This is a combined project that will be in lieu of Assignment 2 (10%) & Design Project (20%), making it a total of at least 30% of the final grade.

You will develop a 2D-space shooter game based on the game engine framework developed in class. Given the time constraints we currently have, I have simplified the requirements. Additionally, I will accept the final submission by April 25, 2020 at 11:59 pm without any penalties, realizing that many of you may not be able to submit it by the official submission date on the last day of the semester (April 23) due to time crunch.

The game should be built on the tutorial videos posted on April 1, 2021and you will get further assistance via videos posted later in Weeks 11 and Weeks 12.

To provide some relief during these times, I am giving you an option to complete this project either individually or in groups of up to 2. You have to submit your choice by self-enrolling in a group (under Management->Groups). The last date to enroll, is no later than Monday, April 6, 2021at 11:59 pm. After this date, I will assume that you are working solo.

**One-on-one development support sessions**

Additionally, I am hosting a one-on-one support session (dedicating a total of 4 hours, to compensate for the upcoming Good Friday holiday) to assist in the development of your game and/or for feedback. You can schedule a 15-minute slot in Weeks 11, 12 and 13 A sign up .ink will be separately uploaded.

------------------

**Project Requirements**

1. **Score & Lives System**: Grant a score to every enemy that the player spaceship kills. The game must start with X lives and each time the player is killed (due to a collision with an enemy or due to an attack by an enemy), you must reduce the lives. If the lives is down to 0, the game must be over.

**2. Enemy System:** Currently, the asteroids are the only 'enemies' that can be killed. Try to add at least two additional types of enemies in the game. Some ideas are as follows:

a. Asteroid having a different size (larger the size, the more points the player gets);

b. start with random asteroid sizes.. each time a missile/bullet hits the asteroid, the asteroid reduces by X% size. The asteroid gets destroyed when the size goes down to 0.

c. add an enemy spaceship: the spaceship would be able to fire and attack the player, just as the player is able to attack the enemy.

**3. Pickup System:** Add some random pickups to the game which would add value or enhance the gameplay. You must play a pickup sound when a pickup is picked. Some ideas:

a. **Shield pickup:** When this is picked up, you add a shield around the player (perhaps, represented by a different ship sprite) and for a duration of time (say 10 seconds) your player spaceship is shielded from any collisions/attacks by enemies.

b. **+1 Life:** when picked, this adds an additional life to the game.

c. **times X score:** When picked, this increases the score upon hitting and killing each enemy (either temporarily for X seconds, or may be for the rest of the game, depending upon your game mechanics).

**4. Game-end condition:** Your game must have a game-end condition (not necessarily a win/lose condition). One way is to end when there are no more enemies left to destroy (by comparing to the number of enemies you generated at the start of the game. )  You must then transition to the game over screen that will display the final score.

**5. UI Heads-up-Display (HUD)**: Display the score and lives on HUD (either via simple text or using sprites for showing lives).

**6. UI Screens**: Show a main menu to start the game. Once you press start game, you must be able to start the game. You must also show a game-over screen with the final score.

You are free to expand the game and implement additional features should you wish (such as a vertical scrolling system; more than 1 level, etc) and if time permits; but they will not be a part of the core requirements.

**7. (Optional): Random Level Generation**: Your level must be preferably generated randomly (currently, the asteroids are being generated randomly). You must extend this to pickups and other types of in-game items that you are generating.

**Presentation**

You will also submit a 10-15 minute presentation video which will explain the core features of your game, and a brief gameplay covering the various possible ways your game-play is designed for (game-end conditions, enemies, pickups etc), walk through the coding you have done to implement the features.

**Submission Requirements**

1. One-page write-up describing how to play your game; the core features of your game and your game-end conditions.

2. A playable build (zip/rar)

3. The source project with the solution (zip/rar).

4. A 10-15 minute presentation video.  You can use OBS studio (free) to record your screen. https://obsproject.com/

**Submit the above 4 files individually in the dropbox**

**Grade Break-down**

1. Scoring System: 10%

2. Enemy System: 25%

3. Pickup System: 20%

4. UI: 10%

5. Game-play features: 12.5%  - How the game functions; whether the game end conditions work; random level generation system;

6. Overall completion: 7.5% (Memory management, coding style and level of completion)

7.  Errors/Bugs/Issues: 10% : (If there are no errors/bugs in your game, you will get the entire 10%)

8. Presentation & Submission: 5% (Failure to follow submission instructions will result in a reduction of this grade).