SPACE COMBAT KIT 2.0

Contents

Getting Started

Add The Player

The first thing to do to create a scene with the Space Combat Kit is to add the player.

To add the player:

1. Drag the provided Player prefab into the scene.

OR, to create your own player:

1. Create a new gameobject and call it 'Player' (or whatever else you like).
2. Add a GameAgent component.
3. Fill out the details in the inspector.

Add a Vehicle

A vehicle is represented by a gameobject with a Vehicle component attached to its root transform.

Add a vehicle to the scene:

1. **To add a space fighter**, drag the SpaceFighterFriendly prefab into the scene.
2. **To add a first person character**, drag the FirstPersonCharacter prefab into the scene.
3. **To add a capital ship**, drag the Battleship prefab into the scene.

OR, to create your own vehicle:

1. Add a new gameobject (e.g. your ship model) to the scene.
2. Add a Vehicle component to the root transform.

Note: in the following sections we'll be building the rest of the vehicle.

Enter The Player Vehicle

To make the player enter a vehicle when the scene starts, drag the vehicle (which must be present in the scene, cannot be a prefab) into the *Starting Vehicle* field in the inspector of the player's **GameAgent** component.

If we run the game, the player has entered the vehicle, but there is still no vehicle control or cameras so nothing will happen yet.

Control The Vehicle

If you've been using the prefabs provided in the kit, you can skip this section, since the vehicle input scripts have already been set up as part of the Player prefab, and the vehicle prefabs already have their vehicler controllers set up.

However, to set up a new vehicle, here are the steps.

Add a Vehicle Controller

Spaceship controller:

To add a vehicle controller for any kind of spaceship, including a Space Fighter or Capital Ship, add a **VehicleEngines3D** component to the root transform of the vehicle.

First person character controller:

To add a vehicle controller for a first person character, add a **FirstPersonCharacterController** component to the root transform of the character.

If we run the game now, still nothing will happen as we have not yet added a control script to send the player input to the vehicle controller.

Add a Vehicle Control Script

Important: Input scripts that are children of a game agent will automatically be started (if the vehicle is compatible) when the game agent enters a new vehicle.

To control a space fighter:

1. Add a new gameobject as a child of the player (the object with the GameAgent component).
2. Add a **PlayerSpaceVehicleFlightControls** component to the new gameobject.
3. Customize the controls in the inspector.

To control a capital ship:

1. Add a new gameobject as a child of the player (the object with the GameAgent component).
2. Add a **PlayerCapitalShipFlightControls** component to the new gameobject.
3. Customize the controls in the inspector.

To control a first person character:

4. Add a new gameobject as a child of the player (the object with the GameAgent component).

1. Add a **PlayerFirstPersonCharacterInput** component to the new gameobject.

2. Customize the controls in the inspector.

If we run the game, we can now fly the ship or walk the first person character around! However the camera will not be following it yet, so that's what we'll add now.

Set Up The Vehicle Camera

Add a Vehicle Camera

To add a vehicle camera to the scene:

1. Drag the provided *VehicleCamera* prefab into the scene.

OR, to create your own vehicle camera:

2. Add a **VehicleCamera** component to your camera.

To make the vehicle camera focus on a vehicle when the scene starts, open the inspector of the VehicleCamera component, and drag your vehicle (spaceship, character etc) into the *Starting Target Vehicle* field.

The camera won't yet follow the vehicle, because we haven't added a camera controller yet.

Add a Vehicle Camera Controller

When a new target vehicle is set on the Vehicle Camera, it will look for a camera controller among its child gameobjects that has a *Vehicle Class* matching that of the new target vehicle, and start one if it is found.

Note that the provided Vehicle Camera prefab already has camera controllers attached for all the types of vehicles in the kit, so you can skip this step unless you are setting up a new one.

To add a vehicle camera controller:

1. Add a new gameobject as a child of the vehicle camera.

2. For a space fighter, add a **SpaceFighterCameraController** component.

3. For a first person character, add a **FirstPersonCharacterCameraController** component.

4. For a capital ship, add a **CapitalShipCameraController** component.

Add Camera Views

Note that the vehicle prefabs provided in this kit already have camera views set up, so unless you are setting up a new vehicle or need a different type of camera view, you can skip this.

Add a camera view to your vehicle:

1. Add a new game object as a child of the vehicle.

2. Position and rotate this game object according to where you would like the camera to be.

3. Add a **CameraViewTarget** component and customize the values in the inspector.

Set Up The Weapons

Add Weapons To The Vehicle

Note that the vehicle prefabs provided in the kit already have weapons set up, so if you are using those prefabs you can skip this.

Follow these steps to add weapons for a new vehicle.

To enable weapons or any other triggerables to be fired, add a **TriggerablesManager** component to the root transform of the vehicle.

Weapon modules can be added via module mounts:

To add weapons to a vehicle:

1. Add a new game object as a child of the vehicle and position it where you would like the weapon to be.
2. Add a Module Mount component.
3. Add a weapon module prefab (such as the provided EnergyProjectileWeapon prefab) to the *Default Module Prefabs* list. The prefabs in this list will be created and loaded at runtime.

