**Required scripts:**

**Player Manager:**

The player manager is used to store the default stats of the player controller. The player controller gets these stats on startup.

**Game Manager:**

The game manager is responsible for keeping track of the game as a whole. It will control all other managers and allow quick changes to things like lighting.

**Enemy Manager:**

The enemy manager is responsible for spawning enemies. It will be easy to set exactly how you want enemies to be spawned including location, number, and type. It also stores the default enemy stats which are received by each enemy when it is created.

**UI Manager:**

The UI manager will handle the state of the UI for both the menu and the HUD. It will handle events sent by other managers as well as the unity canvas input system.

**Game Event Listener:**

The game event listener class is a base class for any script that needs to handle events. It has a function called handle event that takes a game event enum.

**Player controller:**

The player controller will include FPS controls for movement and aiming. (likely a heavily modified version of the standard unity FPS controller). It also keeps track of the players current stats. (health)

**Cupcadrone controller / Doughnut controller:**

Controls the AI for both enemy types. Gets base stats from the enemy manager.

**Weapon controller:**

The weapon controller will store default weapon stats such as fire rate and ammo capacity and allow them to be changed quickly. It also allows the player to switch between weapons.

**Weapon script:**

The weapon script will be responsible for actually attacking the enemies. It will use raycasts from the camera to check if an enemy has been successfully hit and dispatch take damage events to enemies.

**Pickup script:**

The pickup script will have a pickup type enum and a float amount. When colliding with the player it will send the player an event with the type and the float amount before being destroyed.

**Coding Conventions:**

Java naming conventions.

New lines for each scope.

As many comments as possible.

We will use github for version control.

We will use visual studio 2015 for editing code.