag in guerilla, and scroll down until you see into about aiming and looking velocities and acceleration, which is in the "camera tracks" section of the biped tag. Set them as follows:

Aiming Velocity Max: 360

Aiming Accel Max: 720

Casual Aiming: .8

Looking Velocity Max: 720

Looking Accel Max: 1440

If we did not set these, the Master Chief's looking and aiming wouldn't look good. His head and gun would jerk around instead of smoothly gliding into place. Now, save the tag.

Go into tags\levels\test\bloodgulch, and make a copy of bloodgulch.scenario. Rename it to something like "sologulch.scenario." Open that in Guerilla, and use my tutorial on making a solo scenario to make the map work as a solo map. Save, and exit Guerilla.

Open your newly created sologulch tag in Sapien. Once it loads up, immediately add your cinecyborg tag to the biped palette. Place one cyborg outside of red base, a bit away from a side door, and place another on the flag marker. Name the one outside "chief", and the one inside "deadman".

Now, create a new AI command list set (mission/AI/command lists), and name it something like "cyborgkill". First thing to do is set some command list points for the Chief to move to. Set one in front of the side door, one at the wall inside the base, one at a doorway inside the base, one right next to deadman, and the last one you will place on deadman's spot, but use the properites palette to raise it to a height of about .6 (z axis).

Next, make a command list. To make each command, hit "new instance" on the heierarchy view, and then select the command you wish for the AI to execute in the properties palette, and how to execute it. (NOTE: Be careful when switching between commands; sometimes Sapien will crash if you switch from a command with a lot of options, like vocalize, to a command with very few or no options, like pause. When changing a command, make sure to always select the first option on the drop-down box, if there is one.)

Try to use this knowlege to create this command list; poke around a bit and you'll find the correct options:

Go to point 0 and keep moving

Go to point 1 and keep moving

Go to point 2 and stop

Look at point 5 for 1s

Wait until told to advance

Go to point 3 and stop

Wait until told to advance

Shoot at point 4 for 1.5s

wait until told to advance

Save your scenario, quit, and reload sapien. The reason for doing this is because AI doesn't recognize command list points unless you restart the engine. There may be other ways, but this is known to work 100% of the time.

Once sapien is loaded up, add an actor variant tag to it; I usually use "marine_armored assault rifle", but anything will work. create a new AI encounter and name it "cinematic1". Make sure it is not initially created, and initially blind, deaf, and braindead, then set the team index to 1 (player). Add a squad to it, and name it cineborg. Give it the actor type you added to palette, and that should do it. Save the scenario, and we'll move onto setting some camera points.

Go into sapien's camera mode, and find a good point to show the master chief running into the red base. Position the camera as needed, and hit the spacebar to save the camera's position. Name it "cam1". Now, to get good camera points, we'll attach the chief biped to the cinematic AI encounter. Bring up the console and type in (ai_attach chief cinematic1). Now, type in (ai_command_list_by_unit chief cyborgkill), to make the chief biped execute the command list. Find a good spot to show the chief running through the base that doesn't reveal the deadman biped. After getting that camera point (name it "cam2"), get another one that shows the chief looking at the deadman (name it "cam3"). To get a good close-up of the chief owning the deadman, type (ai_command_list_advance_by_unit chief) into the console, and the chief should run up to the deadman. Find your camera point and save it (as "cam4").

After this is all done, make a trigger volume (mission/game data/trigger volumes) named "cutscenestart" outside of the red base entrance the chief goes into, so the cutscene will initiate when the player walks into the trigger. Also, make one way the hell off somewhere in the map that the player can't get to without hacking, and name it something like "hide"

Finally, go into the Flags section, and add a flag somewhere that can't be seen from the cutscene area, and call it "hideplayers." Then, make one on the chief's final position and call it "showplayers".

Now that you have your AI, command lists, trigger volume, and camera points set up, let's make a script to make it work. Assuming you used all suggested names, let's go with this:

```
;;; This first script removes the bipeds used in the cutscene, so
they're not visible until the cutscene starts (script startup
hidethebipeds
(object_destroy chief)
(object_destroy deadman)
)
(script_startup cutscene
(sleep_until (volume_test_objects cutscenestart (players)) 5) ; Stop
script execution until a player walks into the trigger
(fade_out 1 1 1 30)
(player_enable_input 0)
(sleep 60)
(volume_teleport_players_not_inside hide hideplayers) ; Hide the
players by teleporting them out of the area
(cinematic_start)
(cinematic_show_letterbox 1)
(object_create_anew chief)
(object_create_anew deadman)
(camera_control 1) ; Let the script control the camera
```

```

(camera_set cam1 0)
(sleep 30)
(fade_in 1 1 1 30)
(ai_attach chief cinematic1)
(ai_attach deadman cinematic1)
(ai_command_list_by_unit chief cyborgkill)
(sleep 60)
(camera_set cam2 0)
(sleep 30)
(camera_set cam3 0)
(ai_command_list_advance_by_unit chief)
(sleep 30)
(camera_set cam4 0)
(sleep 30)
(ai_command_list_advance_by_unit chief)
(sleep 30)
(unit_kill deadman) ; This is included to insure the biped dies, in
case the weapon's fire doesn't kill it
(sleep 60)
(fade_out 1 1 1 30)
(sleep 30)
(object_destroy chief)
(camera_control 0) ; Put the camera back onto the player's FPV
(volume_teleport_players_not_inside hide showplayers) ; Puts the player
back where s/he should be
(cinematic_show_letterbox 0)
(cinematic_stop)
(sleep 30)
(fade_in 1 1 1 30)
(player_enable_input 1)
)