OR, to add the weapon directly:

1. Drag the weapon prefab into the scene to create a new instance.
2. Parent it to the Module Mount.
3. Make sure the *Load Existing Child Modules* checkbox is checked in the inspector of the parent's **ModuleMount** component.

Add Weapons Input

To fire the weapons on a vehicle:

1. Add a new gameobject as a child of the Player.
2. Add a **PlayerTriggerablesInput** component to it.
3. Customize the controls in the inspector.

Now if you run the game, you can fire the vehicle's weapons!

Set Up The Radar

Add a Trackables Scene Manager

Add a TrackablesSceneManager component anywhere in the scene.

This component is necessary to enable efficient tracking of targets in the scene, and is required for any target tracking to occur.

Add a Tracker To The Vehicle

To add target tracking capability to a vehicle.

1. Add a Tracker component to the root transform of the vehicle.
2. Customise the tracking parameters in the inspector.

That's all that is necessary to begin tracking targets in the scene. The targets tracked by the vehicle are stored in the *Targets* list on the Tracker component.

However, the targets are not yet visualized on the HUD (which includes target boxes, 3D holographic radar, and target hologram). That will be set up in the next section.

First, let's enable the player or AI to select targets from the Targets list.

Add Target Selectors To The Vehicle

To select a target:

1. Add a **TrackerTargetSelector** component anywhere on the vehicle.
2. Drag the **Tracker** component you previously added into the *Tracker* field in the inspector.
3. Customize the selection parameters in the inspector.

The selected target can be accessed via code in the *SelectedTarget* property of the TrackerTargetSelector component.

Multiple target selectors can be added to a vehicle as needed.

Link Weapons To Selected Target

To link weapons on a vehicle to the selected target:

1. Open the inspector of the Weapons component on the root transform of the vehicle.
2. Drag the TrackerTargetSelector component you just added into the *Weapons Target Selector* field.

This enables the guns to perform lead target calculation on the selected target, and the missiles to lock onto the selected target. Only one target selector can be linked to the weapons at any time.

Set Up Health

Making a Vehicle Damageble

To make a vehicle damageable:

1. Add one or more **Damageable** components anywhere in the hierarchy.
2. Customize the values in the inspector.

Receiving Damage

To receive damage, for every collider gameobject that represents something damageable:

1. Add a **DamageReceiver** component.
2. Open the inspector and drag the **Damageable** component into the *Damageable* field.

Now, when a hit occurs on one of the colliders, the DamageReceiver component on that collider's gameobject will send damage information to its assigned Damageable component.

Receiving Collision Damage

To receive collision damage on a damageable object (such as a vehicle):

1. Add a **Health** component to the root transform of the damageable object.
2. Make sure the damageable object has a Rigidbody on the root transform.

When a collision occurs, the **Health** component will receive it in its OnCollisionEnter function, and pass the damage to the **DamageReceiver** on the collider involved, which then passes it to the **Damageable**.

Causing Damage

To cause damage to a damageable object through one of its colliders:

1. Get the **DamageReceiver** component that is on the collider.
2. Call its *Damage* function with the necessary damage parameters.

This kit provides a **DamageEmitter** component that serves as a customizable projectile, beam and area damage emitter for all kinds of weapons. It will be discussed in detail in later sections.

Destruction of Damageable Objects

To enable a vehicle to be destroyed:

1. Add the vehicle's Destroy function to the Damageable's OnDestroyed event.

To deactivate the damageable object's gameobject when it is destroyed:

1. Open the inspector of the Vehicle component on the root transform and add a new entry to the OnDestroyed event.
2. Drag the vehicle's root transform from the Hierarchy view into the object field of the new event entry.
3. Select GameObject -> SetActive from the function popup menu and leave the checkbox that appears unchecked.

You can add other functions to the event as desired. There is an **ExplosionGenerator** component for creating an explosion via event.

NOTE: for damageable objects that are not vehicles, simply use the events in the **Damageable** component to drive what happens when it is destroyed.

Set Up HUD Target Boxes

Before proceeding, make sure you have set up your radar by following the steps in the ‘Set Up The Radar’ section.

To enable any part of the HUD to work, add a HUDManager component to the root transform of the vehicle.

The first steps for setting up target boxes for a vehicle:

1. Create a new gameobject as a child of your vehicle.
2. Add a HUDTargetBoxes component (which will add canvas components automatically) and open the inspector.
3. Add the Tracker component you added in the ‘Set Up The Radar’ section to the Trackers list.
4. Add any TrackerTargetSelector components you added in the ‘Set Up The Radar’ section to the TargetSelectors list.
5. In the TargetBoxContainers list, add a new entry by adding 1 to to the Size.
6. Under the new entry you just added, set the Prefab to the target box you wish to display for your target. Use the TargetBoxGeneric prefab provided if you haven’t created your own yet.
7. Under the new entry you just added, add the type(s) of target(s) you wish to track to the TrackableTypes list.

Run the game. You will see a target box appear around your target (if it is on screen) and a directional arrow appear at the edge of the screen (if it is off screen).