After information presented in the design+ competitive analysis document, I have made several fundamental updates to my term project. The standout of these updates is that my game now has the ability to generate random tracks based on user preference. Here, the preference that the user makes is whether the track needs to be hard or easy. Based on this, the game generates a track. The game does this by an algorithm that randomized the space between two track objects and puts them together. The angle and lengths are randomized too. The bounds of the randomization are what change between the easy and the hard mode.

Additionally, the user has the ability to race on preset tracks and the aim of the user would be to beat their own best time.

One last addition is the ability to save the randomly generated tracks for later use. However these need to be run in the same run() attempt. I accomplished this feature by using several game class methods and using these to store information each time a random track is generated.

The user interface in this game is simple yet robust. This is because the UI itself does not play a major role in the user experience. Therefore, the majority of time and effort were spent in working on the game itself. I have a splash screen with clear instructions about the objectives of the game and I try my best to make the navigation between modes easy.