**IN2026 Coursework**

**Part 1: Start Screen**

Objectives:

* Create screen with title and key prompt using GUILabels.

An alternative would be to create a “texture”/image that is on the display at first before transitioning to the game world.

Outline of changes:

* Two new labels in Asteroids.h/.cpp
* One new Key press implementation
* Set initial visibility of lives and scores labels to false.

Implementation:

Created two new GUILabels alongside the already existing ones in Asteroids.h.

shared\_ptr<GUILabel> mStartScreenTitle;

shared\_ptr<GUILabel> mStartScreenLabel;

Next, I created the GUI.

First, I set the visibility for the score and lives labels to false, this is to make sure they are not present on the start screen.

mScoreLabel->SetVisible(false);

mLivesLabel->SetVisible(false);

Then I made the GUI components for the start screen: the Title and Key prompt in Asteroids::CreateGUI.

Text

Description automatically generated

The visibility of these is automatically true and don’t need to be set to false as they should be present on the initial display.

Finally, I made it so to start the game, the player needs to press the enter key. I did this by expanding the Asteroids::OnKeyPressed method.

switch (key)

{

case ' ':

mSpaceship->Shoot();

break;

**case 13: // Enter**

**mStartScreenTitle->SetVisible(false);**

**mStartScreenLabel->SetVisible(false);**

**mLivesLabel->SetVisible(true);**

**mScoreLabel->SetVisible(true);**

**mGameWorld->AddObject(CreateSpaceship());**

**CreateAsteroids(10);**

default:

break;

}

The tricky part here for me was seeing what could/should be in Asteroids::OnKeyPressed and what should remain in Asteroids::Start.

In the end, the code for creating the spaceship and asteroids on screen were moved to the key pressed method – this is so that nothing besides the title and key prompt are on screen for the start.

The result looks like this:

Graphical user interface

Description automatically generated with medium confidence

**Part 2: Power-Up System**

Objectives:

* Implement a power-up (Extra Lives)
* Update collisions.

Outline of changes:

* One new Header file and accompanying cpp.
* Changes to collision throughout the game. (e.g., bullets)

Implementation:

First, I made LifeUp.h.

Text

Description automatically generated

Followed by LifeUp.cpp (Note: I did use tutorial 2 as a base for this)

Text

Description automatically generated

Next, I looked for changes to other object collisions to see how other objects should interact with items.

* ‘Asteroid’ did not need changes, as they will ignore collisions with items.
* ‘Spaceship’ also did not need changes, player colliding with items will be ignored.
* Bullet needed to be altered to allow item collisions. (Bullet::CollisionTest)

if (o->GetType() != GameObjectType("Asteroid") || o->GetType() != GameObjectType("LifeUp")) return false;

Next, I moved onto making changes in Asteroids.h and .cpp.

**GENERIC CHANGES**

* Added implementation for GLUT\_KEY\_DOWN for special key presses and release.

Text

Description automatically generated