Eric Thoman and Lucas Manning

Introduction

The goal of our project was to create a comprehensive COS 426 platforming game that allowed students to “fight” through the class, with each level of the game corresponding to a COS assignment. We put a lot of thought into which elements of the class we wanted to incorporate into the game to give the user the best experience possible playing. Ultimately, we decided to implement just a few core features from the class, as when we thought of all the possibilities we realized that creating a game with too many mechanics simply was not a fun user experience. Instead, we chose to take a few core mechanics, such as reflection/refraction, animation, camera movement, and world-building, and create an in-depth game that was not only intuitive but incredibly enjoyable as well. We wanted the user to be able to experience COS 426 in a 2D platforming way.

The game is geared at any audience, really. As much as it focuses on COS 426, we tried to make it intuitive enough that anyone with a basic knowledge of platformers would be able to jump in and have a fun time playing. Basic movement is mapped to WASD, and most other things come naturally to people who played Mario as a child. The game employs many simple mechanics that facilitate a seamless user experience, such the player animations, friction/movement and corresponding animations, and linear levels that lead to the “submit” button.

Lucas is the only one with any 2D platforming experience, but even then we started the project in a fundamentally different way than his former game. In high school, he made a simple 2D game that employed basic physics such as jumping and camera movement that kept the screen focused on the box that was his player. From the very beginning, we kept our code as modular as possible which already constituted a big difference from what he had done in high school. By keeping the code as modular as possible, we were able to scale the game efficiently and find bugs that otherwise would have been very difficult to isolate.

The basic approach that we used was an update loop that redrew everything on the screen with the given tick rate. At each interval, new velocities and entities are calculated and displayed accordingly. With JavaScript, this seems like not only the most intuitive choice but the only one – there is no way to micromanage assets in JavaScript as there is in other languages such as C#. All of our code runs out of our main function, which constantly updates every facet of the program depending on player input and state. For us, this approach worked perfectly well, as the project that we emarked on was in no way graphically intensive. The computer had no problem redrawing all of our assets many times a second.

Methodology

One of the trickiest parts of this project was finding game assets that fit our desires and expectations of what we wanted out of this project. It was very difficult finding free characters and otherwise necessary assets that matched our vision of the fight through COS 426. There were not really multiple implementations of our approach – you essentially have to show the right png file from your animation sequence at the right time. We approached this problem by having each animation (running, idle, falling, etc…) contain an array of frames that extended the full length of the animation, and we told the update function which frame to look at depending on the state of the character.

If we’d had more time or access to more artwork, we would have loved to add much more texture and life to the game. This is one of those projects that you can never really finish, you just give up improving on. We could always make something act a little smoother or look a bit better. Another piece of the project that we would have loved to have expanded on is the thematic individuality of each level in the assignment. As of now, the levels share some common elements, which is necessary, but it would have been interesting to see if we could implement more pieces from each assignment in the corresponding level. Also, we really wish that we had the artistic ability to create a model of Professor Finkelstein as an enemy in the game.

// CHANGE IF WRONG

The final challenge that was stumbled across was enemies – we wanted them to act with some sort of artificial intelligence but that was out of the scope of this project. Instead, we made them act according to the player location and distance from the entity. I’m sure there were many ways to approach having enemies in the game act in an interesting way to play against, yet the route we took seemed like the most straightforward for what we wanted in the game. We would have loved to expand on the enemy design and behavior if we had the chance

// answer following

* + What pieces had to be implemented to execute my approach?
  + For each piece ...
    - Were there several possible implementations?
    - If there were several possibilities, what were the advantages/disadvantages of each?

Results

The way we measured success was simple – how fun was our game? The entire point of our project was to create a lighthearted, playable, highlight reel of the course, and we measured our success based on how much joy it brought us to replay back through the course. We did several user tests to incorporate feedback into our game, and we made some changes for the better – one user had a level design qualm, while another thought that a slight tweak to velocity scaling would be good for the look and feel of the game. After we incorporated those changes, people seemed to really enjoy our creation.

Our experiments, namely with user enjoyability, simply gave us feedback with which to further improve our game. We sat in Frist and had strangers, friends, and acquaintances come play our creation. We gauged how easy the game was to pick up for the first time, how intuitive levels were to complete, and how difficult and combat system was in the game. By incorporated all the user feedback into our model, we created a game that we think was the most fun all around.

Discussion

* + Overall, is the approach we took promising?
  + What different approach or variant of this approach is better?
  + What follow-up work should be done next?
  + What did we learn by doing this project?
* Conclusion