Process Book
Jason Sullender for Alma
DAGD 355, Ferris State University
December,2015

This project was used for educational purposes on creating a video game in Unity engine writing with C#. This project was a 2D platformer game meant for an audience who enjoys this genre of games. The audience that would play this is one that also likes pixel art styled games. A company that would be interested in a game like this would be one that has or would like to start in a pixel art styled game that is also a platformer genre.
To start this game a game design document was made to help decide the processes and art style. The game design document also showed how the game controls and levels were going to be laid out. The game design document also was there so that me as a developer of the game could prioritize the the features that would go into the game. Since this game was made for educational purposes the engine and programming language were already planned out to be Unity writing in C#. Once there was something that was learned in class it was to be implemented accordingly in the game. By doing this there was weekly components of the game that were added. For me personally as the game went on the game design document changed as in the assets that were to be put into the game. From the start the assets, such as art, were temporary to make sure that what was written in code was working correctly. These temporary assets were never to be used in the final deliverable. Once the game was close to completion programming wise was when I used the online application piskelapp.com to create the pixel art that was inspired by that art that was in the game design document. Communication of what was done was handled by having milestones in the class to present the updated design document and the progress of the game. Work was done by myself except for one assignment that had the class look up other repositories so that we could work on another persons game for that Jacob Drummond worked on my game by creating the helmet, shield, helmet, and chest assets that are used in the final deliverable of the game. At the beginning of the semester we chose what game we were to work on that is why my game is a 2D platformer. To evaluate the work that was done was that during the presentations the other students would critique along with the professor. The professor would also give us a grade on what work was shown.
To research what work would be needed to be done I played games that were in the same genre to see what other developers did to make their game. I also knew that there would be problems that I would run into such as going down ramps along with the different environment and NPC obstacles that the player must overcome. For these I looked at articles on the internet such as Nintendo for what makes good player obstacles. The main inspiration that I took from to make my game was a game that is available on Xbox called Volgarr the Viking. This game was major inspiration for me because it did pixel art in a 2D platformer very well as well as being fun yet very challenging which is what I wanted to do with my game. To make some of the levels I had a few friends help me sketch some of the design for them that would be fun for them to play. I also found some articles explaining the problems that would come up and try to follow the logic so that the problem wouldn't occur in my game.

To meet project goals there was a rubric to follow so that was one way of making sure that what needed to be in the game was incorporated into the game. There was also the restrictions of the genre itself. Being a 2D platformer there are certain things from that genre that are expected to be in the game such as having a character that you control to jump across different platforms to get to a certain end goal. These things were made by making different prefabs that had collision detection to check where the players and enemies are in reference to

the ground. To meet certain criteria of the game I made a player controlled by the user, two different enemies, multiple types of platforms, and ways that the environment itself can kill the player character. There is also a start and a finish to every level to move onto another level but there are multiple ways to get there. There was a lot of code that was attached to the different objects to make sure that everything worked the way I wanted it to to make the game feel like the game that I wanted to play. The final deliverable of the game had the character but only the Minotaur and Gorgon enemy assets the harpy and bosses were not able to make it into the final deliverable of the game this was due to time. Parallaxing background made it into the game as well. Something that did not make it into the game because of time restrictions was when the boss died the player character was supposed to get the head of the boss and get the power of that enemy character. For example when the Minotaur boss died the player would get a minotaur head and be able to charge at enemy characters. This was one feature that I wish I had time to put into the final game. Something else that didn't make it into the game was endless waves of enemies but I actually like that I didn't put this into the game because it would have changed the feel of the game into something I wouldn't have enjoyed.

Something that was a major problem was that in the making of the levels was that the crumble cube prefab was not doing the code that was attached to it. To help this I did level making a specific way by saving a past level and using that levels crumble cube. Another problem that I encountered was that the transfer of code from the current character controller that I had programmed I wanted to go onto a character controller but with all of the code that was already written it was going to take some time to figure out the same type of syntax for the different controller which I didn't have time to do. So for next time I would use character controller instead of the rigid body. Something else that I didn't have time to do but I wanted to was to refactor the code so that it read easier and was more efficient. Another problem that I encountered was that when I made an animation I had to make numbers to make sure that the animation would stop when the key was not pressed and start again when the key was pressed again. Other problems that occurred was when instantiating something I wasn't really sure where it was going to end up shooting at first so this took some trial and error to get right at first but once I did a few to get practice it came easier. After doing this project something that I would do differently is take time to plan where and what my classes are and what object that they will go on to make the most effective programming as well as the redundancies to make modules to clean the code better.

One of the most interesting things that I learned doing this project was raycasting. Raycasting can be used in many different ways to help your gameplay to run better. You can cast a ray by clicking on the screen to help move a character like in the game Diablo, or you can have a raycast out from the player like a first person shooter to shoot rays at enemies to hurt them. Or to cast rays out from enemy characters to check where things are in the environment. Raycasting is something that was hard for me to grasp at first and didn't really understand until the end of the project so raycasting is something that I would want to use more in the future. Another way that was important for an AI aspect was that you can make pathing and go pretty in depth in the editor with the pathing to have the NPC's move around the scene and detect objects in the scene easily. If I were to make a 3D game this is something that I would look at more closley to move the AI around. Something that was very important to learn at first was how

all of the code that was written was much like a class that was attached to any object. This took time getting used to be able to pass variable information from one object to another but with practice this became much easier. Two things that I learned that made the game feel more real were the animator and the audio source. The animator at first was very confusing and it took a lot of redoing on the animators to make sure that it was acturally doing the correct animation that I wanted in the animation window. Sound was a lot easier to put together but the different methods that are in the source class were strange so I always went with the source.PlayOneShot(variable,volume). Vectors were one of the more important things to learn since they are used to get the locations of objects. Something else that was interesting to learn was the passing of data that it was so easy to do that it only took DontDestroyOnLoad(gameObject). By doing this you keep all of the properties on that game object which helps to move on to another level. Also if you want to wait a certain amount of time before doing something a Coroutine is the thing to do because you can yeild and wait for a certain amount of seconds. For creating levels learning that Unity measurements are in meters was very helpful compared to how other engines do their units. Something that I did not use but did learn how to do was make a terrain which was quite easy it was much like how other engines do it my a click raises the terrain and the right click lowers the terrain but in Unity the terrain cannot go lower than the zero so after the terrain is raised is the only time it can be cut into. When playing games now I also see what I learned about user interfaces where diegetic is in the game world, Nondiegetic is outside of the game world, spatial is in the 3D space but not part of the game world, finally meta is in the game world but not in 3D space. Also when using procedural generation you can use noise that will help to get different terrain types in your game.

From open source of freesound.org

Swing Xxchr0nosxX march 27 2015
Swoosh WHOOSH air effects>>05244fireball whoosh, Robinhood76 September 11 2014
walking on pavement, JohsonbrandEditing August 7 2014
cave drips, EverythingSounds September 12 2013
THMAdventure_Sound>>hurt00[THM_Adventrue], alex_audio May 17 2013
Snake Pit, Andy19 October 21 2014
Bull/Monster>>hfoon, harleto October 1 2013
People>>HighheelsWendyQuick, acclivity November 30 2006
Horse>>Schnauf, Andune October 21 2008
soundsofthings>>woodbreaking, SoundCollectah November 24 2010

Music from open source dewolfe.co.uk
Find The Truth, Tim Whitelaw

For Shield, Helmet, Chest assets with permission
Jacob Drummond