Due: 10/27 (11:59PM)

Requirements:

- Create a 2D game named *Space Shooter*. The game should meet the following requirements:
  - Create a new 2D project, then import accompanying assets `test1_assets.zip` into Unity.
  - See the accompanying video and implement this game. Your game should basically look and work like the one in the video.
  - The object of the game is to shoot down all incoming Asteroids before they hit you (or die).
  - Set the Aspect Ratio of the game as 5:4. Use `Jupiter.ttf` as your UI font.
  - As shown in the video, Asteroids (enemies) must be generated outside the screen, then coming in all possible random directions and always moving towards Player at the center.
  - Make sure each Asteroid self-destructs after it gets shot down or moves out of screen (so they won't clog up the memory).
  - Each Asteroid must animate through all of its 64 sprites (see video).
  - User presses LEFT and RIGHT keys to rotate Player counter-clockwise and clockwise, respectively. Note that Player's position never changes. It just rotates.
  - Player must rotate at a constant speed without ever getting sluggish or out of control.
  - User presses Space bar to shoot laser Bullets.
  - As shown in the video, each Bullet must move in the current *up vector* direction of Player (spaceship).
  - When a Bullet hits an Asteroid, the hit Asteroid must disappear with Explosion. Also, the current score must be updated on screen (see video).
  - Note that the Explosion itself must play sprite animation (see video).
  - If Player is hit by any Asteroid, Player is destroyed with Explosion. Then, display *Game Over* using *Jupiter* font (see video).
  - After Game Over, Asteroids must keep moving but never collide with each other (see video).
  - Add *Restart* button to let user restart once the game is over. (*Missing restart button will lead to point deduction*)
  - Make sure each object moves/rotates at adequate speed so the game would be enjoyable by average user.

- Finally, build your game as follows:
  - First, create *Build* folder under your Unity project directory.
  - Select *Windows* as your target platform (even if you're using Mac).
  - Click *Build* and select Build folder you just created.
  - The build file (.exe) must be created under Build folder. (*Missing build file will lead to point deduction*)

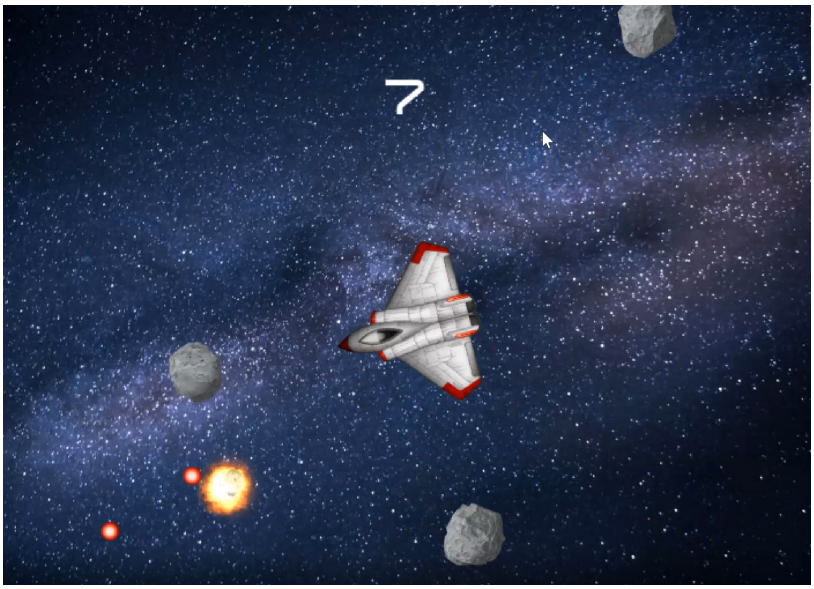

Figure 1: Space Shooter

How to access other object in the scene without using public variable:

```
// assuming Player is the tag name for the object of interest
private GameObject player = GameObject.FindWithTag("Player");
```

What to submit:

- Your entire Unity Project directory (containing all assets and build files) must be submitted.

How to submit:

- Use Canvas Assignment Submission menu to submit the assignment electronically at Canvas.
- Make sure to zip your entire Unity Project directory into `test1.zip`, then submit your `test1.zip` as a single file.

Policy

- At the top of each Script file (.cs), provide comments specifying the author, date, and a brief description of the file.
- Each Script file (.cs) must contain enough comments here and there to make it easy to follow your code. Insufficient comments could lead to point deduction.
- Incomplete project will get almost no credit (e.g., game does not run due to script errors or game terminates prematurely due to run-time errors).
- *Thou shall not covet thy neighbor's code*. If identical (or nearly identical) submissions are found among students, every student involved will get automatic zero for the assignment. The same goes for copying existing code from online source.
- If a student makes multiple submissions, only the last submission will be considered valid.