Objectives:

The purpose of this assignment is to give you experience working with a commercial-grade physics engine to manage the physics of a game world. In this Quest, you will:

- ➔ model game objects using Box2D bodies of various types, shapes, and physical properties,
- ➔ numerically integrate the physics world, and
- ➔ marry the physics world with an SDL2 rendered world.

In addition, you will add one other important characteristic to the system:

- ➔ include dynamic object creation and destruction.

Summary

- ➔ The primary task in this assignment is to incorporate the Box2D physics into the graphical world developed in Assignment 3. You will need both dynamic and static objects as part of your game. To implement these changes, you must create a physics device which will be used by the body components to properly construct a physics body (using body definitions, shapes, and fixtures as described in the videos), and register these bodies with an appropriately constructed b2World.
  All bodies must be registered with the b2World via the CreateBody() method. Note, all length measures must be converted into Physics World units such that the physics engine stores them on the range [0.1,10] (see example programs in GitHub)
- ➔ You must implement the capability for objects to be dynamically added and/or removed from the game world. For this assignment, the instructor solved addition problem by modifying the Object's update method to return an Object*. The removal of an Object is implemented via a boolean member stored within the Object class (initially false). When the object becomes removable, this member is set to true. Within the main game loop, objects having this member equal to true are removed from the container of all game objects.

Impulses and Numerical Integration

The Physics World should be updated using a Δt of 1/100 second per frame. Linear and Angular friction can be implemented using the SetAngularDamping() and SetLinearDamping() methods provided for the b2Body class. Forces to be imparted on the body can be applied using the ApplyForce() method provided for the b2Body class or the SetLinearVelocity() method. (Note: any Box2D-based approach may be used for this assignment; if you can think of a better way, you are welcome to use it). Angular changes to objects can be achieved using the b2Body class's SetAngularVelocity() methods. You should not attempt to change the location or angle of the sprites yourself but should make adjustments to the physics world and read the position and angle from there in order to adjust the sprites.

Physics Device

A physics device should be created for your assignment to separate the physics engine from the game code. Look at the examples in the Signature Experience Game Engine code!

You may not need all of the methods in the physicsDevice for your solution. Note that Box2D has a void* userData field that can store any pointer. It is best to use this to store the object being controlled by the body being created. In this way, the body component can pass in it's owner and the FindBody(Object* object) method can find the corresponding Box2D body in the world.

Suggested Division of Labor

Together:

- ➔ Create Physics Device

Member 1:

- ➔ Adjust components to implement physics

Member 2:

- ➔ Dynamic creation/deletion

Submission

All code should be submitted to your repository using GitHub desktop to "Commit" and "Push" your code. The last push date will be considered your date of submission.

Before the last "late day" you must submit a video via the Assignment 4 submission link in Blackboard. You **MUST** use Kaltura to submit the video and should be less than 10 minutes long. Focus on the new content. No need to rehash old code!

!!!!!Videos submitted directly into blackboard, not using Kaltura, will not be accepted!!!!!! *We have limited space on Blackboard for our class!*

Both members of the class will need to submit a video walkthrough of the code. DO NOT read the same script!! I need to be able to tell you understand ALL the code, not just the portions you wrote!