**Drag Race: Post Mortem**

***What 3 prototypes of Drag Race taught me***

**1. Put down the keyboard, pick up the pen**

Heard of GDD (Game Design Document)? Do it a Coding DD too. You can kill a bunch of bugs before ever touching the ";" key.

I followed @michaeljohn 's advise and it worked wonders for my code.

- Game Design Document

- Feature List

- Priority

- Class lists

What I'd have done differently: dedicate more effort to this step. If I'd put in just 10 more minutes into drawing a timeline to capture what timings I needed to store, I could have saved 4 hours debugging my currentTime code. I'd have realized I used it in two definitions in different parts of the code - current lap timing versus current race timing.

**2. CTRL+S is the worst thing you can do**

BECAUSE IT OVERWRITES YOUR PREVIOUS WORKING VERSION! Then you spend 4hours trying to undo that one little innocent thing you reorganized. insert twit here

Sobs. Please use Save As, or even better use version control. Github saves lives. Not only does Github keep backups, it shows you exactly which lines changed.

For some time, Github's alien ways scared me. Five different tutorials and a classroom lecture later, Github still confounded me. Then I found out even poets can use Github too.

I still dont speak GitHub, but 3mins int the Poet Video and I had enough EUREKA to delete drag race v1 v2 v3 v3\_bugged\_lights v4 v5 and rainbow jet final, FINAL FINAL, REAL FINAL UPDATE. I can go trigger happy with ctrl + S now, just gotta GitIt.

**3. Love squares. Squares are your best friend.**

You made a box on the screen? Play test it!

Don't bother with UI, or titles or images. MVP for the win.

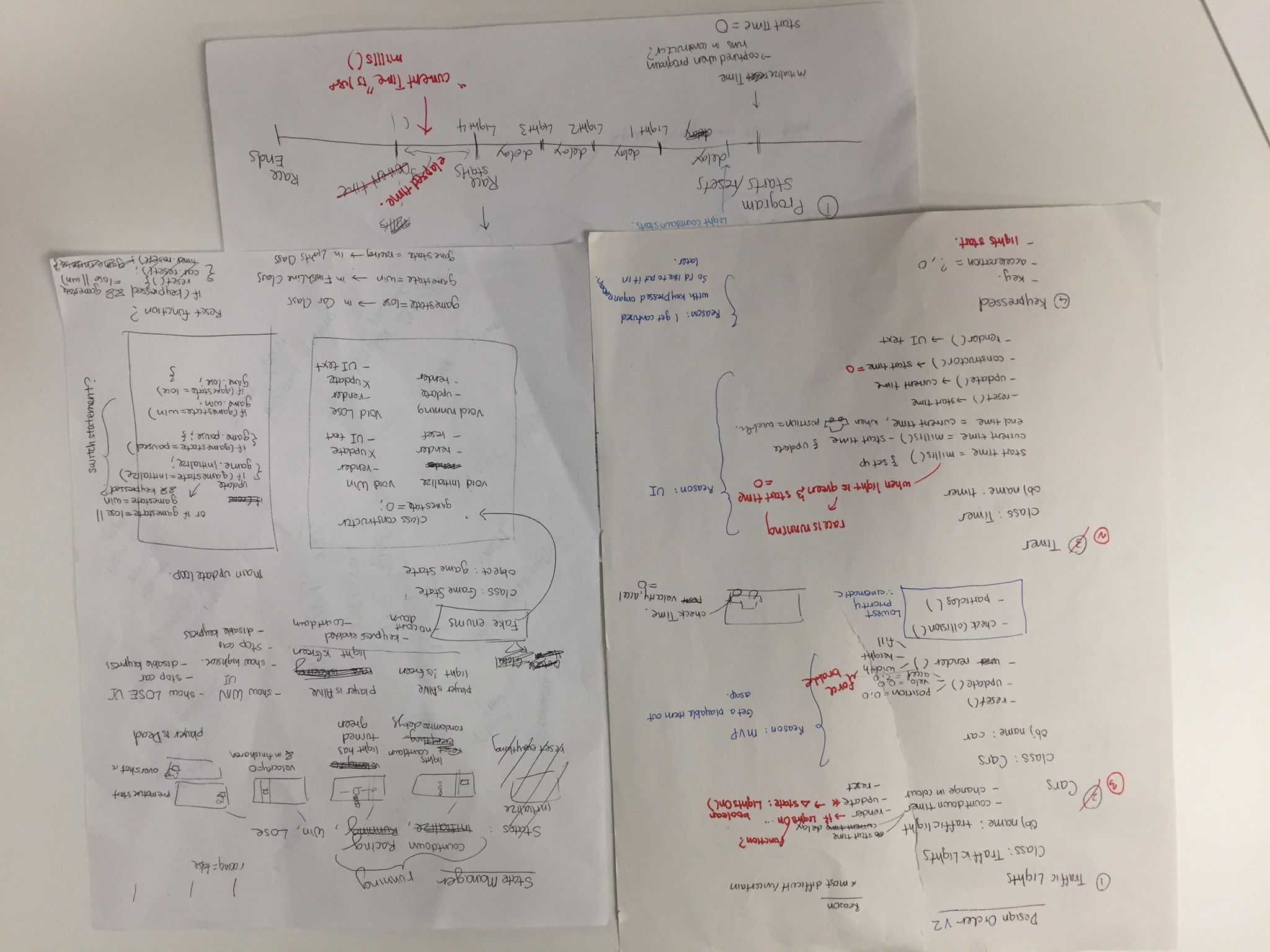
What I found was people loved my ugliest baby. I had actually birthed that stupidly simple prototype that as a joke. The version with added adrenaline-packed features, complex difficulty and 6 different car models was the least loved. Oh my.

