

## **Project Description**

Project Name: No Brakes!

Description: No Brakes is a game where you are a small block that is constantly accelerating as you try and solve a maze. A new maze is generated every time the player restarts. There are drifting and collision physics present, so it will be harder to control the player than if there wasn't. After the player reaches certain thresholds of speed they move on to another "level". Every level has a base acceleration value and a player color to indicate which level it is. As you increase levels, it will get harder to control the player since your base acceleration will be faster. At the same time, you want to try and complete the maze as fast as possible, since there is a leaderboard of times that other players took to complete the maze. If the player wants, there is also an AI mode that can be enabled before the player starts playing the game. In the AI mode, there will be another player on the map that is being controlled by an AI. This AI will attempt to chase the player and run into them to cause them to lose. After the player loses, they can enter 3 letters to save their time to the leaderboard.

## **Competitive Analysis**

Many parts of this game were heavily inspired by a popular App store game called No Brakes. In the app store version of the game, players can choose from levels to play on, all of which are different variations of tracks that are closed loops. The goal of this game is to reach the highest speed possible without crashing. It has a similar speed level progression system to my version, as well as drifting and collision physics system. However, in my version of the game, the goal is to make it to the end of the maze as fast as possible, with the highest speed possible, without crashing! Also, my version implements an AI enemy and a local leaderboard, which is not present at all in the app store version.

## **Structural Plan**

I plan to break up the game into many different modes. A few of these being the intro screen mode, the player selection mode, the game mode, and the game over mode. These will all be coded in the same main file, using an app.mode variable to keep track of which to display. I also plan to have the leaderboard in a separate file, which will be edited whenever somebody wants to enter a new score.

## **Algorithmic Plan**

### Maze Generation

One of the most algorithmically complex parts of this project is maze generation. Since there is a new maze generated at the beginning of every round, I cannot just pre-make them and save them to a folder. The algorithm being used for maze generation is a randomized version of Prim's algorithm. Prim's algorithm works by starting with a point and its walls, then randomly choosing a point outside of the walls and connecting it to the first point. This results in a maze

with many long corridors leading to the end as well as short dead ends, which is perfect for gaining speed.

### Enemy AI

Another algorithmically complex part of this project is the enemy AI. I plan to create an enemy AI that follows the player around while also doing its best to avoid crashing into the walls. I hope to use the A\* algorithm to calculate what moves the AI would need to make to advance on the player.

### Collision and Drifting

The final complex part of this project is collision and drifting physics. Although not using any known algorithms, I plan to use an adapted version of physics equations and concepts to generate close to real results.

### **Timeline Plan**

4/23 - Finish drifting and simple collision physics

4/26 - Finish smart maze generation

4/30 - Finish enemy AI

4/2 - Finish polishing collision animations

### **Version Control Plan**

I am using a public repository on GitHub to back up my code. This can be found at <https://github.com/Quarks-1/No-Brakes>

The screenshot shows the GitHub repository page for 'Quarks-1 / No-Brakes'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, there's a section for the repository's state, showing 'master' branch, 1 branch, and 0 tags. There are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files and folders in the repository, including 'Notes Example Code', 'Testing', '\_\_pycache\_\_', 'cmu\_112\_graphics.py', 'mapGen.py', 'movement\_test.py', and 'tp0.py'. Each item shows the commit hash, the commit message, and the time since the last commit. At the bottom, there's a prompt to 'Add a README'.

File/Folder	Commit Message	Time Since Commit
Notes Example Code	dijkstra	4 days ago
Testing	tp0	4 days ago
__pycache__	fixed edge detection	3 days ago
cmu_112_graphics.py	first commit	11 days ago
mapGen.py	improvements	2 days ago
movement_test.py	trying	22 hours ago
tp0.py	tp0	4 days ago

**Module List**

None