**Target Game:**

Description – There are several demons chasing a player. The Player can score by killing demons.

To kill demons, he must shoot at them. The other way to score is through PowerUp portals. These portals increase score as well as multiplies the subsequent scores earned by killing demons. Another kind of Reward portal is the Size Up Portal. These should be used sparingly. They increase the player’s size. Some Demons fly way up in the sky and the only way to reach them is by growing in size through the collection of these rewards. But once sized up, player cannot size down.

To Shoot use – use Mouse Left Press

To Move: Forward /Backward – W/S; Right/Left – E/Q;

To Rotate: Back/Front – D/A. The Guns’s Nozzle can also be adjusted Up/Down – UPArrow/DownArrow.

**Build:**

The project contains a Build folder with an .exe file that can be used to play. There is no exit condition. When the player is killed by a Demon, the game ends.

**To Run and Test in Unity:**

The Project is build using Unity 2017.4. Any 2017 version would do. 2018 versions may give some issues.

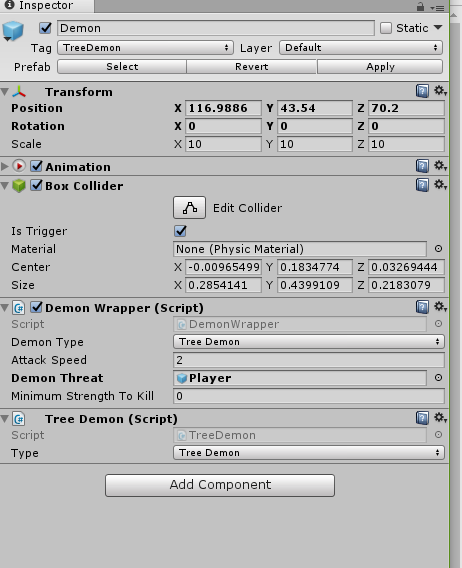
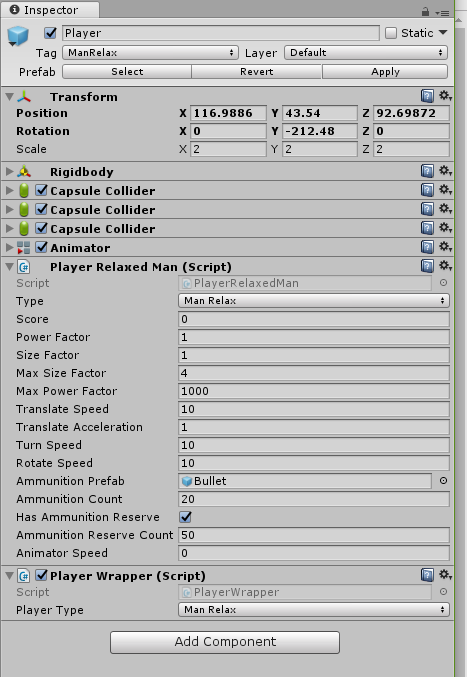
Few Free Assets have been used from the Asset Store.

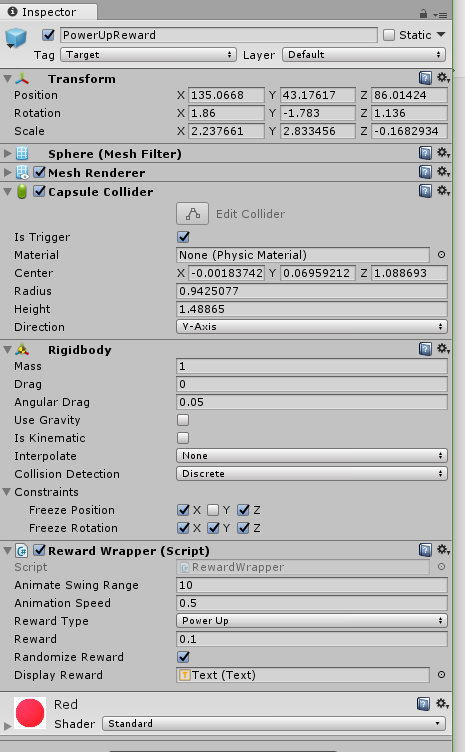
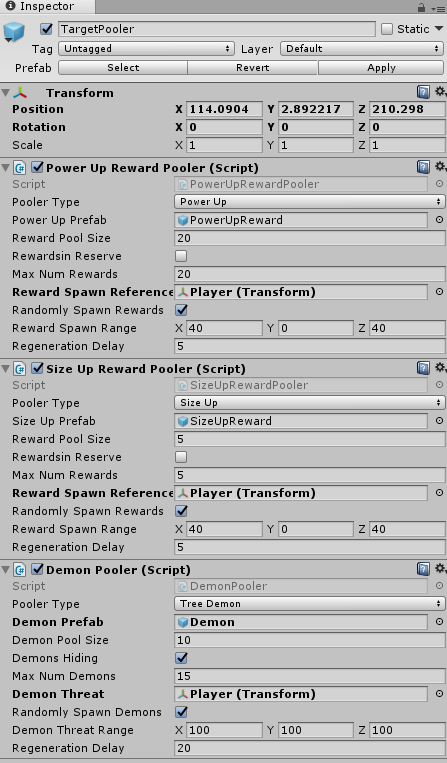
Inside the Asset folder, there is a Final folder.