```

Save that to a .hsc file in the folder data\levels\bloodgulch\scripts. When done, go into sapien, and click "Compile Scripts" in the File menu. If you get any errors, debug them using a little common sense. Once the scripts are compiled correctly, save the scenario, exit sapien, and compile the map. Run Halo CE, and at the main menu, bring up the console and type in "map_name sologulch." Walk into the spot where you placed your trigger volume, and the cutscene should play. Sit back, relax, and enjoy. Even if the cutscene looks like crap, just take a bit of pride in knowing that you now know HOW to make a cutscene, and you can build upon it to make it even better.

SoloRules

Rules of Thumb for Making Solo Maps

by p0lar_bear

Here's a list of some tips, tricks, and other stuff with scripts to make a solo map good.

Friendly AI

Whenever you make an AI encounter you want to be firendly with the player, add in the (ai_allegiance) command to make an AI team friends with the player. For example, if you

make a solo map in which you are a Covenant Elite, and would like Covenant encounters to buddy up with you, use this command in a startup script: (ai_allegiance "player" "covenant"). Valid teams are "player", "human", "covenant", "flood", and "sentinel".

Checkpoints

To make a map less frustrating to play, you have to know when to make the game do a checkpoint. Usually, I'd put checkpoints before and after BIG battles, long sequences, or when you have to traverse a large distance and would like to not have to drive/walk a long ways again, like in the Silent Cartographer. The checkpoint script command is (game_save).

Objective Parameters

Let's say there's a vehicle, person, or position you have to protect, or else the mission fails, like protecting Capt. Keyes in the end segment of Truth and Reconciliation. How they did that was, they placed a Keyes biped down with a name, and attached it to the AI encounter of the marine squad that follows you around. There was a script that continually checked for Keyes's health, and if it hit 0, the game would revert to your last checkpoint.

To make a game continually check a unit's health, make a continuous script with the command (unit_get_health <unit>) in an (if) statement, so when the unit dies, the game is lost. Here's a sample, which assumes your biped's name was "protectme":

```
(if (= (unit_get_health protectme) 0) (game_lost))
```

You can also make it wake a dormant script that uses (camera_control 1) and (camera_set_dead <unit>) with (game_lost) to show how badly you phailed and rub it in your face. >:D

Friendly to all

Make sure your map can be played by all. Do not force the player to use a "skill" weapon in an area that may cause a newbie to get killed 50 times before they finally get it. You should usually provide alternative routes and strategies. For example, before an enemyinfested

area, stick some dead marines with dropped pistols, ARs, shotguns, maybe rockets, and ammo. And maybe there's a cliff next to you with a sniper rifle on it? You don't have to set all of that down, but don't just lay down a plasma pistol and expect everyone to be able to get through the area. Also, do not make enemy AI overpowered or overwhelming in number without a powerful weapon or vehicle to help the player get by. Keep in mind that it takes at least 8 headshots to an fully sheilded elite with a pistol to kill, and not 3 like the cyborg biped.

TriggerVolumes

Trigger Volumes for Dummies

by p0lar_bear

The first thing you need to know about making ANY kind of single-player map is how to use trigger volumes effectively. I will explain all of this here. All you need is Sapien, and

a text editor like notepad. As an added bonus, you could also grab the HEK Plus app by SteelIXB and rip some actor/actor variant tags out of a Halo Campaign map, or you could download TheGhost's large tagset, which includes every element used in the Halo campaign.

First, let's make an AI encounter and squad that will be placed when the player enters the trigger volume. If you need help with setting up an AI encounter, watch Aztec's video tutorial. Just make sure that when you make the encounter, the "Not initially created" flag is checked, and let's give it a name of... oh, encount_1.

Now, let's place a trigger volume in an area that the player will **have** to go through. When it is initially placed, it may be smaller than you need it to be. If you want to make it larger, click and drag on a side to move it, thus making the volume bigger. Then when done, name it something easy to remember. Let's name this one trig_1.

