Oratile Tlhabanyane
Student no. 1686967

Intent

I will give a summary of the intention that went behind the development of this prototype game. Based on the group's project plan, we each had to create a squirrel 2D exploration game in a platformer level design format. The game has a straightforward gameplay where a squirrel who progresses through one massive level, collecting objects, solving puzzles, and avoiding enemies and obstacles. The squirrel needs to have collected enough acorns and make it home before the forest starts getting dark. The aim is to create a game according to what the project plan states, however if during development some of the project plan goals were not met, the game may continue as is as long as there is justification for the certain changes.

The intention was to create a prototype as close to the way the project plan states. The player needs to control an avatar which is a squirrel named Mr Chonks. My Chonks is going to be collecting as many acorns as her can before his time runs out. The acorns are like coins, they are the currency of the game and the more acorns you collect, the better. The squirrel can use these acorns to win the main objective of the game which is to collect as many acorns as possible to bring back home. However, because of the obstacles Mr Chonks is going to face this simple objective will not be such an easy thing to accomplish as one would think.

The player (using the squirrel) is given the following encounters:

1. The player is going to encounter enemies which decrease his health.
2. Checkpoints where the player have an option to either buy health or resources.
3. And lastly, the player is also going to encounter Friendly Non-Player Characters that will offer a trade.

The prototype should include game features such as:

1. Animation
2. sound design and other basic UI features
3. A system that will control inventory of acorns, health potions and building resources.
4. A system that will control trade between the avatar and the NPCs.
5. Spatial puzzles

In the duration of the development of this prototype should take place within a 6-week period. Each week of the development process there will the a playtestubg session that will take place. Each of the playtestubg session will be focusing at testing an element of the prototype that was created tbat week and after each session necessary changes that need to take place will then be made and the game will be adjusted accordingly when fixing these elements. This is called the iteration process. The reason why we are creating a prototype is

to show what a full running game would like without having the pressures of creating a fully fledged game but have essential features to show the general game play.

Process

The first week, see Figure 1.1 in the project plan, it should start with the development of the character controller, basic character placeholder, basic layout, and spatial puzzle. I had first started with the character controller as it control the affect of the rest of game. To see that character movement is working in the controller we need to set up a basic level layout. I added a rectangle platform and a simple oval shaped object to act as the character. The movement to left, right and jump was fine.

TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)
First Iteration				
Character controller	5/25	5/26	1	1
Basic character placeholders	5/25	5/26	1	1
Basic level platform/layout	5/26	5/27	2	1
Spatial puzzle setup	5/27	5/28	3	1
Playtest iteration	5/28	5/29	4	1
Second Iteration				
Collection system implementation	5/30	5/31	6	1
Inventory system implementation	5/30	6/1	6	2
Checkpoint system implementation	5/31	6/2	7	2
Playtest iteration	6/2	6/4	9	2
Third Iteration				
NPC basic character creation	6/5	6/7	11	2
Enemy Patrols	6/5	6/7	11	2
Friendly Interactions	6/5	6/7	11	2
Playtest Iteration	6/8	6/10	14	2
Fourth Iteration				
Trade system implementation	6/11	6/13	17	2
Introduction of additional resources	6/12	6/13	18	1
Character abilities	6/13	6/14	19	1
Playtest iteration	6/14	6/16	20	2
Fifth Iteration				
Main Character Art and Animations	6/17	6/19	23	2
NPC Art and Animation	6/17	6/19	23	2
Level Design Assets	6/19	6/21	25	2
Playtest Iteration	6/21	6/22	27	1
Polish				
Polish	6/23	6/25	29	2

Figure 1.1 Screenshot of Schedule

I always struggle with making sure that the character is grounded before it could jump, however this step was easily understood this time around as I used a Boolean in the animation to check if the character is grounded. The next thing I did was the animation of the character. I used the sprite sheet method to create each animation which are idle, run and jump. I made a sprite sheet for each animation and added them to the scene of the game to create an animation with an animator. I chose this method because it was simple and fast, however it requires one to have a bit of an artistic hand for the drawing of the sprites.

The next thing I did was to create the platforms of the game. I used tile maps to create the platform. Not only is this method faster but it is accurate. The tilemap uses tilesets to create platforms. These tilesets, I used the paint option to paint my platform on the areas I wanted platforms. The game does not have mini levels, but it is just one big level that the player needs to accomplish so the level design just only needed to happen in one scene. I could not recall what character placeholders were and I had planned to do the spatial puzzle once I have done the basic game play. Thus, I went into testing the elements I have implemented which are the character controller(movement), character animation and basic layout design of the game.

The following week the project plan state that we do a collection system, inventory system and a checkpoint system. This week came with a lot of push backs and the risks we accounted for in the project plan took place very hashly during that week of my development. Load shedding, and a lot of assignments due. Thus, a proper inventory system was not able to be decveloped. However, I was able to create a simple acorns collection system and a checkpoint system. The collection system required acorns sprites so once I created them, I put two in the scene to use for quick testing.