I wish I had prototyped 10 hours of code ago. Do it for the people. Leave behind your overdressed baby.

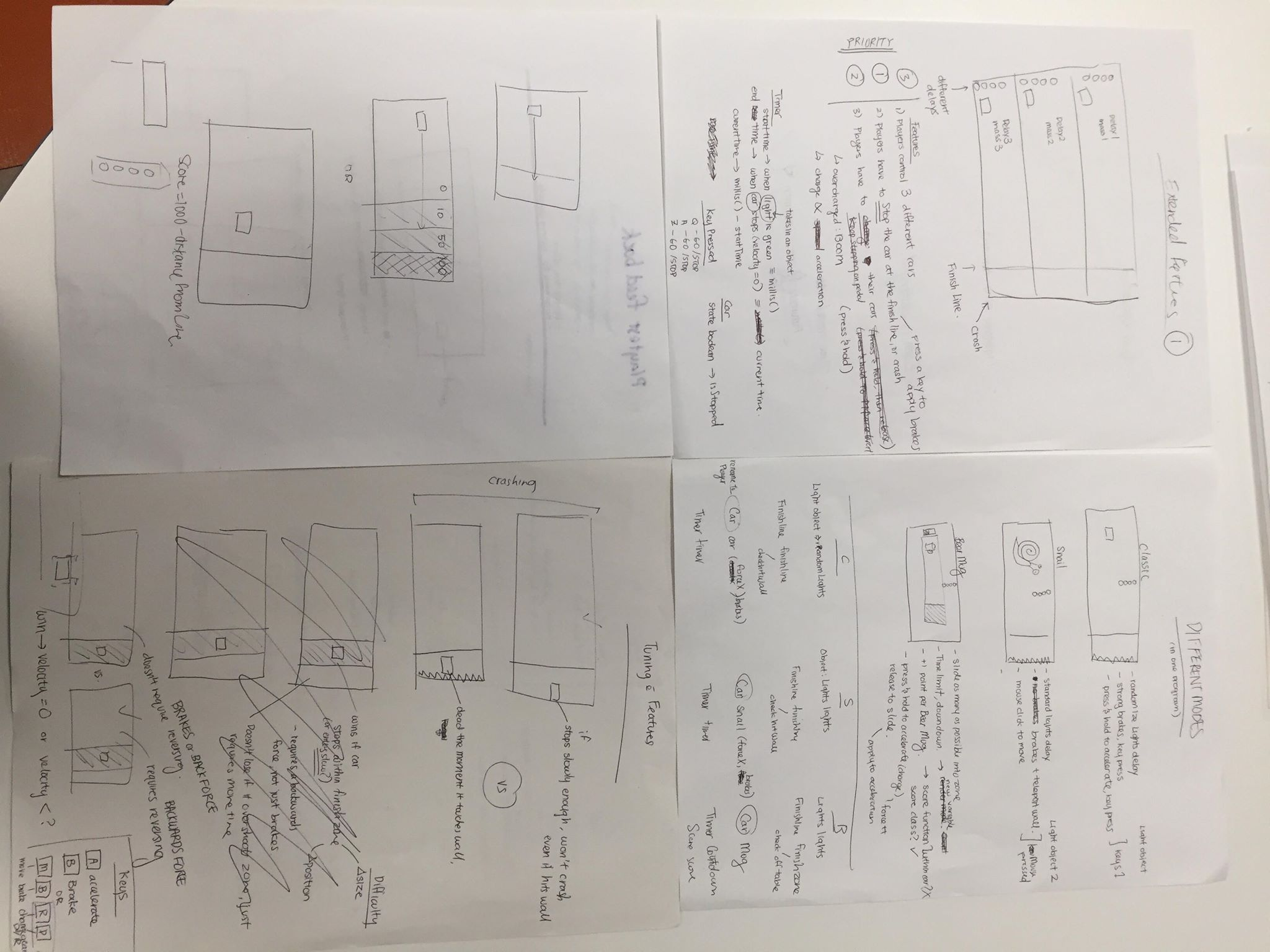
Github ( <https://github.com/Haiyoooo/DragRace> )

