Tammy Hartline

08/20/2023

CS-330 Final Journal

For this assignment, true to my style, I might have made things a bit more complicated than necessary, but that's how we learn, right? I strategized and implemented several approaches to ensure my final output aligned with the assignment's rubric. One effective tactic I employed was organizing my classes into separate .hpp files and then seamlessly linking them to the source code. This played a crucial role in keeping my project organized and allowed for efficient testing of individual features.

I began by reshaping the layout of the bricks on the screen. By meticulously arranging them in a pattern on both the left and right sides of the window, I achieved a visually appealing arrangement. Additionally, I integrated a paddle at the bottom of the window that can be controlled using the left and right arrow keys. However, while I aimed to make the paddle reflective, I encountered some challenges in achieving this behavior.

My next hurdle was mastering the physics of altering the direction of circles upon collision with walls, the paddle, or bricks. Strangely, achieving this proved to be quite intricate. Balancing the correct collision direction while maintaining realistic ball speeds proved to be a delicate balance. It involved adjusting various factors such as speed, velocity, mass, impulse, and friction. Striking the right balance here was a bit elusive, as the speed sometimes ended up too slow or excessively fast. Another puzzle was the spacebar's sensitivity, as I aimed to limit the number of circles released with each spacebar press. However, despite my best efforts, I faced difficulties in achieving this fine control, leading to circles clumping together or freezing in place.

Shifting my focus to the state of the bricks upon collision, I managed to succeed in one area. I implemented a system where, upon collision, a brick disappears if it's marked as "destructible." For "reflective" bricks, the collision transforms them into triangles. Implementing this feature was surprisingly straightforward and yielded the intended results.

My final endeavor was modifying the state of circles upon collision. Given the changes to the bricks' behavior, I opted for subtle yet noticeable modifications. As circles were generated, I assigned them random colors. Collisions with walls, paddles, bricks, or triangles only affected their direction. However, if two circles collided, the one responsible for the collision turned pink. This implementation required persistent trial and error before achieving the desired outcome.

Reflecting on the project as a whole, I am satisfied with my progress. Though challenges emerged, I aimed to create a 2D simulation resembling rubber balls bouncing within a transparent enclosed environment, complete with "pinball-like" elements to interact with. While I integrated realistic effects, fully mimicking a real-world environment proved more complex than anticipated.