

Nolan Schirripa, Christine Seng, Erick Martinez, Georgia Rushing, Graham Balas

Professor John

CSI 1430

6 December 2024

Project Report

Game Mechanics: The user is initially presented with a black ball at the start of the screen, and a set of colored blocks at the bottom of the screen. Using the mouse, the users can launch the ball with a click, and the ball will shoot in the direction of their mouse. The ball will bounce if it hits the top, left, or right walls of the screen, or if it hits any of the colored blocks. After the first hit, gravity is “turned on”, and the ball will now also experience a constant downward force to make the bounces more similar to the real world. The main objective of the game is to maximize the user’s score, and to prevent any of the colored blocks from reaching to top of the screen, causing the game to end. Upon creation, each block is given its own life, and the block’s color reflects its life. To reduce the life of any block, the user must hit the block with the black ball. If the block’s life reaches zero, the block disappears from the screen. In the upper left corner, there is a text box that says “Score:”, and displays the user’s current score. The score is calculated based on the number of blocks the user has destroyed. As the user’s score increases, the starting value of the lives of new blocks increases, making it harder for the user to prevent any blocks from reaching the top.

Story/Plot: Three clans. Squares, triangles, and circles. Each one different from the other, but each one cannot live without the other. All three coexist in a land called Geometria. The squares are mighty, courageous, and resilient. Their kingdom—Quadratrica—stretches among the planes, is vast, and their people enjoy many luxuries. The triangles are great blacksmiths and swordsmen,

and no other clan has points as sharp as theirs. They spend their days in the mountains of Pythagorea waiting for their lost king Right Triangle to return. The circles are simple farmers rolling along in Roundhaven. They are said to have the most delicious pi. Their people are full of greed and are missing points, however. Born into the clan of circles was a cursed circle named Ball sent to punish the people of Roundhaven for their evil. Ball contained no color and was smaller than the rest. He was given the power to suck the color out of a shape until they disappeared. Ball—controlled by a force that was not from Geometria—reigned terror on Roundhaven. The circles begged for help. The other clans decided to help them so the balance would not be broken. The allied army of shapes stand at the gate of Roundhaven to stop Ball and keep the balance. What they do not know is that Ball stands ready to fight to rebirth the world into his User's new and just image.

Aesthetics: By combining elements of SDL Plotter, SDL Mixer, and SDL TTF, we were able to create a cohesive and visually and aurally pleasing game. With SDL Plotter, we learned how the predefined color struct worked to choose the colors for the blocks. We also randomized the difficulty and locations of the blocks as they appear on the screen in order to add an element of randomness and more colors on the screen. With SDL TTF, we were able to create a text box which displays the score for the user as they play the game. Using SDL Mixer, we play a popping sound effect every time the user hits a block.

Technology: When installing our version of the SDL game engine, our team collaborated on multiple IDEs, including JetBrains Clion, Microsoft's Visual Studio Code, and IBM's Eclipse. We organized our code using the GitHub repository feature, which allowed us to push and pull code from every team member. The game was almost entirely created with the `SDL_Plotter.h` functions,

which simplified SDL2 functions. However, SDL2 native renderers were used for images and text in our game window.

Teamwork: In our first meeting as a team, we divided tasks among each other, giving each of us a starting point. Then, as the game grew in complexity and size, our tasks grew in the category of our first task. We did this to create "specialties" within the team that allowed for a smoother workflow within the team. As we debugged our code and got familiar with each other's code, we split bugs evenly to the team and again focused mainly on our specialties.

Goals: We knew that our game wouldn't stray from the example Falling Balls game, so we mainly followed that reference when creating it. For players, the goal is to reach the highest score, or number of blocks destroyed, without allowing a block to hit a certain threshold at the top of the screen. The threshold gave players the objective to strategically bounce the ball between blocks to hit as many as possible. The blocks also carried a health attribute that increased in difficulty after every level, so every hit wasn't necessarily a kill.

Reflection: We started the project around the end of October. Our team met around 3-5 times a week almost every week after the start. The process started off slowly at the beginning because we had to familiarize ourselves with SDL_Plotter, animations, GitHub, and even our IDEs. At the start of the project, we did not know each other well, so it took us a few weeks to get comfortable with each other. It also took us a little to become comfortable receiving and giving criticism. After a few first slow weeks, we hit the ground running. The first half of our project was implementing the physics of the game. We started out by making a hitbox full of flags around the ball and when a flag was hit the ball would change directions based on the angle and speed. We ran into several errors here, however. The ball would occasionally not bounce off walls, would glitch in all sorts

of directions, and would get stuck in shapes. After days of debugging and tweaking our design, we were ready to move on to the shapes. This was not as difficult, but we did run into a few issues. After debugging and testing we found that we were going to have to use more pass by reference variables because our blocks were not being updated when hit, and instead copies of them were. The remaining weeks we worked on making our code clean and designing a UI. After over a month of working on the project, our programming skills, teamwork skills, and management skills drastically improved. We started off with a completely blank slate and with people we barely knew, and we finished with a fully functioning game and new friends. After all the bugs, stress, and late nights, the project turned out to be a fruitful experience.