

Noah Pujol, Jeremy Schneider, Leo Kartsonas
Larry Gates
C200 - Intro to Computers & Programming
19 June, 2020

C200 Final Report

Description of Project: Our final pygame is called Galactica-y. It is a simple game in which you play as a small red spaceship that has to dodge incoming asteroids. You control the spaceship by using the arrow keys or WASD keys on your keyboard. You are allowed to move in all directions by pressing and holding down the keys. You are not allowed out of the boundaries of the screen. As the game continues, it becomes increasingly harder to dodge the asteroids as more begin to spawn in at a faster rate. The asteroids themselves spawn at the top of the window and move down with a standard velocity. Your score is decided by how long you survive, and once the game is over your time is recorded in a separate file.

Ideas About Designing The Project: We came about this idea after our initial plan, a simon-says type game, was scrapped due to it being too simplistic and not adhering to the guidelines of the final project. We decided to make something that was similar to what we had learned during lab, as well as taking inspiration from other pygame examples we found online. Eventually, we came across a pygame example on the internet that was more complex than our game, but still had a similar concept. We decided to go with our final idea of Galactica as it fit all the requirements of the final project.

Troubles Along The Way/Issues Encountered: As previously mentioned, our major problem was that our original plan had to be scrapped as it was too simplistic. The initial plan was scrapped on Monday night the week the final project was due, and this resulted in us ditching the old code and starting back from square one with only four days to start creating our game. After deciding on our final idea, we began running into problems with our code. The most glaring problems came from the inheritance of the character class, passing down into the hero and enemy subclasses. The first issue encountered was getting the hero to move. That was solved by placing the movement code into our main while loop. We then had to figure out how to keep the hero within the boundaries of the window. This was done by giving the hero subclass its own method that held it in bounds. Next, we were struggling to work with the enemy subclass, getting it to spawn randomly, move, and detect collision with the hero. We were first able to solve the collision issue by giving the enemy subclass a collision detection method. The method used the pygame sprite features to detect collision with the hero sprite. We were able to solve the movement and spawning in one move by giving the enemy an empty list within the main game loop, then defining its movement at a constant speed, then constantly updating the enemy list with an increasing difficulty of spawned enemies. Our final issues came with displaying the game-over text and timer text on screen, as well as recording the player time onto a separate file. Everything but the game over text has been resolved.

Discoveries Made: Across all members of the team, we discovered more about pygame and using object oriented code to write code. We each had different skill levels in programming, but we played on each other's strengths to reach our final code. Within our game code, we learned

new ways to make things happen. Using the sprite collision detection was something that had only been mentioned in lecture once before and required our own research to figure out how to implement. Allowing the hero to move with a continuous key press was something that was not taught in lecture or lab and required outside research. Getting the asteroids to spawn at random and in an infinite amount required help during office hours, but we were able to utilize that code and build upon it to enhance the difficulty of the game . Overall, it was a total team effort, with every member discovering something new about pygame, object-oriented programming, and their own strengths along the way.

Reflection On The Project: By the end of the final project, we are proud of where we have ended up. It is not easy having your initial plans and code scrapped the week of the final project, but we were able to overcome the time constraints and create a functioning game. It required a large amount of effort from all members to think outside of our own respective comfort zones of coding and communicate effectively to have everything come together in the way that it did. As a group, we all worked well together. Everyone was quick to respond and provide help or feedback on the game code, and everyone contributed something to the code as well. The game itself incorporates much of what we have learned throughout the summer semester, and has its own additions from our own collective research that gives us a final code that is totally unique to our group.

Problems With The Final Product: As proud as we are of our final game, there are a few additions and fixes that we would have liked to implement. One idea that we had was to have the

hero spaceship shoot a laser at the asteroids. However, we could not figure out the code in time and had to resort to just having the hero move around the screen. We also would have liked to tweak the enemy spawning code to limit the amount of asteroids on screen at once, as well as increase their speed after a certain amount of time. In the game's current state, the amount of asteroids that spawn can be slightly overwhelming, but it is still manageable. They all travel at a constant velocity. This small fix would have made the game a bit smoother and more polished, but we worked around this by just slowing the spawn rate of the asteroids. Lastly, we would have liked to have a game-over screen.