

Combat Starter Kit

Starter Kit

Table of Contents

1.	The existing zone	2
2.	Modifying The zone	2
3.	Creating a new zone	2
4.	Options	2
5.	Prefabs	2

Zone setup guide

This guide explains how to setup capture points, flags and the red and blue bases in a CSK map. Topics such as how to use the TGE world editor and terrain creation are outside the scope of this guide.

1. The existing zone

Combat Starter Kit ships with 1 main zone called 'BFTTerrain.mis'. This zone is built at the standard terrain size of 8 with tiling turned on, although none of the area outside the initial land block is accessible in the current mission.

Bases

The zone has 2 player bases in opposite corners of the map, the northeast corner and the southwest corner with a river running through the middle.

Trees

In the center of the map (coordinates 0,0), there are 3 FxShapeReplicator instances that supply all the trees and rocks for the whole zone. Liberal use is made of fxReplicatorBlocker instances to ensure the roads are clear of trees. The shape replicators have been slightly modified to use a "MaxHeight" attribute, this functions like a tree line for each replicator. The leafy trees have a lower tree line, while the evergreen trees have a higher tree line giving a little more realism to the trees.

Roads

The terrain has roads carved through it. The mission layout makes extensive use of fxShapReplicatorBlocker to keep the roads free of trees. This is a resource from the GarageGames site here:

<http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=6467>

1.1. Spawn points

Each base has 3 spawn points, 2 on land and 1 in the air. The aerial spawn points will cause the player to parachute down. Each spawn point is an instance of SpawnSphere. The data block is set to SpawnSphereMarker or ParachuteMarker for aerial spawns. Each spawn point also belongs to a capture group, or simgroup.

1.2. Flags

Each base also has 1 flagpole. The mission file only has the flagpole dts object, but each flagpole has a number of dynamic fields setup. The flagpole has an animated mount point that a flag object is mounted to on startup. Every flag has to have an ObjectiveNumber and ObjectiveName set. Current objective numbers are 1 to 8. The flags also have a special field called "dynamicFunction". This is used to setup each flagpole on mission startup. The dynamicFunction field is the name of a function that is run during mission startup. See GameUtils.cs for a description of how this works.

The red and blue flags are given the dynamic function "SetRedTeamOwner" or "SetBlueTeamOwner" to setup the red and blue base flags at the beginning of the mission. The rest of the flagpoles are given the function "AreaTriggerInit". This function sets the flags

to neutral on mission startup.

1.3. Capture Areas

Each set of spawn points and its associated flag are contained inside a capture area, or Simgroup. When a flag is captured, all the spawn points in its associated capture area are made available to the capturing team. This simple mechanic is essential to making capture points work properly.

1.4. The Radar

Each base also has a radar station associated with it. The radar is setup with a data block named 'radar' and a single dynamic field named 'TeamID'. This field should contain a 1 or a 2. 1 for red team, 2 for blue team. Only 1 radar per team is supported, but the radar can be anywhere on the map.

1.5. The Control Panel

Each base has a master control panel hidden inside the supply depot. The control panel is a dts object called 'Panel' with a datablock DefaultPanelData. Destroying a control panel gives the team 300 points. Multiple control panels are possible.

1.6. Repair Areas

Each base has 2 helipads and 1 vehicle pad. Driving a vehicle over the vehicle pad will repair and reload ammo for the vehicle. Landing a helicopter on one of the helipads will repair and reload ammo for the helicopter. A helipad is a DIF (or any shape really) with a dynamic field 'dynamicFunction' set to a value of CreateLandingPad. A vehicle repair pad is a DIF (or any shape) with a dynamic field 'dynamicFunction' set to 'CreateFuelTrigger'. See the mission file for examples of this.

2. Modifying The zone

Modifying the terrain or textures for the zone can be done using the usual TGE world editor tools. Combat kit specific modifications are outlined below.

