Dylan Li

COS 426

Dec. 16, 2022

Web Swing

Abstract – Web Swing is a fully functioning 3D game created using Three.js that is inspired by the titular swinging mechanic that is famous in Spider-Man movies and video games. In Web Swing, the player controls a hero character that is attempting to reach a clock tower in the distance. However, there is a city of buildings in the path that the hero must avoid or restart a level. In addition, falling to the ground will likewise restart a level. Once the player reaches the clocktower, the game will advance to the next level, restarting with more building to increase the difficulty. The player relies on the swinging mechanic that moves the hero through the air to avoid the obstacles reach the clocktower. Swinging always goes forward in either the left, straight, or right direction in order to maintain a smooth motion transitioning between swings. Changing direction allows the player to avoid buildings in their path. In this manner, Web Swing is able to emulate the satisfying swinging mechanic of Spider-Man while making the mechanic the core feature of the gameplay.

Introduction

The goal of this project was to reproduce the web swing mechanic seen in Spider-Man media and use it as both the method of travel and the key to the win-loss condition of the game. Because of how integral the mechanic would be to the overall gameplay, it was important to narrow down a method that would be achievable and still reasonably realistic. There is a gauntlet of movies and video games that have enacted, reproduced, and innovated the mechanic. While none are physically realistic, there is a spectrum of realism maintained throughout the years.

Thus, there were multiple features and elements to pull from in creating a custom web swing mechanic.

On one end of the spectrum, Spider-Man movies have generally attempted to maintain a sense of reality in a fantasy superhero story. Think of *The Amazing Spider-Man* scene where Andrew Garfield's character is wounded and must get to a building in the distance. He uses a series of conveniently placed cranes in the city as pivot points to carefully swing to his destination. Because of the injury, the act of web swinging, which is usually automatic or natural, became a difficult procedure for Spider-Man. In turn, the director slowed down this scene when depicting web swinging, making realism necessary as more focus would be turned on it. The mechanics of web swinging become apparent then: it is a pendulum where the web connection is the pivot and Spider-Man is the weight. Spider-Man himself can impart additional energy into the system upon release of the web, allowing him to more easily reach his destination.

In contrast, Spider-Man video games are not bound by the same priorities of the movies in catering the web swing mechanic for their audience. For example, a recent and acclaimed iteration, *Spider-Man: Miles Morales,* uses web swinging as a high-powered and convenient mode of travel. The web connection is always off screen and is not a real connection. In some cases, it is suggested by the direction of the web that it is connected to some tall object that clearly does not exist. In addition, the player is able to boost Spider-Man forward with a mysterious electric force. As an added convenience, the player can also shift Spider-Man's position and direction mid-air without any physical backing. Since the game is open world and web swinging is the main mode of travel, it is clear that convenience and speed were priorities in

this version of the mechanic. However, the physical meaning behind the web swing is lost as a result.

Since this project uses the web swing mechanic as more than a traveling mechanism, it would be improper to emulate all of the convenience factors that Spider-Man games have introduced. This would cause the game's win condition to be too easy. However, trying to completely create a realistic web swing mechanic would lose the basic convenience that is necessary for smooth gameplay. Therefore, it was decided to combine one main feature from each of the movie and video-game ends of the spectrum realism. First, the swinging part of the mechanism would obey the motion of a physical pendulum with no external forces. Second, the pivot point is abstracted away for convenience off screen so that the player does not have to worry about a web connection.

Methodology

There were several large components in the implementation of the full project. First and most important was the web swing mechanic. Then, the camera's movement in relation to the hero had to be figured out. The character design of the hero was also an important component. Afterwards, the city environment was created. The city included the buildings designed as obstacles, the ground, streetlights, clouds, and the clocktower, the end destination.

In order to implement the web swing mechanic, the location of the pivot point had to be placed. The pivot was determined by the constant web length, constant angle with respect to the x-z plane, and an angle of rotation about the y-axis for the left, right, and forward directions. Then, the position of the pivot relative to the hero could be calculated right after the web is shot. In order to simulate the movement of the hero, a physics method similar to that of the Cloth Simulator in Assignment 5 was used. The scene is incremented by a constant timestep at each

frame and the hero's position is then updated via Verlet integration to generate the next frame. Only three forces were necessary: gravity, tension, and a release force. Gravity is always present, tension is only applied in a web swing, and the release force is only applied instantaneously after a swing is ended. In updating the scene, the surrounding environment is actually moved relative to the hero so that the hero is always at the origin in world coordinates.

