Spring Interactive Reflection C00302210 RoBert McGregor

Reflection Video should be in the Project Repo

Time Management:

TOTAL TIME:

56 hours. (Probably longer, missed some logging)

How Many Sessions:

17

Art / Code / Design (Total vs Estimate):

0 time spent on art – the coding / physics side of the project proved to be far more time consuming than my initial estimate.

Workflow:

March 01 – Design Document created and added. Brief values on gravity, friction. Created Ball class.

March 08 – Built some debug tools (cursor pos printed on scoreboard), recreated Ball class. Began work on nudge class as this would allow me to move the ball and test collisions. Added math defines. First problems with constructors.

March 09 – Rudimentary bouncing against screen bounds working fine. Wrote some helper functions (eg. Clamp). Ball array to accommodate multiball. Started to work on screenshake – curious about SFML Views and I know it'll add value

March 10 – Built a Lerp function, created kick and settle for screen shake. Have rotation, movement and size for screenshake.

March 11 – Spent the Sunday working on audio, cutting samples, setting up audio functionality. Also working on random range function, though this is tricky with just ints – not really happy with how it affects screen shake. Ambient audio working well.

March 12 – Rectangle and bumper collision working now – rectangle should work for targets, walls and kickers. Really pleased that reflection works properly!

March 13 – Changed scoreboard font. Got some great advice from Ben Millar on Random Range – now works properly for floats also.

March 14 – Project cleanup. Moved helper functions into their own class. Got flipper collision detection working (not collision reaction) - fought to use Dot Product for HOURS, eventually gave up and used aTan and it was good!

March 15 – Fateful decision to heavily refactor project. Moved Collision into its own class, moved Rendering into its own class. This needs to be done, as I'll be reusing my code heavily and I'm a big fan of DRY.

March 16 – Well, nothing's working and I'm in serious trouble. My foot is on fire, I'm behind schedule and I'm finding it very difficult to follow errors through the debugger. Not much progress.

March 17-18 Recovering from foot issue, taking painkillers. Starting to panic.

March 19 – Have got most of the code refactored now, but the ball is refusing to react to collisions – velocity not changing properly. Following this in debugger, I can see the value being changed in one class, only for it to step out and it's reverted to the original value in the calling class. ABSOLUTELY INFURIATING, BAFFLING.

Jay on Game Maker's Club points out the problem I'm having – I'm copying a value, not a REFERENCE. Of course, you copy by reference by default in C#/Unity. Christ.

March 20 – Project is basically back to where it was pre March 15th, but with lots of little classes instead of monolithic files. Have to believe this will pay off for me. It's clear I won't be doing art for the project at this stage. Gravity, friction applied to ball – if you squint your eyes, you can see what the game will be like. Got extension, thank god.

March 21 – Implemented all controls, using a bastardised version of a Control/binding pattern. NOT hooked up yet! Gutter added, launch with randomised kicker, teleport functionality.

March 26 – Slow mo (delta time scaling). Nudge thresholds. Flipper mostly working. More visual debugging for the ball and flipper.

March 27 – BEHOLD THE FIREPOWER OF THIS FULLY ARMED AND OPERATIONAL BATTLESTATION. Flippers up and working, visuals working, instantiated multiple times. Same for bumpers. Scoring working. Bed early to prep for medical appointment next day.

March 28 – Tidying up code and adding some additional comments. Not sure I'm going to hit the style guide exactly, but doing my best.

New Technologies:

SFML View (for screenshake/scrolling).

Atan, dot and cross product for the development of correct reflection vectors.

C++ architecture – starting to internalise the importance of passing by reference (as opposed to C# which handles referencing differently).

Pete's Inverse Radial Collision – instead of detecting when two circles are in touch with each other, detect when an object is a certain distance from the centre of a circle position. This allows us to create the effect of concave curved surfaces in 2D!

Building a helper library has been very helpful – I'm sure I'll be using these in forthcoming projects.

I've also spent a lot more time with Visual Studio and the debugging tools.

Problems Encountered:

- Physics – I'm used to calling physics functions, so discovering how they should be built from the ground up was a challenge. Getting proper reflection bounces was tricky, as was trying to get the flippers to work in a way which was reasonably fun and responsive. Sadly, it's still possible for the ball to fall through the flippers and into the gutter below.
- C++ architecture, specifically referencing objects. I've also wrestled with circular references, though in the end I was victorious.
- Refactoring the project was essential, as I knew I would need to reuse multiple pieces of code and place them at different parts of the table. My lack of knowledge of c++ hampered my ability to efficiently reshape the project, though I got there in the end.
- Poor time management left several features I'd been trying to accommodate unfinished, eg. Multiball. I was unable to implement targets, holes, saucers, spinners, gates and stoppers. I really wanted to add a lot more 'missions' and visual/audio effects!

Lessons Learned:

Overconfidence (and feature curiosity) hurt me several times during this project, causing me to bungle my time management and not leave slack for when I ran into unexpected delays and other difficulties. It's possible I should have planned this better while speaking to some of my lecturers.

Refactoring a project this complex with a new language was something I should have sought help with – it would have been an interesting learning experience instead of an excruciating learning experience.

I've developed an appreciation for the craft and techniques behind the creation of pinball tables – virtual or otherwise.

FURTHER STEPS

I'd honestly like to investigate more about collision through this project. I feel I was just on the cusp of making it great. =/

Credit and thanks to Jay and Ben on the Game Maker Discord for helping me track down the referencing bug that was driving me up the wall. Credit to Pete for his inverse radial collision idea!