**How the game is different from the original design**

* Original design in **black**
* Differences are noted in **red**



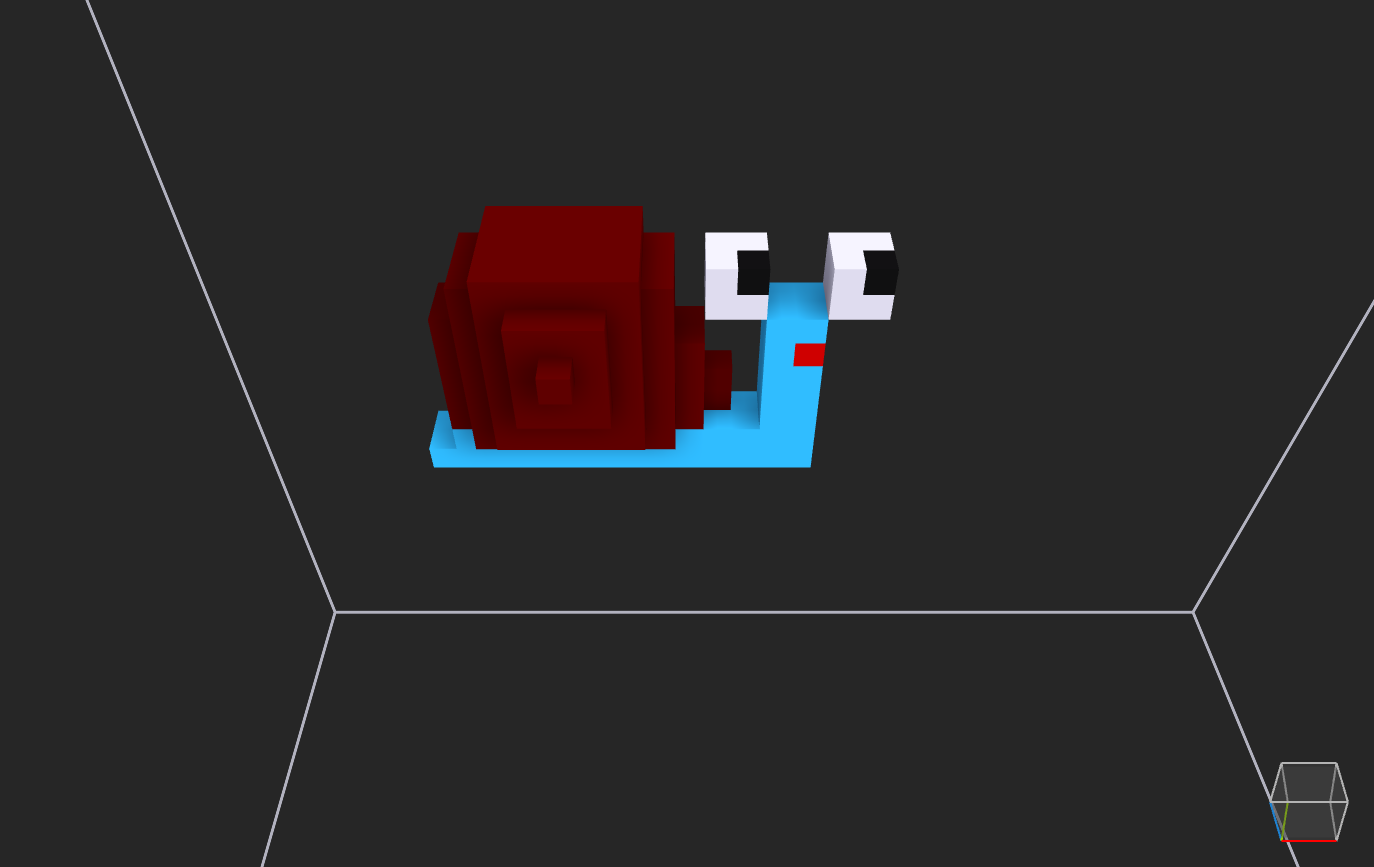
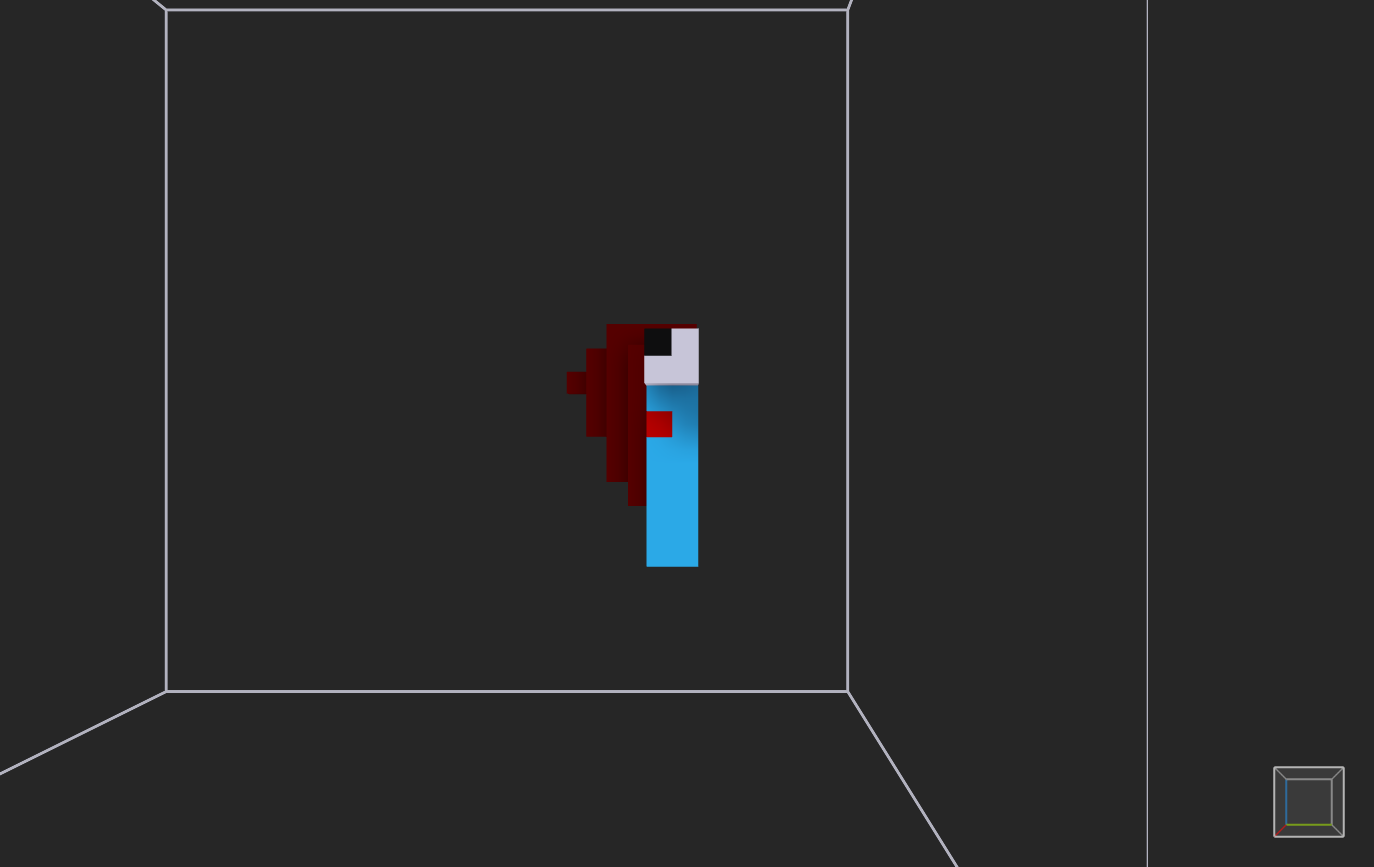
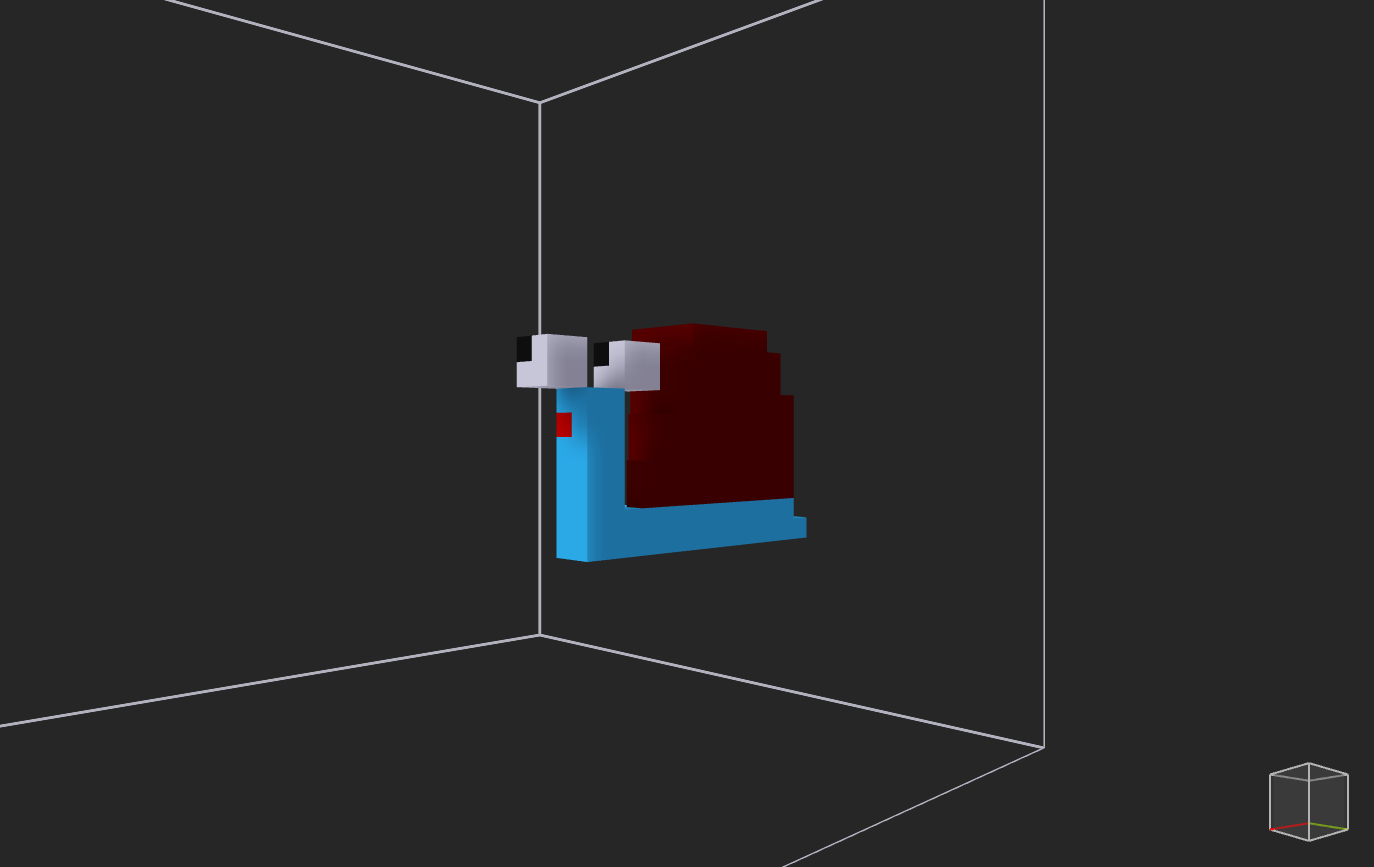
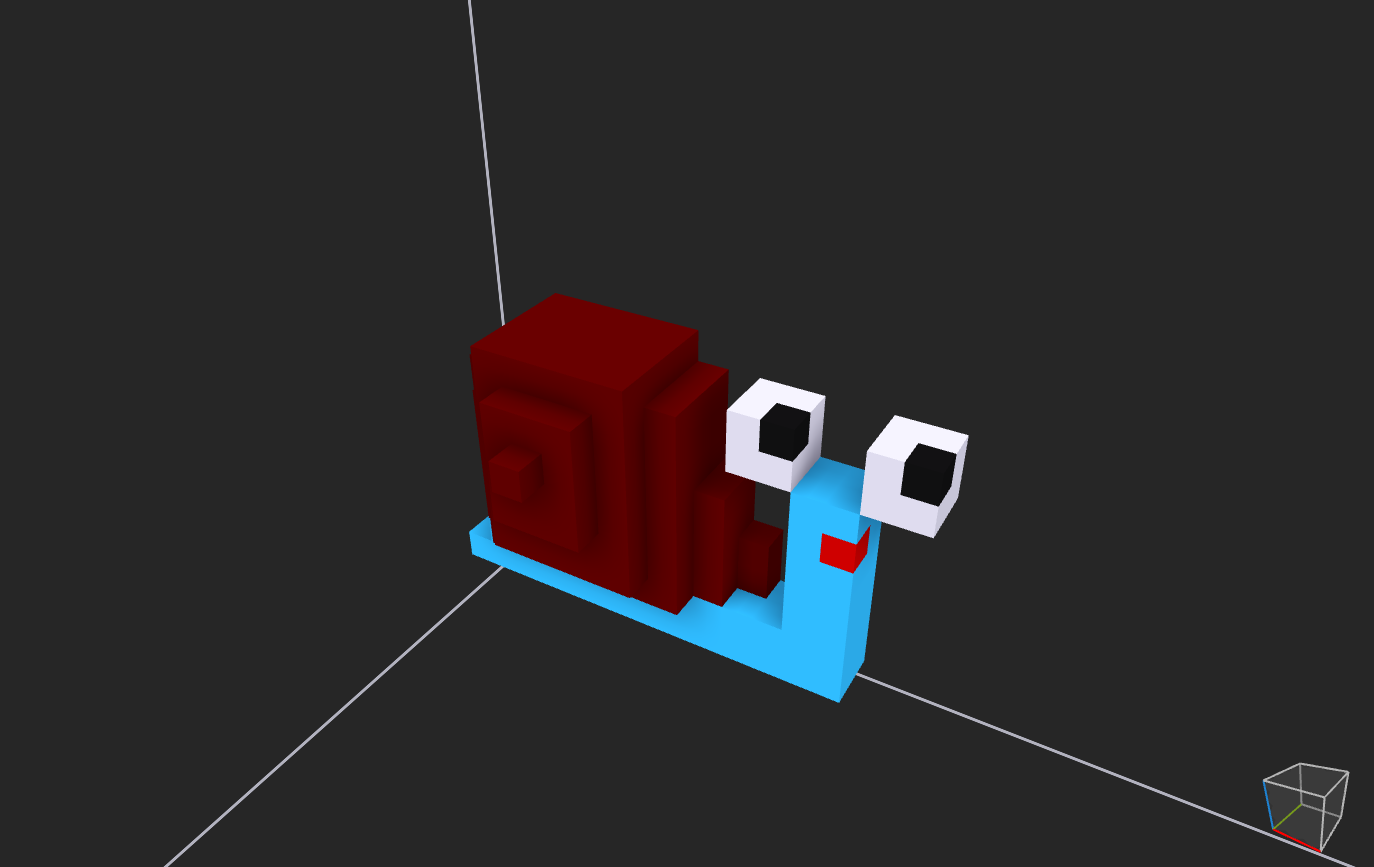
I had many extended feature ideas and picked three to prototype (<https://haiyoooo.itch.io/drag-racing>).



After playtest feedback, I chose the one players liked the most…. Which turned out to be the simplest version. So the code ended up way more complicated than necessary.

* No chance of crashing & losing into the right screen because the snail moves too slow
* No need “finish zone” object anymore
* etc

Then I made all the art assets in Voxel and put it in.



**THEN**

I realized I didn’t include “player presses a key to start light tree”.

That was the last thing I tried to do.

Everything fell apart.

Couldn’t fix it so I ended up hacking it.

**Greatest difficulties**

* Time was confusing
  + what timings I needed to capture
  + how to capture just a specific point in time instead of having the draw loop keep updating it
  + Naming the timings intuitively
* State manager vs State machine vs Enums…
  + Are they the same thing? Should they be different classes? What?
  + What goes under the state manager?
  + What goes under the draw loop?
* Debugger
  + Errors with variables not appearing
  + Errors with debugger not running or stepping the way I intended it to