There were a few complications with the swinging motion and the gameplay. The web is meant to be connected at a pivot with a building while it is in practice connected with a point in space. It would not be feasible to complete a full revolution about a building, but it is possible to do so with a point. Thus, the motion of the swing is released automatically after the cross product of the velocity and the direction of the pivot crosses the original plane of motion. In practice, the motion is stopped when swinging forward before the hero swings backwards and is stopped when swinging to the side at a quarter of a revolution.

Another important component is the camera's position. It is always positioned to keep the hero at the center of the screen but moves to generally point in the direction of the hero's velocity. Since the hero's previous position, or offset, is always stored for Verlet integration, it is simple to ascertain the direction of the velocity. However, the camera should also move smoothly, where if the player changes direction, the view should not jarringly change to reflect the new velocity, but gradually shift to the new direction. To accommodate this, the camera's position is changed via rotation that is capped at a maximum value. In addition, because of the varying initial velocities going into a swing, if for example the player chooses to swing left, the hero could initially move to the right first in the elliptical path of the pendulum. This detracted from the otherwise smooth motion. Thus, an additional check of the direction of the rotation and the direction of the swing had to be consistent in order to rotate the camera.

The character design of the hero consists of Three.js box geometries. This was chosen over an imported mesh to allow for simplistic manipulation of the hero to hold onto the web and to be at rest. When the hero is in a web swing, the arms are raised and rotated to appear to grasp onto the end of the web. Otherwise, the arms are at the side of the body. The hero is also rotated to be in line with the direction of the pivot and to face direction of the velocity.

The next large component of the implementation was generating the city environment. To increase efficiency, the city is predetermined at each level as an array of grids. At each grid, there can be a default building, a streetlight, the starting position, or the destination clocktower. Imported textures and models were used for the default building, streetlights, and the clocktower. Collision detection was able to be efficiently implemented by determining the hero's position in the grid from the world coordinates, having to check collisions with four buildings at most. The streetlights did not all have lights as this would be too much calculation to render. To compromise, only the streetlight the hero is directly over is lit with Three.js point lights. These streetlights acted as a guide to the player to indicate the hero is near the ground. The clouds act as an additional guide at the height of the tallest default building to indicate the hero cannot shoot a web at that height. The clouds are randomly scattered across the city. Finally, the default buildings are generated with constant probability in columns that linearly decreases from the center to the edge of the city. This discourages the player from going along the easiest direct route by blocking it.

Results

The success of the game was measured qualitatively in how well it simulated the web swing mechanic and how well it followed the intended behavior. The web swinging turned out to be smooth and was able to advance the hero forward in a convenient fashion. However, the

simplified controls detracted from the realism. Because the player was always able to shoot a web in all three directions, the connection of the web to a building was not taken into account. This breaks the illusion of the web swinging at times when the hero is not surrounded by buildings. In terms of the intended behavior of the game, the only part that did not work one hundred percent of the time was the rotation of the hero to face the direction of their velocity. When the hero takes a large swing to the left or right, the hero will end up sideways. However, this did not affect the gameplay.

Discussion

One component of the game that be further developed is the city. Currently, the city is limited by its predetermined size and a random placement of buildings. This results in the later levels where the probability of a building placement is higher to be near impossible. An improvement upon the game could be to either have predetermined levels with buildings placed strategically, or to keep a probabilistic approach that can guarantee paths from the start to the end point.

In addition, the city as well as the hero could be aesthetically improved. Including more variety in buildings, streetlights, and possibly other components would support the realism of the game. Adding a model with animations would also improve the hero character.

Conclusion

The biggest achievement of this game is the smooth web swing mechanic that reached the goals for its design. This hinged on the physics and the camera movement, which both proved to be seamless in all cases. Overall, this project helped me learn about 3D game design, especially with Three.js and JavaScript. Further steps would be to port the game over to a better performing system that can better handle larger components.

Works Cited

The Amazing Spider-Man Crane Scene: https://www.youtube.com/watch?v=jhpax7lcLYs

Spider-Man Miles Morales Web Swinging: https://www.youtube.com/watch?v=6Lw_66C76Vo