Indyracers: A VPython Program

Nicholas Kasner

December 2016

Contents

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 Background

My appreciation for automobile racing comes from watching races with my father. When I was little, I would watch the NASCAR races every Sunday on television. Once I was old enough, my dad woud bring me with him to spectate the races in person. Watching racing with my father was not only a great bonding experience, but it also captured my interest in riding in a racecar. When I was thirteen, I was afforded the opportunity to participate in a ridealong inside a real NASCAR automobile. In this venture, I rode for three laps around Chicagoland Speedway at speeds close to 150mph. In my program, I aim to capture a glimpse of not only the race itself, but also the exhilirating experience of riding in a racecar.

I have decided to use the Indianapolis Motor Speedway as the basis for my program's location. This course is one of the most iconic and historic tracks in racing; its construction dates back to 1909, before NASCAR was even founded.

2 Project Description

My original objective was for the user to switch between two cameras to yield two perspectives from which to view an automobile race. The first viewpoint is an overhead view, which depicts the car as it progresses around a fixed course. This was accomplished by first establishing a racetrack frame called "ftrack" and then constructing a 3D car frame called "fcar" which rotates around the axis of the track frame. The car rotates with respect to the track as it enters corners so that it is always pointed in the direction it is travelling.

The other camera is an in-car point of view, in which the user is simulated as the driver. This view was intended to depict a still image of the dashboard of a vehicle with a moving image inside a windshield. However, it was a large enough undertaking in itself just to move the camera. As such, the final version of the RTICA merely depicts the view from the car without the accompanying details. Even so, the landscape shown to the user adjusts based on the racecars position along the track. This was originally intended to be accomplished by texturing mapping an image of a racetrack to a cylinder on the inside. I was unable to successfully texture map an image of a racetrack onto a cylinder, but with the help of the professor the final code includes the image mapped onto a square landscape.

---

3 The Mathematics of Racing

3.1 Kinetic Energy and Inertia

The concepts of kinetic energy and inertia are pertinent in the analysis of automobile racing. First, we will discuss the basics of kinetic energy and inertia, later describing their relevance to racing. Kinetic energy is dependent on two components: mass and velocity. In physics, kinetic energy can be calculated with the following equation:

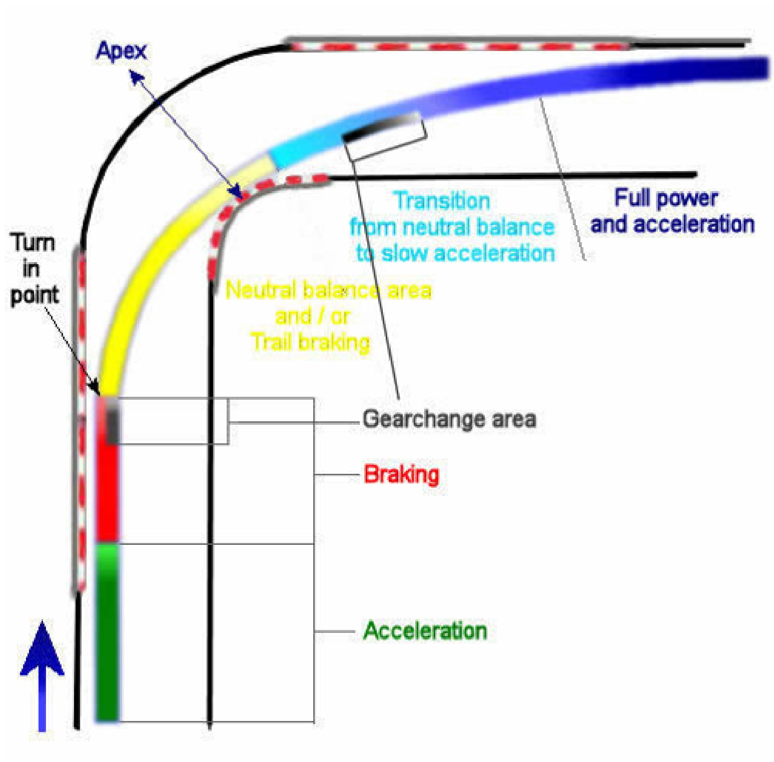
$$KE = \frac{1}{2} * m * v^2$$

Automobiles racing around a racetrack have a lot of mass as well as a lot of velocity, thus giving racecars a high amount of kinetic energy. In addition to energy, racecars also have a lot of inertia. Inertia is the ability of an object to resist a change, and the larger an object the more inertia it has. The importance of these two concepts is manifested in the vehicle's ability to maneuver the turns of a racetrack. At high speed with a lot of mass, the car will have a tendency to keep moving in a straight path. However, in order to be competitive in a race, the car needs to be able to maintain motion throughout a turn to be able to exit with as much speed as possible. As such, the idea of cornering arises, a concept that will be discussed in the next section.