2.1. Adding a new spawn point

To add a new spawn point to an existing capture area:

- 1 Place a SpawnSphere in the world editor. Set the data block to SpawnSphereMarker or ParachuteMarker. Note that if you create a spawn sphere with SpawnSphereMarker that is too high above the ground, spawning players will not parachute in.
- 2 Add the new SpawnSphere to the simgroup for the capture area being modified.

2.2. Adding a new flag

To add a new flag to a capture area:

- 1 Place a flagpole shape in the mission area using the world editor. Make sure you get the flagpole under shapes/Flags in the world editor creator.
- 2 Name the flagpole object something unique. Unnamed or duplicate named flagpole objects will not work right.
- 3 Add a dynamic field named 'ObjectiveNumber' and give it a unique objective number

(currently 1-8 are used).

- 4 Add another dynamic field named 'ObjectiveName' this is the text that will display in game when the flag is captured.
- 5 Add a third dynamic field named 'dynamicFunction' set its value to 'AreaTriggerInit'. This sets up the flagpole trigger and TeamID when the mission loads.
- 6 Add the flagpole shape to the simgroup for the capture area being modified. Note that more than 1 flagpole in a given capture area is unsupported and may cause unpredictable results.

2.3. Adding a new capture area

To add a new capture area to the mission:

- 1 Create a new simgroup and give it a name.
- 2 Add a new flag using the instructions in section 2.2
- 3 Optionally add 1 or more spawn points using the instructions in section 2.1

2.4. Adding a new radar

To add a new radar to the mission:

- 1 Place a radar shape in the mission using the world editor. Make sure you get the radar under /shapes/AnimatedShapes.
- 2 Add a dynamic field named 'TeamID'. Set the value to 1 or 2. 1 for red team, 2 for blue team.
- 3 Only 1 radar per team is supported.

3. Creating a new zone

The BFT starter kit expects to see a red base and a blue base in all zones, so a red flag and a blue flag is required.

3.1. Red Base

To create a new red base:

- 1 Add a new capture area and a new flag using the instructions in section 2.2 and 2.3
- 2 Edit the flagpole and set its dynamicFunction to 'SetRedTeamOwner'
- 3 Add at least 1 spawn point using the instructions in section 2.1
- 4 Optionally add a radar using the instructions in section 2.4
- 5 Optionally place some buildings and vehicles in the area for players to use.

3.2. Blue Base

To create a new blue base:

- 1 Add a new capture area and a new flag using the instructions in section 2.2 and 2.3
- 2 Edit the flagpole and set its dynamicFunction to 'SetBlueTeamOwner'
- 3 Add at least 1 spawn point using the instructions in section 2.1
- 4 Optionally add a radar using the instructions in section 2.4
- 5 Optionally place some buildings and vehicles in the area for players to use.

4. Options

Not everything that is possible is required. Listed below are some optional features that

can be used to enhance Gameplay. Ammo and health crates can be placed in the mission, the capture trigger for flags can be adjusted, and spawn points can be set so they cannot be captured.

4.1. Flags

When creating flags, an offset value can be set to move the trigger position. Add any of the values below as a dynamic field in the mission editor to change the relative offset of a flag's trigger. See the flags on the bridges for an example of how this works.

6 xOffset – left/right offset. The default is 0.

7 yOffset – forward/backward offset. The default is 0.

8 zOffset – up/down offset. The default is -10.

4.2. Ammo and health crates

Crates to reload ammo or heal injured players can be placed as a part of the mission and associated with a given capture area. If the crates are placed in the same simgroup as a flag, they will change teams when the capture area changes teams. To use this functionality place an AmmoDrop item or a MedicDrop item in the mission. Both are found under /shapes/Weapon in the mission editor. Then move the item into the same simgroup as one of the flags.

Note that any crates placed outside a simgroup will not work properly.

4.3. Command panel

The current mission has a command center (panel) in each base. Destroying the panel gives 300 points and that team wins the game.

