

# Code Overview

---

The Unity development tools tightly integrate themselves into all facets of game development. For the best possible code navigation experience, we recommend downloading Unity 4.6 Beta 17 (<http://unity3d.com/unity/beta/4.6>).

Much of what has been implemented was created using Unity's editor via a GUI. For this reason, a lot of the source code for the project is machine generated, and not meant to be navigable. Similarly, Unity automatically provides and includes all of the library files needed to build a complex game. We have not yet needed to include any functionality that isn't provided by Unity, and so there are no additional libraries in the project.

Everything that humans are meant to understand can be found in the Assets folder. In this folder we have:

- Scene files – these were generated by Unity, and are incomprehensible when viewed in a text editor. The Unity Editor can open them to show each of the two scenes in the game: the Main Menu, and the Game.
- .meta files – these are yet more files created by and for the Unity Editor.
- Fonts Folder – a selection of fonts we've compiled for use in this project.
- Models Folder – a handful of 3D models for our ingame objects.
  - Materials Folder – A Unity Engine construct.
- Resources – a number of Unity Engine constructs. These are known as "prefabs", which are game objects we have designed to be reused.
- Scripts – all of our custom scripts.
- Textures – all of our texture files and images.

## Scripts

All scripts are written in C#, since it is the fastest language Unity supports for scripting.

### bulletScript

A very basic script at this point; it is attached to bullet object as they are spawned, and simply ensures that they despawn after a certain time, to prevent steady memory and CPU needs from growing over time.

### enemyScript

A very basic script at this point; it is attached to asteroid objects as they are spawned, and handles the destruction of the asteroid when it collides with a ship or a bullet. It also causes their models to rotate about themselves in space and initializes their velocity such that they move towards the player

### **GameScript**

The GameScript is the script primarily responsible for handling clientside game logic. When the game begins, it spawns the player's spaceship. It handles all player input, allowing them to move and shoot, and it provides a number of Remote Procedure Call functions for the server to call to enforce itself on them.

### **HostScript**

The HostScript is the script primarily responsible for handling serverside game logic. When the game begins, the computer that is also the server will spawn this script in tandem with its normal GameScript.

### **MainMenuScript**

The MainMenuScript contains all functionality required whilst the player is in the main menu. It contains functions called by various buttons in the main menu, and handles the creation and joining of network servers.

### **PlayerShipScript**

A very basic script at this point; it is attached to any player-controlled spaceships in the game and currently is responsible only for reporting a collision to the GameScript.