3.2 Cornering

In racing, cornering refers to the process by which a racecar progresses throughout a turn. In NASCAR, there are four turns for drivers to maneuver on the typical superspeedway. The exceptions to this are road courses (such as Watkins Glen and Infineon) as well as the Pocono Raceway in Pennsylvania, sometimes referred to as the "Tricky Triangle", which only has three turns. Cornering is key to gaining a competitive edge against other drivers; cars are subject to many restrictions which makes it difficult to gain an edge through other means. Whereas in street racing teams can modify the engine to produce more power, most of the discrepancy between drivers' lap times in NASCAR comes from their ability to maneuver the corners as efficiently as possible.

The hardest part to cornering on a racetrack is maintaining a balance to braking and acceleration while finding the best line of travel. [1] As mentioned earlier, racecars have a lot of inertia which diminishes a driver's ability to control the car through a sharp turn. It is not possible to traverse a corner with the same speed as a straight stretch of raceway without sliding out of the turn. Therefore, drivers must slow down while entering a turn so that the car does not run up into the wall. The other issue with speeding through a turn is grip. Tires on a racecar help stick the vehicle to the racetrack and aid its motion through a turn. However, friction from the tires can only do so much at high speeds, which is why superspeedways enlist the help of banked curves to keep racecars moving in circular motion.



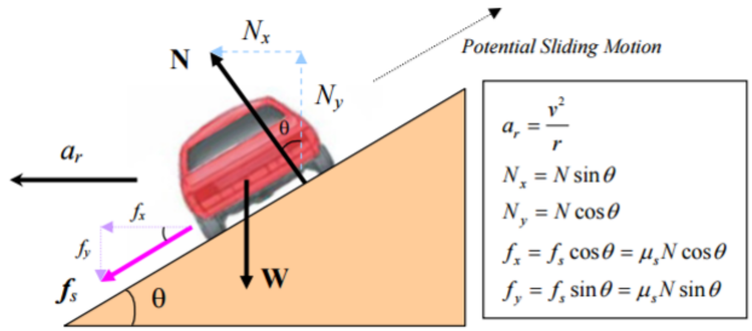
Above is a graphic to demonstrate ideal cornering technique. [2] For an object moving in a circle, a force is necessary to keep it in circular motion:

$$F_{centripetal} = \frac{m * v^2}{r}$$

In this equation, m refers to mass, v refers to velocity, and r refers to the radius of the curve. However, even following this technique for cornering still may not be enough to hold the car in the turn. The nature of superspeedway racing requires an additional measure to aid drivers in cornering on the racetrack. The next section will briefly discuss the physics behind banked curves and their part in racing.

3.3 Banked Curves

Banked curves are crucial to drivers for holding the vehicle in the turn. Due to the high speed of NASCAR racing, friction from the tires coupled with swift cornering technique simply is not enough to prevent the car from crashing into the wall. Banked curves make use of the force of gravity to help guide the racecar around the corner. [3] What follows is a diagram to demonstrate this concept.



In order to accomplish the centripetal acceleration, a_r, needed to keep the car moving around the turn, banked curves make use of the normal force to complement the friction provided by the tires. In this diagram, the normal force N is the reactionary force to the force of gravity from the weight of the car, W. Specifically, the component of normal force that contributes to the centripetal acceleration is denoted N_x. This force can be calculated with the following equation:

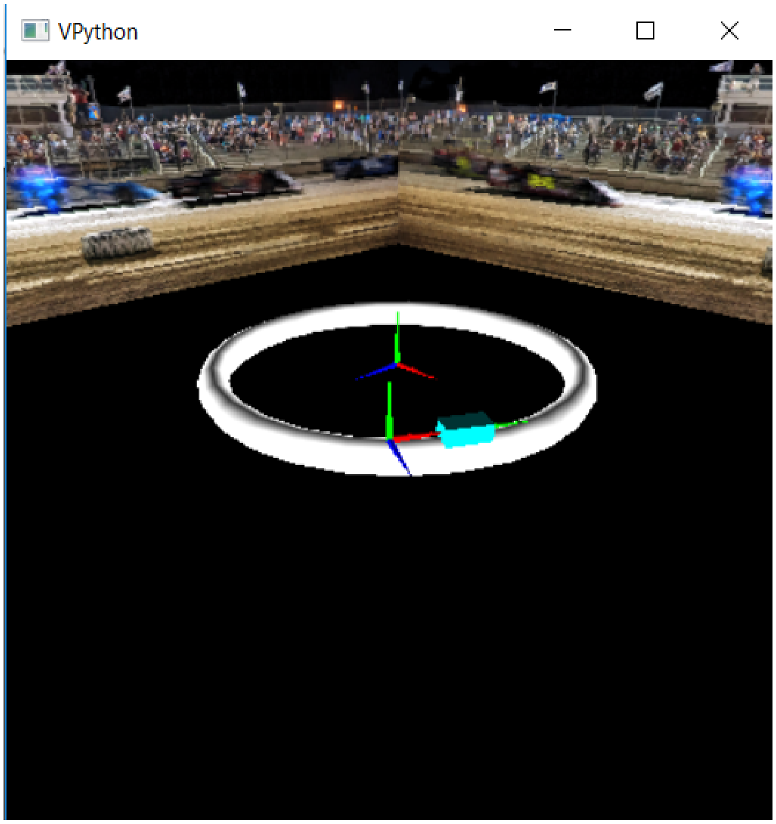
$$N_x = N * sin(theta)$$

where *theta* represents the angle between the surface of the earth and the inclined racetrack. When added, the force of friction from the tires combines with the normal force provided by the racetrack to guide the vehicle through a circlular motion around the turn.

