Game Design Project Reflection: Last Post

Emmanuel Dade

Old Dominion University

GAME 201T

Prof. David Prihoda

Game Design Project Reflection: Last Post

For my game design project, I continued working on Last Post, a 2D side-scrolling platformer about a mailman. I wanted to create something that was feasible to complete. Some unique mechanics I would like to include were a storm that grows over time, a rival mailman that gets in the way, a shrinking and growing mailbag with magic, shaky and breakable rope bridges, and more. My Design Challenge research also influenced the project. I came across a job description for a level designer at Naughty Dog and saw that skills needed included level design, scripting and documentation. I felt most comfortable with scripting so I wanted to gain experience using that in the project.

I decided on an engine for the game. I used three different ranking websites to compare Godot, Unity and GameMaker and also read some user reviews on Capterra for each. I decided to go with Godot because it is free and has no licensing fees. This is important to me as a student because I do not want licensing or subscription fees to come up later. Another reason I chose Godot is because it focuses more on 2D than Unity does. Unity is primarily a 3D engine with 2D features added later. Last Post is a 2D side-scrolling platformer so 2D physics and rendering are important. The scene structure in Godot, which uses nodes to represent objects, also works well for objects that can take on different states. For example bridges can be stable, shaky and about to break. Likewise mailbags can be shrinking or growing. Also some reviews of Godot said it is less intimidating than Unity and I did not want to struggle with the engine while I was working on game design.

With this choice there were disadvantages. The community and asset store is much smaller than Unity. This could mean fewer tutorials or forum posts when something unusual comes up. Another issue is that Godot is more complicated to set up if I need to port the game to

a console. However this only applies if the project is to go beyond a class assignment. I felt that the 2D focus and the ease of learning were worth it.

Some things worked better than I anticipated. During my postmortem I used MinaPecheux's Mini-2D platformer tutorial from their godot-tutorials GitHub repository. I chose this because it is in the same genre as Last Post (Mario style side-scroller), has platforming, enemies, collectible items (coins) and a full game loop. I found the movement mechanics to be easier than I anticipated. After writing a character controller, I found that Godot's built in collision and physics system made implementing the player movement much easier. Using tilemaps also helped in level design because I was able to paint terrain instead of placing pieces individually. It was also easy to add the coin collectible items because of the Area2D collision system which is also similar to how mail will be collected. The slime enemies helped me learn some basic AI behaviors that I can apply to the rival mailman. I also found the player state system (such as a score) to be easier to implement than I anticipated.

I also had success creating a tutorial for the shrink/grow mailbag. I used an enum to assign a name to each mailbag state. I used a dictionary to map each mailbag state to a scale and a tween to smoothly change the scale rather than teleport between scales. This gives a more magical effect to the mailbag. I also had experience with scripting from my internship so I could view GDScript as a new syntax rather than a new skill. Godot also allows me to write a script and then play it without needing to compile first. This allowed for a quick testing cycle. Also in creating the tutorial I had to think about potential issues a person might face. For example mismatched input action names to script names, scaling issues with collision shapes and indentation errors.

Most of my issues were technical. My postmortem used C# rather than GDScript which meant I needed to install the appropriate .NET SDK and a Godot version with Mono support. This took more time than I anticipated. I also had issues with node structure. I had to experiment to figure out what nodes should be children of other nodes. I also had issues with collision shapes and sprites. They did not map perfectly so unexpected collisions would occur. I also had issues with C# syntax. I am used to other programming languages and had to slow down and read the code rather than copy and paste. The most difficult to debug were the errors which did not always tell me which script or node was causing the error. I overcame these issues by paying more attention to the explanations in the tutorial and changing one thing at a time rather than trying to make multiple changes at the same time.

Finally, this tutorial also affected the game I was creating. The ease in implementing the coins made me want to implement mail collection as a simple pick-up instead of some more complex system. I also had to reassess my storm mechanic because there was no example in the tutorial that had a storm that grows over time. I will need to do another tutorial to cover this mechanic. This is also something I mentioned in my Design Challenge assignment. I noticed that larger systems (such as a storm affecting several game elements) may require Godot knowledge that I do not currently have. I also found that it was not as important to choose the "right" tutorial. Even tutorials that do not implement the exact mechanics I desire can still teach concepts that can be applied to my project.

For the next step in the mailbag I can add a collision shape to allow for the small mailbag to only pass through narrow areas and I can also connect the mailbag states to a signal so the bridge can react to whether the mailbag is heavy or not. This project has given me greater understanding of how Godot works in a day to day basis. I found that nodes and scenes do more

than I had anticipated and I also have a better understanding of what is still needed to make Last Post into a fully playable game. I also practiced level design, troubleshooting and scripting which are also skills I would like to further develop as a level designer.