Random Group 4--Jiapeng Zhao, [Martin Bruce Miller](https://canvas.eee.uci.edu/groups/46241/users/251521), ([Devon Gunn](https://canvas.eee.uci.edu/groups/46241/users/28091)?)

Prof. Edwin Ho

ICS 161

February 7 2019

**Space Invaders Explanation**

**Base requirements**

1. **Players**

* **Shape:** Player sprite is 300\*100 and is set as 100 pixels per unit. The bullet will be instantiated above the center of player’s position.
* **Movement:** The keys of ‘A’ or leftArrow, ‘D’ or rightArrow are used to move left or right. Player can only move between [-14.5, 14.5] which is half width of the screen – half width of player (**PlayerController**).
* **Shooting:** Player will shoot a projectiles bullet (controlled by **BulletController**) when space or left mouse button is pressed, isFiring in **PlayerController** script will be true to prevent next shoot until the bullet is destroyed.
* **Death:** Player dies when collides enemy bullet. Lives number, lives--, and gameover (**PlayerController** script)

1. **Enemies:**

* **Shape:** There are 11\*5 enemies with 1\*1 unit set by pixel per unit. Enemies are set equally spaced apart by some math calculations.
* **Movement:** The **EnemyUnionMovement** script controls the enemies move together, collide the borders and move down 1 unit and change move direction. The enemy will be destroyed when it is shot, so any alive enemy collides the borders will change the moving direction and position. The **EnemyController** set on each enemy will end the game when any enemy touches the lower border.
* **Shooting:** Enemy’s bullet is controlled by **EnemyBulletMovement** on the prefab of **EnemyBullet**. The **EnemyFiringController** (on the empty gameobject) “Enemy” sets the randomly shooting method, and randomly pick the bottom shooter of each column. Enemies are set as sub-gameobject in order of columns and rows, it is easy to randomly pick an exist column and find the last element in the column to shoot.
* **Death:** Enemy will be destroyed when it collides the bullet and the points will be calculated by the row of the enemy. (**EnemyController** script).

1. **Gameover**

* **EnemyController** script: enemy collides bottom border.

**PlayerController** script: lives == 0.

1. **UI**

* **Start** **Menu:** Start menu is set as the first scene. Buttons are in the canvas, and **MenuController** script provides the functions for onClick of buttons.
* **In Game:** The number of lives and its anchor are set in the top right of screen, and the score text and its anchor are set in the top left.
* **GameOver:** When game is over, load a new scene called “GameOver”, which has text of game over, two buttons. Two buttons are also use the functions from **MenuController** script.
* **Winning:** In the **EnemyUnionMovement** script, check if there is no child under the enemy manager which is an empty game object called “enemy”. When the number of child == 0, load a new scene called “You won” in the original version. For the extra credit, it will reload the level 1 scene, keep the score, add 1 life, and move down all enemy 1 unit.

1. **Extra Credit:**

* **Shield Bases:** There are 4 shields made by 10 small pieces, and each piece is added a **Shield** script. The script can handle the collision between any type of bullet and the small shield. Each time the small shield is shot, the hitCount++, and the sprite changes to the next one in the list. When hitCount is 4, the small piece will be destroyed.
* **UFO:** There is an empty game object “UFOmanager” with **UFOcontroller** script to randomly instantiate a UFO. Only 1 UFO will be shown on the screen because of the isFlying condition. The uncertainty is created by generating a number from 0-999 and check if the number == 0. For each second, the probability UFO will be instantiated is 6% based on a 60Hz computer. There random number 0-1 will control the spawn position of the UFO, and the movement direction will be changed based on the spawn position by **UFOmovement** script in the prefab of UFOenemy.
* **Difficulty Ramp-up and Level Advance:** In **EnemyUnionMovement** script, there is a timer to get the time of game play, and the speed will be enemyMovespeed = 1 + timer / 40 and fireInterval = 10f - 5f \* timer / 50f,which are good functions to increase the difficulty after a few testing. Also, in **EnemyUnionMovement** script, killing all enemies will reload the level 1 scene, keep the score, add 1 life, and move down all enemy 1 unit.
* **Pause Menu:** By **PauseMenu** script added in Canvas, game will pause when escape is pressed. The pause panel will also be active, and everything will be frozen by Time.timeScale = 0. Two buttons in Canvas will use the functions from **MenuController** script.