---

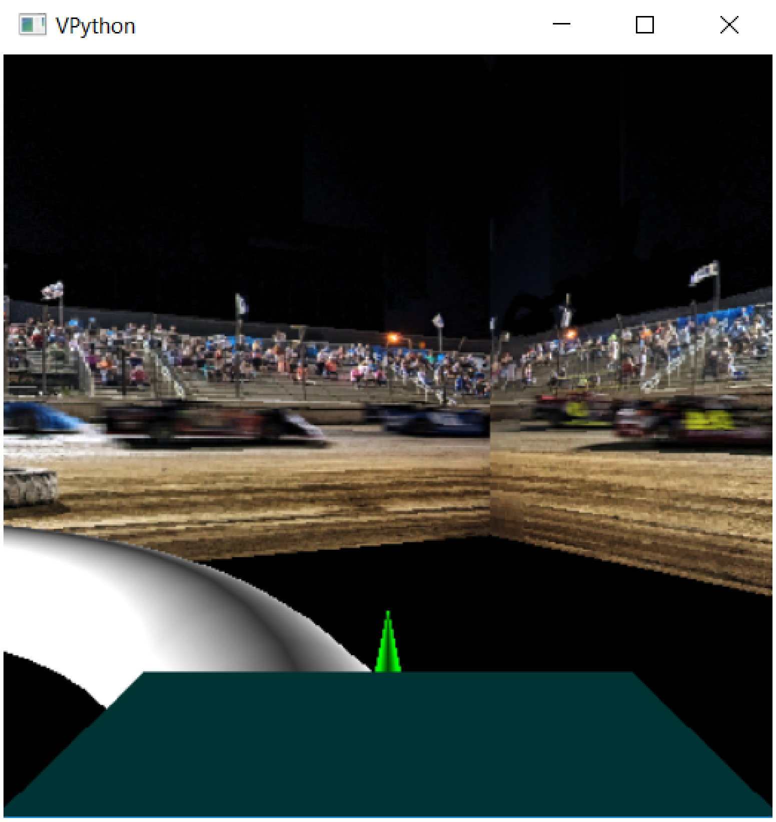
4 Explanation of Code and Final Program

The final version of the RTICA makes use of a number of techniques to achieve the goal of creating an in-car perspective. For starters, the program uses frames to define the main objects within the scene. Specifically, there is a track frame named ftrack as well as a car frame named fcar. The animation of the car moving around the track is achieved by rotating fcar around the axis of ftrack to simulate circular motion around the track. The track itself is a white colored ring object while the car is a blue colored box. In addition to the frames, the code also makes use of a texture wrapped image to construct a landscape around the racetrack. The landscape consists of four texture-wrapped Targa files that form a cube around the scene. Below is a screenshot of what this looks like in the final program.



In order to organize the code, the desired features were compiled into a series of functions which could be called on demand. For instance, the creation of the main objects within the scene are grouped under the function makefcar. There are other functions that each perform their own task, one of which is called carcam which contains the code relevant to positioning the camera on top of the car. I learned that this method of organization is helpful in testing particular portions of the code; instead of commenting out entire sections of code at a time, I could simply call a function to test a specific feature.

The largest undertaking was the manipulation of the in-car camera. My greatest progress came from gaining an understanding of the VPython scene as well as its related functions. Specifically, this code makes use of three core scene functions to achieve the in-car perspective. [4] First, I define scene.center to be the position of the car. The center of the scene can be thought of as the origin of the world. This is important because of the way that another function works: scene.forward. This scene function is defined as a vector which constantly points toward the center of the scene. Moving the center of the scene is what effects the perceived movement of the camera. The most difficult part of moving the camera was finding the best way to manipulate these functions to point toward the front of the racecar. This was done by defining scene.forward to be the axis of the car. That way, the scene center is located on the car and scene.forward always points in the direction the car is traveling. Another scene function, scene.range, has the effect of zooming in an appropriate amount to place the camera on top of the car. Otherwise, the camera is able to see the whole car, which does not achieve the desired effect.



Above is a screenshot of the view from inside the car. Although it is not perfect, the view that is simulated is still an accomplishment considering the difficulty in the way that the VPython scene is defined. If more time was spent on the program, one could continue to make improvements to the in-car cam as well as include features to simulate a dashboard and controls.

5 Bibliography

[1] "Corners and Cornering." *Technical F1 Dictionary.* Technical F1 Dictionary, n.d. Web. 13 Dec. 2016.

[2] "Cornering Technique." *Technical F1 Dictionary.* Technical F1 Dictionary, n.d. Web. 13 Dec. 2016.

[3] Stanbrough, JL. "A Banked Turn With Friction." *Physics at BHS.* JL Stanbrough, 6 Feb. 2006. Web. 13 Dec. 2016.

[4] "VPython Help." *VPython.org.* N.p., n.d. Web. 13 Dec. 2016.