Now, you may notice that there is nothing you can do with the trigger volume in Sapien to make it place an encounter when a player enters it. The reason why there's nothing there in the original version and the modified version of Sapien is because trigger volumes are, always have been, and always will be, referred to in scripts.

So, let's open up notepad, and create a script. The best and neatest way to make a trigger volume-related script is to make a startup script that sleeps until the player walks into the trigger volume, then continues executing. So, let's make this trigger volume create the AI encounter, wait until the encounter has been totally wiped out, then make a checkpoint and play the "killtacular" sound just for kicks.

```
(script startup h4xmb
(sleep_until (volume_test_objects trig_1 (players))15) ; Pauses script
execution, and checks every 1/2 second to see if a player walks into
the trigger "trig_1".
(ai_place encount_1) ; Places the ai encounter "encount_1"
(sleep_until (= (ai_living_count encount_1) 0) 15) ; Pauses script
execution, and checks every 1/2 second to see if the encounter
"encount_1" has been totally wiped out.
(game_save) ; Checkpoint... done.
(sound_impulse_start sound\dialog\multiplayer1\killtacular none 1) ;
Killtacular!
)
```

Save the file into your level's scripts folder (if your map's folder is in data\levels\mymap, you would save the script into data\levels\mymap\scripts) as myscript.hsc. Then go back into Sapien and compile the scripts (File | Compile scripts). There should be no errors.

Now, save the scenario and compile your map in Tool.

Now, start Halo CE and when you get to the title screen, load the map by typing "map_name <your map's name here>" into the console. When you load the level and walk into the trigger volume, the encounter should be placed, and when you kill it, you will get a checkpoint, and the Killtacular voiceover should play.

NavPoints

Navpoints/Waypoints Tutorial

by p0lar_bear

Meh. Same difference. Via scripts, there are four ways to set a navpoint (the engine refers to them as waypoints), and those are by attaching the waypoint to either a cutscene flag or an object, and have it visible to either one unit, or a team. To set a waypoint, use (activate_nav_point_flag) or (activate_team_nav_point_flag) to attach the waypoint to a cutscene flag, and use (activate_nav_point_object) (activate_team_nav_point_object) to attach a waypoint to a named object. When done with a waypoint, use a (deactivate_nav_point...) or (deactivate_team_nav_point...) to turn it off. Since I don't know how to refer to the players as units, I usually set a team waypoint. The full syntax of a team waypoint command is (activate_team_nav_point_flag/object <waypoint type> <team> <flag/object> <vertical offset>).

<waypoint type> can be, by default, "default", "default_red", "flag_blue", "flag_red", "target_blue", "target_red", "skull_blue", "skull_red", "crown_blue", and "crown_red". Singleplayer maps should use only the default, and maybe, if you have to kill a certain enemy, the target nav points. the "default" type and red types of waypoints should be used on urgent things, like the exit on an area that would have a timed escape, or an enemy that is about to own you.

<team> should always be set to "player", if you want the player to see the waypoint!

<flag/object> is what you are attaching the waypoint to, which depends on the type of command you're using. Simply put in the name of the desired object or flag you wish to attach the waypoint to.

<vertical offset> is how much higher on the z-axis the waypoint is placed from the object or flag's marker in sapien. If you attach it to something on the ground, like a door, biped, or weapon, make sure to set it to about .6 or higher, but if its something like a picture on a wall, or if the flag/object is otherwise not on the ground, set it to 0.

(deactivate_team_nav_point_flag/object) is a bit different. The syntax for it is (deactivate_team_nav_point_flag/object <team> <object/flag>), where <team> is what team you wish to not see the navpoint anymore, and <object/flag> is the object or flag the navpoint is attached to.

So, let's say we wanted to set a normal waypoint at a position we want the player to go to. Lay down a cutscene flag, name it nav_1, then add a script in your map that has this command: (activate_team_nav_point_flag default "player" nav_1 .6)

Now for something a little more useful. Let's make it so we put a waypoint on a warthog, and it turns off whenever the player is in the driver seat, and turns on if the player is ejected. Place a warthog and name it drive_me. Now, make and compile this script: (script continuous hog_checker (if (vehicle_test_seat_list drive_me "wdriver"

(players)) (deactivate_team_nav_point_object "player" drive_me)
(activate_team_nav_point_object "default_red" "player" drive_me 1)))

In theory, that will constantly deactivate the waypoint while a player is in the driver seat, and then constantly activate it when there is no player in the driver seat. I don't know if more than one waypoint can be put on the same object or flag, but if it doesn't, then this should work fine.

Anyway, read up hs_doc, or the HS Bible, and find out more ways on how to make and use waypoints effectively.