I placed the acorns under PlayerPrefs to store this data. It was the first time I used PlayerPrefs, and I do not know if it worked properly. I wanted to check if the number of acorns is being recorded by the game system and also check if the UI text reflected the same number of acorns recorded by the game system. However, I ran into a s glitch, one day my code was working perfectly then the next day the acorns were not adding up correctly. I had stated under my character controller that if player collided with an acorn only one point must be added to the collection system, but the game was recording two points when player collided with one acorn.

I still could not figure out why it was doing that, but I figured out it was the use of PlayerPrefs that could be causing the glitch. However, I resolved the glitch by separating when game recorded the point and when the UI text recorded the point and it worked perfectly. I then moved onto making the checkpoint system. The checkpoint system is where the squirrel will have an opportunity to do three things:

1. Buy a health which would increase the health bar of the player if they took damage from an enemy NPC.

2. Buy resources that will help build the bridge to get to the house which is the end of the game.
3. Deposit acorns to the high score.

I made the checkpoint system into a shop that the player will encounter whenever he collided with a checkpoint game object. Whenever the squirrels enter the trigger of the checkpoint, a shop panel appears and that is where the player is given three options and must choose how they would like to use their acorn currency. I made a condition that the buttons 'to purchase' were not to be interactabke when acorns a below a certain amount.

However, the numbers of acorns on the game were not being recorded by the shop so even if my condition were to have more than 20 points in order to purchase anything in the shop and the player had collected that amount, the button would still not be interactable. I assumed that thus is because of that error of the game system not recording the number of acorns. However, when I figured how to make the game system record the number of acorns then the checkpoint system m (shop) could be read in it then the buttons became interactable and purchases were able to be made. I, then tested the game and the collection system and checkpoint system worked decently together.

The next week we had to create NPCs, both with friendly interactions and enemy patrols. I think I made these NPCs systems quite simple. I made an enemy patrol where I first started with the drawing of the fox enemy sprite. Once I did that, I then moved onto the code of the patrol. I wanted when the player collided with enemy patrol then the squirrel's health would go down. I showed the health of the squirrel using a health bar which was easy to set up. Thus, whenever the squirrel collided with the enemy, it would lose 20% of its life. The next step was to create friendly NPCs. These characters are mainly there to out forth the narrative side of the game.

I wanted the friendly NPCs to patrol like the enemy NPCs but instead of damaging the squirrel, it will stop, and a speech bubble would pop up to communicate with the player. However, I did not have time to figure out how to make the friendly NPCs stop, I tried using Vector.zero but that did not work so I decided to move on. The first friendly interaction I just a clue to the player on what the objective of the game is and the second friendly interaction, the character is offering the squirrel a trade. The player would have to trade in 30 acorns in order to get the ability to triple jump or to get more resources to build a bridge to get home. However, I was not able to make this trade system because of the time. Thus, I gave the player double jump whenever they've collected certain amount of acorns. This double jump will help the squirrel reach acorns that are too high up on a platform.

Reflection

I am going to discuss all the features of the game that I did not add, the adjustments I made and the reasoning behind these decisions. To start with the spatial puzzle, I wasn't too sure how to implement a puzzle in an exploration game, but it probably could've been a "get a

clue to unlock gate" type of puzzle, however I felt like it wouldn't add to the objective of the game. It would have seemed like just a mechanic added in the game just to make it feel complete.

However, one thing I have learnt about the past micro projects is that the point of the project is to insert mechanics that are meaningful to the game and not mechanics that are just are going to be implemented just to make the gameplay of the prototype longer. I could have replaced this spatial puzzle mechanic with something more explorative like have moments where the game scene goes completely dark for 20 seconds every 5 minutes during gameplay. During this dark mode, the squirrel would have followed, maybe a specific sound to lead it to a specific item or something like that. This dark mode would've added to the game's thematically driven aspect of it being tension-driven.

The second feature I did not implement was the trading system. The reason why I did not add was simple because of time. This reasoning might be common but living in times of loadshedding and piles of other assignments from other modules and life's misfortunes, it's really hard to implement everything one wanted to for their project. However, this is a risk that was taken into account which is time mismanagement and this needs to be improved in order to reach design goals for future projects.

The third feature I did not develop is a timer to show the player that it is getting darker and needs to get home before it gets dark. I honestly did not want to add a timer or clock to show that time is ticking and the reason for this was that it was something that was quite obvious to implement to show time is moving. I wanted to implement a change of background colors which would be controlled by a script. In second year, during the time when we were learning Twine, there was this mechanic that they taught us where the time of day would be shown by the background color of the game. For example, if it were night the background color would be black and as the time went, the lighter the background color would get. This sort of mechanic could be added to this prototype in the future.

In conclusion, the one thing I have learnt this semester is that these projects are meant to teach you something about one specific mechanic at a time and not aiming at making a fully fledged game. Thus, the mechanics I have implemented in the game work decently as I've tried my best to focus and fix glitches that I encountered instead of just leaving them broken because I'm rushing to implement all the features I've stated that I'm going to implement in the game. The reason why we called this project a prototype is to emphasize that this is a working progress and features not yet implemented will be implemented in the future. However for now, learn to create working mechanics and features that have meaning to the whole game.