Étienne Racine

Project 2: Proposal

General Description

This project is planned to be an extension of the final project I've done in CART-253. The goal this time will be to add more content into the game (both in terms of levels and of what's in them).

This project will only use general JavaScript, p5.js as well as some JSON for storage. Pre-built game engines like Phaser won't be used as to maintain a high level of control as to what exactly happens on screen and in the background.

The game will still be a platformer with a focus on speed and the player's ability to dash and so all content to be added should support that idea.

Path

As of writing this, I'm not exactly sure of what I will be adding to the game in terms of content. I plan to follow an iterative approach (similar to FTL's development) where I implement an idea which sounds good on paper, see if it works well and complements the whole and then make a decision as whether to keep it or not.

Some ideas within levels include:

- Optional or alternate objectives in levels.
- Floating objects that need to be pushed or shot into specific areas.
- Floating objects that refresh the player's dash mid-air.
- Obstacles that fly towards the player and try to slow them down or knock them back.
- Rising lava (or something similar) to add a sense of urgency.

Regardless of what is added, more levels and possibly a tutorial are to be expected.

The user's score or time will be stored into browser storage as to allow them to come back later and try to improve on it.

I also would like to expand onto the player's action set to allow more freedom of movement. This will likely take the form of contextual actions (like for example, jumping as soon as landing on the ground at high speed makes the player 'bounce').

Technical Challenges

There are still a lot of technical challenges that will need to be resolved in order to bring the project to its desired state. Most of them are structural elements related to the game's engine.

The current engine is *functional* but needs serious improvements to continue development. As of right now it's very rigid, not modular enough for a game engine. These improvements include:

- Generalizing game states into classes to allow loading and unloading data. Generalize state transitions to allow an arbitrary amount of levels.
- Adding a hierarchy of classes to represent all obstacles within a level.
- Re-write the current 'TriggerBox' and 'Effect' with the new class hierarchy.
- Add collision for entities that can be interacted with or moved.
- Grouping functionalities together into distinct objects / scripts (like music playing, for example).
- Almost completely re-writing how menu buttons work.
- Allow the user to re-map controls of the game to what they want. (Optional)
- Save the above controls re-mapping as well as other game settings and score into browser storage.

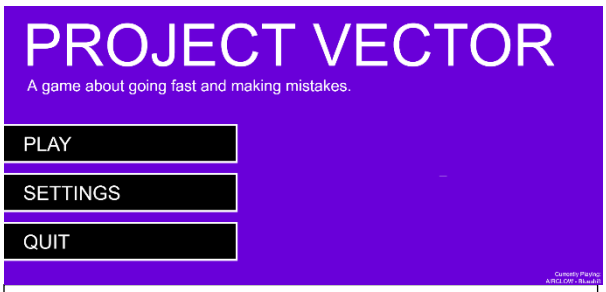
Some other general improvements would include:

- Properly displaying game information on screen. For example, in the current game, there is no way to know if the player can still dash if they are in the air (only the cooldown until the player could theoretically dash again is shown).

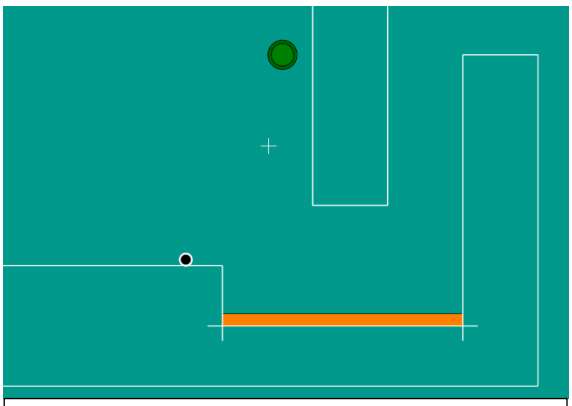
Artistic Vision

The project's initial rudimentary art style is very likely to be maintained. The plan is to only make use of the shape drawing in p5.js rather than using sprites.

Although rudimentary, the style will aim to be informational. This means that the function of every object or platform will be identifiable at a glance. Ideally, specific patterns will be used to make this information independent of colour, so that someone who is completely colorblind wouldn't stand at a disadvantage.



Main menu. Likely to stay visually unchanged.



Game state. Boundaries are likely to be made more obvious and more might be added to the background to make it more interesting.