**Iteration Plan**

**High Level Objectives**

***Deployment:***

1. To deploy the game, you must go to <https://github.com/cncgen18/CSC440_RedSnapper> to download the zip file.
2. After getting it, you must also install gamemaker to open the files here: <https://www.yoyogames.com/get>
3. You will need to sign up for gamemaker, or if you have a steam account you can also grab the free version on steam for free.
4. Afterwards, open gamemaker and open the file debauchery.gmx. Then hit the “run the game” button. It should open the game, and you should be able to play around with it.

***Implementation:***

The four use cases we picked for this iteration is: dash, checkpoint, pause menu, and title screen. For each of these we had criteria of what we wanted to do. Most of what we did in the last iteration was fixed, and upgraded. Such as the pause menu in pause. While most of this iteration was focused on how to make do menu related use cases, the next will be gameplay related use cases. We also had to take off the gameover menu for now, as it was causing some bugs to arise for the checkpoint object.

**DASH:**

For dash, we wanted to make it timed, so that you couldn’t just continue holding one button constantly for the whole level. This was important, because it felt like the player could easily be go through the level.

**CHECKPOINT:**

One of the most important gaming concepts. The checkpoint makes sure if the player object collides with the checkpoint object, the checkpoint object changes the global checkpoint variables to that location. If the player dies, he or she will respawn at the checkpoint they collided against earlier.

**PAUSE MENU:**

Pause menu is just an addition to pause. With the menu, we added features of restarting and going back to the title screen. We wanted to make sure we added these features, as we didn’t want the player to have to close the game to be able to go to the title screen to be able to quit. With the restart feature, we wanted to make sure the player would be able to restart at an earlier checkpoint. This will also be useful when we go into bug testing in stages.

**TITLE SCREEN:**

The title screen is simple enough. You press enter to start the game, or you press escape to close the game. We wanted to add some features, such as the scoreboard, but that would need to wait until the next iteration.

**List of work items with assignments**

1. ***Demo:***
   1. *Creation of objects*
      1. Player
         1. Added dash key
         2. Added time limit for dash
      2. Pause
         1. Attached new sprite to the room
         2. Created key functions that allow you to go back to the title screen or restart level
         3. Fixed the problem with the last iteration where the color would stay changed despite unpausing
      3. Logo
         1. Attach to the title screen.
      4. Title Screen
         1. Add key functions that allow you to start the game, or exit the game by pressing enter and escape respectfully.
         2. Added the words “press enter” to the title room.
      5. Checkpoint
         1. Created objects
         2. Created the global variables to keep track of which checkpoint to respawn from.
   2. *Sprites*
      1. Checkpoint
         1. Created a rudimentary block
      2. Logo
         1. Created a logo to put up at the title screen
      3. Pause
         1. Created a sprite to be added to the pause screen.
   3. *Rooms*
      1. Init
         1. Added the code to initialize the global checkpoint variables
         2. Added the pause object
         3. Made sure to put the init room before all others for the checkpoint to be initialized.
      2. Title
         1. Created an empty room for the logo and “enter key” sprite.
2. ***Vision:*** 
   1. *Use model changed*
3. ***Use Cases:***
   1. *Use model changed*
   2. *Fully Dressed Used Case*
      1. Move
      2. Jump
      3. Gameover
      4. Pause
      5. Dash
      6. Checkpoint
      7. Pause Menu
      8. Throw
      9. Lockpick
      10. Pickup
   3. *Brief Description*
      1. Inventory
      2. Stats Screen
      3. Ceiling Trap
      4. Save
4. ***System Sequence Diagrams:***
   1. *System Sequence Diagrams*
      1. Dash + alt
      2. Checkpoint + alt
      3. Pause Menu+ alt
      4. Title Screen + alt
5. ***Iteration Plan:***
   1. *Deployment Instructions*
   2. *Implementation*
   3. *Evaluation Criteria*
6. ***Design Document:***
   1. *System Sequence Diagram*
      1. Dash
      2. Checkpoint
      3. Pause Menu
      4. Title Screen

**Evaluation criteria:**

1. ***Dash (90%)*** – Dash does work. The implementation we thought of did succeed. There was of course a bug that was found too late into this iteration. We wanted to make sure that you couldn’t just hold down a button and dash constantly. While we succeeded in doing that, we found another problem, which is the fact you can continue to dash as long as you press the button after the time limit is up. We do not want this, and for the next iteration, we hope to fix it.
2. ***Checkpoint (95%)*** – Checkpoint is pretty much complete. It works like we imagined, but we do want to add a better sprite, and we do want to make sure that a checkpoint cannot be used again, once it’s been passed. We believe those should be a bit easy to work with.
3. ***Pause Menu (85%)*** – While we were able to complete the pause menu and able to put the sprite for the pause menu in the middle, we discovered that the pause menu is in the middle of the stage. We understand to fix this, we need to implement a dynamic camera, so rather than putting it in the middle of the stage, it will be in the middle of the player’s view. Other than that though, the pause menu works as describes.
4. ***Title Screen (95%)*** – The title screen was easily implemented. The only thing we wish to add is a score screen to be able to view your score from the cloud.