The Game Logic

The game consists of a rocket-like sphere object that automatically fires bullets.
The aim of the game is to destroy all the falling objects without letting any object pass the player.
The total number of falling objects that can pass the player before the **game ends is 15**.

Game Mechanism

As the time passes in the games there are multiple changes in the game.

- The falling object's speed increases over time till it reaches max speed.
- The pattern at which the object is falling changes back and forth between random and continuous.
- After a certain time period the player type changes to a different structure and firing pattern.

Power Ups

To aid the player there are certain power ups that help the player destroy more enemies. To collect the power up the player needs to destroy the falling power up object. They are denoted by different colours.

1. Kill Pass Enemies (Green Falling Object)
   This helps destroy passed enemies allowing the player to reduce passed enemies.

2. Slow Motion (Orange Falling Object)
   This reduces the speed of the falling object by half.

3. Triple Attack (Blue Falling Object)
   This increases the number of bullet spawn, based on the type of rocket currently in play.

Assets

For better gameplay certain assets have been added to the game.

Unity Assets Store Asset

1. msVFX Free Smoke Effects Pack
   - This asset is used for the hit VFX.

2. Retro Attack Sound lite
   - This asset is used for the hit SFX.

3. Boss Fight/Side Scroller Background
    - Used as backGround Music

Technical Documentation

Several new mechanisms have been added inorder to achieve optimization and create better gameplay.

Pool

Pool system is created in order to reuse already created gameobject. Thus an object is now not destroyed but is reused.

The pooling system uses Stack database to store not used gameobject and when required is either popped from the stack and if not present a new one is created.

Once the object's purpose is finished in the game the object is stored in the stack in its deactivated state.

Spawner

A spawner class is created to handle the movement of the spawner object and to change its property. It helps isolate the spawner mechanism away from the stage loop class.

Database

Databases for the prefabs and audio are created in order to store the prefabs and audio. The audioDB stores the list of audio along with other properties allowing multiple sound effects with single audio. Scriptable Objects are used inorder to create an in-game database.

Game Config

Game Config stores all the configs within the game. These values are accessed by the gameobjects. None of these values are modified in the class and are read-only properties for all the other classes.