Hannah Norman
SUNet ID: hnorman
Professor James
CS 248B – Fundamentals of Computer Graphics
26 October 2023

Programming Assignment #1: Pinball

Link: https://openprocessing.org/sketch/2063463

Discussion of Features

I modeled my pinball game off Pac-Man in terms of visuals and sound effects. The playfield is a special box that is thinner in the middle of the screen with two tunnels extending out of it. If the ball, which is an animated Pacman, goes down one of these tunnels, it will be transported across the screen and come out of the other tunnel in the opposite direction. The ball begins each round in the left tunnel and is shot out of it to the right when the player hits [*Space*]. Also, the playfield includes three basic block obstacles for the ball to bounce off.

Another feature I implemented is the four moving ghosts. Each ghost moves on its own path and is animated while doing so. The ghost obstacle itself was created with a combination of a circle and box SDF. If the ball hits one of the ghosts while they are rainbow, then the player's score will go down by 20 points, but if the ball hits one of the ghosts while they are blue (in power-up mode), then the player's score will go up by 100 points.

I also implemented dots and power pellets (the four larger dots) which the ball can travel through. Each dot and power pellet hit by the ball will award the player 5 and 50 points respectively and disappear from screen for 10 seconds before being regenerated. Each power pellet will also trigger power-up mode, where the ghosts turn blue and can be hit for points over 10 seconds. Additionally, one of five fruits (each worth a different number of points—cherry for 100, strawberry for 300, orange for 500, apple for 700, and melon for 1000) will appear in the center of the playfield every 10 seconds. Once spawned, the fruit will remain on screen for several seconds, the total of which decreases as the point value of the fruit increases. Furthermore, if the ball passes through a fruit, it receives an impulse to double its velocity.

Finally, I added score and high-score trackers above the playfield. The score updates live as the player plays, and the high score updates after each round. Moreover, I track player lives and fruits eaten per round below the playfield in image format. The player begins with three lives, and once they run out, the game is over. If the player collects all five fruits in one round, they earn an additional 2000 points, a fun song plays, and the fruit counter is reset.

Beyond creative components, I also implemented discrete and continuous collision resolution. These implementations are located in the `normalFD2()` and `Ball.timestep()` / `rayMarchTimestep()` functions respectively.

Discussion of Bugs/Issues

While implementing ray-marching, I encountered a bug where the program occasionally enters an infinite loop when the ball hits the flipper at specific positions. I have struggled to reproduce the bug but believe it still exists in my program, although I have implemented some light fixes to ensure the remaining time step amount is always decreasing. Unfortunately, this means some interpenetration might occur, despite a concerted effort to avoid it (e.g., I attempted to implement the post-timestep position correction described in the instructions).

Additionally, at times, the sound effects will layer when the ball hits a flipper and produce a robotic, loud whirring noise instead of the quick *boing* noise. Given more time, further optimizations could be added to solve this issue.

Finally, due to the various flat horizontal surfaces in my playfield, there are instances when the ball becomes stuck in place. To fix this issue, I apply a small impulse to the x-component of the ball's velocity to shake it out of its position when it gets stuck.

References

Font: https://www.fontspace.com/emulogic-font-f3327
Sounds: https://www.sounds-resource.com/arcade/pacman/sound/10603/

I created all the image assets on my own using Keynote, though I referred to publicly available images of Pacman on Google for reference.