

Project Post-Mortem

of Solo Carp Game

by: Catherine Austria

COSC 491

Steven Kennedy

Table of Contents

The Reasons Behind ………………………3

The Plans and What Not …….……………3

The Process of Everything…………………5

The Reasons Behind

This project got me thinking of making a game that is quick, nice on the eyes, relaxing enough and simple in concept. In terms of simplicity, I thought of the old simple Nokia games like snake and asteroids. I suppose my inner child missed that. I do however plan to shape the game from where is it now near to those form of games in the future. Although it is quite simple right now, the game has the basics I need. A movable character, spawning enemies and things to eat. I also wanted the game to have some relaxing elements to it. I didn’t want a game where you have two problems in mind, spawning hordes of enemies and the ugliest assets you’ve seen, heard and played with in your life.

The Plans and What Not

My project proto wasn’t as detailed as I wanted it to be. ~~(Because a certain part of thyself, attached the wrong document and ended up attaching the picture with the same name instead, AND didn’t review her submissions. Lessons Learned.)~~ But the first main concept was to have a koi (carp) fish that ate only the foods that was its own colour and avoid other foods of a different colour. Then ‘those’ other foods became enemies, because what responsible fish owner would feed the wrong food to their fish? Maybe an ignorant one. So the player now had enemies. I thought of a few enemies besides the odd-looking tadpole-frog-thing actually. The enemies I had in mind would also have special characteristics. For example, I wanted to have a turtle (people do put turtles in ponds right?) that is basically very large for the screen and make it hard for the player to avoid him. It’s like I wanted to have that attitude of, “Out of the way, big diva coming through”. Another concept of mine was toads. And because they would croak loudly, they would cause … just work with me here… the food to scatter away from him. Why? Because ultra-sonic sound waves that’s why. These enemies don’t spawn as fast as tadpoles though, so the player has a chance to not get smothered with enemies. However, for the purpose of this simple submission, it will be just freakish-tadpole things for now.

Another aspect of the game I wanted to implement but had an embarrassingly long-time doing (ultimately making me postpone its implementation) was making the fish change colours as he ate the food, making the player have to be wary of what food he ate. Basically I wanted to implement an array of colours then change the koi’s colour schema based on that array. This would have allowed me to make a spawn for food of a different colour which may give rise to poisonous foods which the first original concept had.

As for the enemies, I had actually implemented an AI where the tadpole would follow the player. However, my AI acts really funnily where the tadpole would follow the player weirdly like it would zoom past the player, zoom past the player again, and then follow the player with z-axis applied, making the tadpole look anorexic and high. When I implemented this, it should be noted that it was 3am in the morning and I ultimately decided to scratch this out and came back to it later.

The last part of the game that I thought I might need for the future, but was not given that much attention to (due to the demand of graphics rendering) was to have the player shoot a burst stream of bubbles at the enemies and destroying them. I wanted a nice animation for this but rendering them out was harder than I thought. If I really wanted to, I would’ve gotten down to my old basics of drawing this out in paper, scanning those drawings, animating them in Unity and then doing the same process with the ripples. I realized this when I found that there aren’t many examples of water sprite sheets. I also cringed at this thought considering I already spent quite a lot of time rendering the game’s background. And when I looked at the unity asset store, the 2d water physics were all sideways versions and not top down or you had to pay for the assets. And due to the cheapskate that I am, I decided to just come back to this implementation in the future.

The Process of Everything

My approach to making the game was to visualize it through assets and then implementing it then. I remember having spent a great deal of time making my assets and animating them out before actually scripting them out. Some of my assets, I decided to script and animate at the same time. For example, the Start Menu had me create the objects and buttons, but at the same time I had to script them out immediately after as to not forget their associations in code.

I remember also spending a lot of time thinking how I would shape the game and code in such a way that it will not just satisfy the requirements of this class assignment, but also help me out in editing for the future. I made sure to modularize my methods. I have a lot more scripts than I had originally for the sole purpose or pinning down their use in the future and making their functions either reusable to apply to the other game objects or available to more edits in the future.

Since Unity 5 is a bit recent in version, I had a hard time looking for scripts that mapped out what I needed in concept. Most of them were version 4 and below. Some even dating back to 2011. Although I had multiple browser tabs open of sites with multiple interpretations and possibilities as to what I actually wanted to do, I had fun learning and finding out the differences of Unity 5 and 4, 3D versus 2D and how to script my methods.

No matter how simple my project seems, I am proud of what I’ve done so far and am looking forward to implementing and developing it more in the future. Given a bit more time to learn and apply what is learned, I think I can apply more challenges to my project and shape it to be even more ambient.

fin.