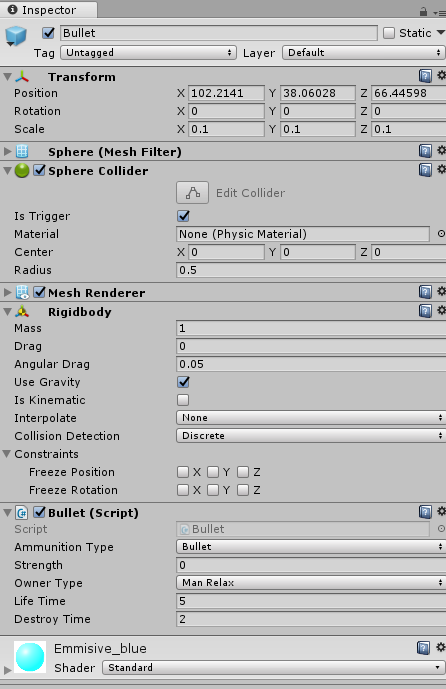
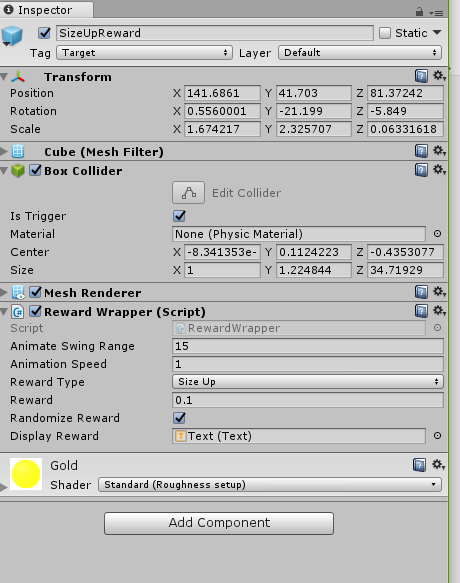
Final>Scene>TerrainScene.unity is the Game Scene.

All the relevant Models are in Final>Models folder.

The Inspector setting on the Models are shown in the images below.







**Major Components Description:**

Controller – Takes a Controller Type Script. In this case it is a DesktopControllerA.

TargetPooler – Creates and maintains pools of Rewards: PowerUp, SizeUp and Enemies(Demons).

For this PowerUpRewardPooler, SizeUpRewardPooler and DemonPooler scripts must be attached to it.

Player - takes two scripts – Player Wrapper and PlayerRelaxedMan. The later script can be changed for different types of players (Component Added/Removed). The Wrapper controls the runtime operations on any Player Type object.

Similarly, the Demon - also takes a DemonWrapper and TreeDemon scripts. The latter can be changed as per a particular demon. The wrapper operated on generic Demons Types.

The RewardPortals – SizeUp and PowerUp each only take a Rewards Wrapper that inherits from a RewardsPortal class. By design, all the reward portals are similar in behavior and their individual traits are exposed through the wrapper.

**Scripts:**

The scripts have been organized in the following folders:

**Interfaces:** In this folder, ArtifactTypes.cs contains different collections of Game Objects and their operational mapping. The different interface files implements logical groups of interfaces for different contracts between the classes and the functions they should perform.

**Abstract:** These folder classes contain abstract definition of classes such as Player, Demon, Ammunition, ShootDevice and ArtifactPooler.

**Implementation:** This folder contains the specific implementation of the game objects (artifacts).

The are a specific types like PlayerRelaxedMan, TreeDemon, bullet, Gun,RewardsPortal, PowerUpPooler,SizeUpPooler and DemonPooler. It also contains specific type implementations of Scoreboard and ControllerDesktopA.

**Helper:** This folder contains helper class ObjectPooler. It is a non Monobehavior type class. Pool Objects are created in different specific Poolers using this class.

It also contains castArtifacts.cs containing different Extension methods for casting between different artifacts and TerrainHelper.cs that has an Extension method to fetch the terrain reference while spawning and pooling different game objects.

**Wrapper:** This is the final folder that contains the wrapper classes which hide the internal implementation of Player, Demon and RewardPortals classes. The Rentime magic methods are implemented in these wrappers as Builder for these types of objects.