To place a panel in the mission:

1. place a 'Panel' object in the mission. Make sure you get the panel under /shapes/AnimatedShapes.
2. Add a dynamic field named 'TeamID'
3. Set it to a value of 1 for red or 2 for the blue team.

4.4. Doors

Doors can also be placed in buildings in the mission. Doors can belong to red team, blue team or neither team (neutral). When a player stands close enough to a door, it will slide open. A red team door will only open for red team members, and a blue team door will only open for blue team members. Neutral doors will open for anybody. The doors are electrified and will damage opposing players if they get too close. Players can also destroy doors of the opposite team.

To place a door in the mission:

1. place a 'door' object in the mission – found under /shapes/Doors/
2. Add a dynamic field: "DynamicFunction"
3. set the field value to: MakeRedDoorTrigger, or MakeBlueDoorTrigger or MakeNeutralDoorTrigger depending on the purpose of the door.

4.5. Static spawn points

It is possible to add spawn points to the mission that cannot be captured. This is normally used to prevent shutouts during game play. To setup a non-capturable spawn point:

- 9 Add a SpawnSphere object to the mission.
- 10 Set its datablock to 'SpawnSphereMarker'
- 11 Create a dynamic field and call it 'dynamicFunction'
- 12 Set the value of this field to SetRedSpawnOwner for the red team or SetBlueSpawnOwner for the blue team.

At mission startup the new spawn point will be flagged as owned by its named team.

5. Prefabs

CSK Also contains a number of prefab options in the mission editor to make level building a little easier. Once a prefab is placed in the mission, all the generated parts behave just like any other object that would usually be placed in the mission editor. The mission designer is free to edit the resulting objects to suit the mission.

The prefabs are found under Mission Objects\prefabs. Currently 10 prefabs are defined. The code for these prefabs can be found in example\creator\objectBuilderGui.gui starting around line 697. The key function is named "SetupPrefab"

5.1. Red Base

This prefab creates a very basic red base. 1 flag pole with 2 ground spawn points and 1 parachute spawn point. All of which are in a SimGroup named 'RedBase'. The flag object is setup with the proper dynamic function and objective number.

5.2. Blue Base

This prefab creates a very basic blue base. 1 flag pole with 2 ground spawn points and 1 parachute spawn point. All of which are in a SimGroup named 'BlueBase'. The flag object is setup with the proper dynamic function and objective number.

5.3. Capture Area

This prefab creates a capture area as outlined above in section 2.3. Just fill in the objective name, the objective number. The object name is not used. 1 flag pole with 2 ground spawn points and 1 parachute spawn point is created

5.4. Capture Area with tower

This prefab creates a capture area as outlined above in section 2.3, but with a watchtower dif placed next to the flag pole. Just fill in the objective name, the objective number. The object name is not used. 1 flag pole is created with 2 ground spawn points, 1 parachute spawn point and a watchtower dif object. All of these are placed in a simgroup named after the objective name.

5.5. DTS Fence

This prefab creates a row of DTS object all inside a single simgroup. You will need to choose the dts filename, and fill in the other values. The most important value for a true 'Fence' is the object spacing. This controls how far apart each object is placed. The direction value is a 3 value vector that controls how far to offset each dts object in the x and y directions. The Z direction is not used as each dts object is placed at

terrain level as it is added to the world. There is also a rotation parameter that will be set for each dts object as it is placed in the world.

5.6. Landing Pad

This prefab creates a landing pad with a helicopter on it. The landing pad dif object is setup to be a helicopter repair pad.

5.7. Landing Pad with Lights

This prefab is identical to the regular landing pad except landing lights are added to each corner of the landing pad.

5.8. Empty Landing pad with Lights

This prefab is identical to the landing pad with lights, except there is no helicopter added.

5.9. Transport Landing pad

This prefab is identical to the normal landing pad except a transport helicopter is used.

5.10. Transport landing pad with lights

This prefab is identical to the landing pad with lights except that a transport helicopter is used.