Indyracers: A VPython Program

Nicholas Kasner

November 2016

Abstract

As a novice programmer and a racing enthusiast, I embarked on a semester long project in (Visual) VPython to create an animation of a racecar moving around a superspeedway. A distinguishing component of the completed program is the ability of the user to toggle between two views, one from above the track and another from inside the moving vehicle. This narrative describes not only an explanation of the premise behind the project, but the mathematics behind racing and the coding involved in the featured program, "indyracers".

---

1. Introduction and Goals

My objective is to create a real-time interactive animation using VPython of a racecar moving around a track. Since I am inexperienced with VPython programming, I will rely on thorough analysis of other VPython projects in order to learn the things necessary to achieve my project goals. I may also require the assistance of the professor and mentors to assist me with coding errors that I am unable to understand. I hope that my enrollment in CS 101, a course which teaches the fundamentals of Python, will give me a basis for working with VPython.

In my program, the user will be able to switch between two cameras to yield two perspectives from which to view an automobile race. The first viewpoint will be an overhead view, which will depict the car as it progresses around a fixed course. This will be accomplished by first establishing a racetrack frame and then constructing a 3D car frame which moves along the closed path. The car will rotate with respect to the track as it enters corners so that the car is

always pointed in the direction it is travelling. Similar projects from which I can gain an understanding of a particles motion include the orbiting planets example.

The other camera will be from an in-car point of view, in which the user will be simulated as the driver. This view will depict a still image of the dashboard of a vehicle with a moving image inside a windshield. Time permitting, the dashboard may contain elements such as a steering wheel, a speedometer, and a gear shift. Ideally, the user will be able to interact with certain elements of the vehicles interior, such as moving the gear shift or turning the steering wheel. An example of user interaction that exhibits an effect on the program is controlstest, which enables the user to move colored rings from one peg to another. I am considering making it so that the user can adjust the position of the gear shifter to various locations in the gear box.

The windshield of the racecar will project the view out of the front of the racecar. The landscape shown to the user will adjust based on the racecars position along the track. This will be accomplished by texturing mapping an image of a racetrack to a cylinder on the inside. An example of this colloseum-style imaging is called look-around.py.

---

2. Project Timeline

October 31 - Proof of Concept RTICA
November 2 - Seminar Presentation
November 11 - Bird's Eye View of Racetrack
November 18 - Toggle Between Camera Views
December 2 - Inside the Car Camera

---

3. Update

As for what has already been done as of 21oct16, I have successfully configured the Repository on my personal machine which allows me to access files remotely that I edit in the ATLAS Lab. In addition, I have worked to outfit my personal computer with Python 2.7 and VPython, the programming languages I will be using to complete the project. I am also familiarizing myself with commands for the Windows Command Prompt, including dir, cd, and other methods useful for navigating the command line. Finally, I have installed MiKTeX for Windows and have begun to familiarize myself with its functionality.