Student Name
studentName@student.lnu.se
21 August 2019

Technical Details
V1 No 1C - 4T04
1DV437 Introduction to Game Programming

Design Pattern

For the Player object I decided to use a Singleton because there is no multiplayer support and there should always only be one active instance of the player object at any time during run-time.

I would restrain from using a Singleton in most circumstances however I felt it was a good idea to learn how to use them (as I haven't done so properly before) and I am currently building the game on a very outdated iMac from 2011 which handles many scripts poorly. Thus overcoming the issue of the singleton being referenced in many different scripts, making it ideal for first time use.

For the Enemy AI I have used several observers to detect where the "ground" and "walls" are located to properly tell the AI when to turn around. I have also used observers on my Player object to allow it to climb structures. It first detects the wall and latches on and then it also detects if there is a ledge it can grab on to and climb over. Bullet projectiles also have a collision observer to tell it if it hit an enemy or an player.

It first seems as if there are a lot of observers in the scene however the levels are very compact and short and also within a 2D environment the number of active observers at one given time is not high enough to significantly impact performance.

I also make use of the built in HUD (Composite Design Pattern) to display health and current score in form of collectibles and score. However because of time constraints they are not displayed at the end of each level. I also wanted to display enemy health but that will have to wait for version 2 of the game.

Architecture

When I started making "V1" (the game) I followed a modular Architecture because of the scope of the project I was confident there wouldn't be any issues with dependencies and cluttered folders with scripts. However since I am building my game on an old iMac there were some performance constraints I had to consider after about half way through. It took almost 5 minutes only to compile one script and I felt it was too time consuming when jumping between scripts back and forth adding and fixing code.

Therefore I changed midway through to a combination of Ad-Hoc, DAG and Modular architecture. Certain classes are one huge code-review nightmare and another is a dream. This allowed me to keep some of my more structured scripts that I rarely changed and also made my "workflow" faster as I didn't have to spend hours upon hours of waiting for Unity to compile my scripts. Also Visual Studio could handle less tabs open thus it also improved performance greatly.

Collisions & Geometry

As I made a 2D Pixel “Adventure” game there isn’t a great deal of performance loss to clutter a structure with five Rectangle2D Colliders, though it made it frustrating at times when you couldn’t find the right one you wanted to edit. At this stage I simply left the collision detection to Unity’s default settings because accuracy wasn’t the most important requirement for this type of game.

The geometry is mostly made up of squares and the occasional capsule colliders for player objects and enemies so they don’t get stuck on terrain.

Game Design

I designed almost all models and sprites myself in Photoshop CS4 using pixel art as my design style. Therefore all shaders, materials and lighting is predefined and not affected during runtime. If my computer could support the Lightweight Render Pipeline or High Definition Render Pipeline I could use shaders and real time lighting on 2D objects. But unfortunately this was not an option.

When animating my objects I took the long route because I already had a few designs ready made beforehand which I could trace over with my new models making the walking animation fast and easy.

I could’ve opted for bone-based animation however I find it looks clumsy and ugly so I prefered to use multiple-sprite based animations.

Developer Comments

Unfortunately I overestimated how much work I could do in the allocated time of this course and therefore there are a few features missing from my submission. I would’ve liked to make a few more levels and an end game boss and also a small storyline which the player could follow. However since I got stuck for a few weeks with enemy AI I couldn’t finish in time.

Next time I will limit myself better and perhaps after a few additional tries come up with a reasonable scope for my projects. I would’ve also liked to have a pause screen, save and load function, but this is unreasonable for this course with the other features I have already spent too much time on.

Thanks to this course I think I have a much better understanding of game development and the processes of creating a game from scratch.