Documentation

Sample #1 – Final Year Project:
The project shows basic level designs with different types of objects the player can interact with. The game also highlights different enemy behaviors. The game can be played on the desktop as well as on mobile devices.

Code Files and Resources:
This folder contains the following folders and their contents:

```
-libs
  |-CEV3-1-10-ALL.swc
  |-feathers.swc
  |-starling.swc
-src
  |-Game.as
  |-Game2.as
  |-GameEnd.as
  |-GameOver.as
  |-Main.as
  |-Menu.as
```

The game flow of the project is as follows:

```
Main => Menu => Game ===> Game2 => GameEnd
                  ↳> GameOver <↲
```

The function of the individual files in the project are given below, in the order as listed in the directory:
Libraries:
CEV3-1-10-ALL.swc – This is the library that holds the Citrus Engine. Citrus is used to add physics and create the platforms, hero, enemies and other elements that are visible in the game. It is required to import the citrus package.
feathers.swc – The feathers library is used to add UI components. The game utilizes it to import files under the feathers package.
starling.swc – Starling is the framework used to create the game and with it we import the starling package.

ActionScript files:
Game.as – This file contains the code for the 1st level. At the beginning of the class, in the initialize function, the AssetManager class is used to load the assets from the bin folder. This contains the image files with can be attached to the various game objects. After this the game world is created. The first thing that needs to be added is the physics. Then the objects that make the level are added, i.e. is the platforms, moving platforms and crate. We then add the hero, which is the player's character, enemies, coins and the camera. I had initially created a camera variable to test its various parameters. After figuring out the ideal values for the game, I simply set up the camera and assign it to the hero character. Functions are attached to the coins which are called when the hero begins contact with them, i.e. when the hero collects them. After this, the on screen buttons are created for movement and added functionality. There is a flag associated with each button which is set if you begin pressing it and is unset when you stop. This is used to check if a button is being pressed as it may alter the behavior other functions, for e.g. if the _blockFlag is set, the hero cannot move or jump. The first button is the jump button. It checks if the hero is on the ground and is not blocking, only then the hero can jump. The behavior of an enemy is also attached to this button. It checks if the hero is within range of the enemy, if so there is a 61% chance the enemy will jump and block the hero's path if the hero tries to jump over it. The next button added is the block button. This simply sets a flag which,

as seen in the heroHurt function, prevents the hero from taking any damage but at the same time, the hero is also unable to move. The final 2 buttons are the left and right movement buttons. They set their respective flags which are used in the heroMechanics function, and also are used to move the hero after it has jumped, since the aforementioned function only deals with movement when the hero is on the ground. The update function is called once per frame. The first 3 conditions are used to affect the game behavior. If the hero falls off the ground platform, out of the level, it is considered as an instant kill, but the player gets a chance to retry the level. If the hero collects coins, it updates the count, and if the player collects the required number of coins, they proceed to the next level. If the hero is hurt 5 times, then the player loses the game and is sent to the Game Over screen. The next condition deals with behavior of another enemy. If the hero enters the range of the enemy, the enemy will start following the hero until the hero exits its range. Finally, the update function makes a call to the heroMechanics function. The first part of this function deals with the hero's movement. If the hero has not jumped and isn't blocking, the hero moves either left or right if the respective buttons are pressed. The next condition determines the hero behavior when it is blocking. If the hero is blocking, as it takes no damage, it remains stationary, otherwise, it is thrown back and takes damage.
Game2.as – This file has the code for the 2nd level of the game. The basic mechanics of the level is similar to that of the previous level with the same objective to collect all the coins the finish the level. The level has a bigger design with multiple levels to climb and collect the coins. We take a look at the update algorithm which showcases 3 more enemy behavior algorithms. The first enemy jumps after random intervals of time under 70 frames. The second enemy teleports right next to you when you come into its range and then follows you until you exit its range. The final enemy will follow you once you enter its range and when you try and exit it for the first time, it will teleport right next to you.
GameEnd.as – This class is called when the player finishes the game. It notifies the player that the game is completed.
GameOver.as – When the hero takes too much damage and dies, this class is called and displays "Game Over".
Main.as – The Main class is the first class that is called when the game starts. We first set multiTouchEnabled to true and then we call the first scene of the game.
Menu.as – This is the first scene which starts the game. It was used to test the working of buttons which are used to control the hero character.

Output Images:
This folder contains the image files showcasing the output the code file generates and contains the following files and its contents:

```
-Level 1
  |-Image 1.png
  |-Image 2.png
  |-Image 3.png
  |-Image 4.png
  |-Image 5.png
  |-Image 6.png
-Level 2
  |-Image 1.png
  |-Image 2.png
  |-Image 3.png
  |-Image 4.png
```

The above images showcase the design of the 2 levels and the enemies, coins, various types of platforms and crates that are present in them. They also show the hero character and the multiple buttons used to control it.

User Guide

When the player starts the game, the Main class opens the Menu scene. The Menu scene has a button that starts the game.
The game starts at the first level and displays the 4 buttons, hero character, enemies and the elements of the level. The hero has to collect the first 2 coins to reach the next level and either kill or cross the enemies in its path. Alternatively, the player can simply bypass the first 2 coins and go to the end of the level and collect the third coin after crossing a chasm using a moving platform, with which the player will proceed to the next level. If you get hit 5 times while not blocking, you die and reach the game over screen. There are 2 enemies in this level, the first enemy has a 61% of jumping and blocking your path if you try and jump over it and the second follows you around if you enter its range.
The second level is similar to the first but has a larger design. Here the player has to collect 10 coins in order to finish the game. The coins are located on platforms and the player has to navigate upwards to collect them. Also, the enemies in this level are slightly more advanced than the first level. The first enemy jumps and blocks your path after random intervals of time, the second one teleports near you and then follows you if you enter its range and the final enemy will follow you if you enter its range and the first time you try exiting it will teleport in front of you. After you have collected all the coins, you reach the end of the game. As with the first level, if you are not blocking and get hit 5 times, reach the game over screen.