

# COS 426 Report: PrinceTron Legacy

Kevin Finch, Nebil Ibrahim, Catherine Nguyen, Alice Xue

May 2020

## Abstract

*PrinceTron Legacy* is a 3D reimagining of the lightcycles game *Tron Legacy*. Two players use the same screen and control motorcycles that leave colored trails behind them. The goal of the game is to survive the longest without intersecting your own trail, your opponent's trail, or the borders of the game board[3].

## Introduction

In *PrinceTron Legacy*, we created a game inspired by *Tron Legacy*. The game includes two players on motorcycles. As the players move, their motorcycles leave a light trail behind. If they touch their own trail or the opponent's trail, they will be killed. Likewise, if they touch the walls of the game board, they will also die. The goal of the game is to survive longer than your opponent.

We were inspired by *Tron Legacy* because of the strength of the graphics of the game. With our emphasis on minimalist aesthetics and intuitive controls, we hoped to bring something new to the limited Tron-based games available on the internet. *PrinceTron Legacy* differs from *Tron Legacy* because we allow the motorcycles' turns to be more continuous and fluid. The turns are more similar to the game "Slither.io" [1]. This approach requires a new way to represent the trails that follow the motorcycles because they are no longer turning at a 90 degree angle. The trail will have to move smoothly with the motorcycles as it turns. By allowing players to not be limited by 90 degree turns, they have more control over their bike and movement during the game.

## Methodology

First, we added two motorcycles to the scene. After this, we created a trail following the motorcycle. This was done by including a flat cube each time the motorcycles moved. The scene of the game was designed so that there is a stage where the two motorcycles will compete. The stage is surrounded by orange walls and the background color is black. The stage also has a glow of

blue light radiating underneath it [2]. This creates a competitive and suspenseful atmosphere for the players to compete. We utilized the UnrealBloomPass from three.js to make the game board and motorcycles glow. We chose to be minimalistic in our approach because we believed simplicity would allow the game to be easier to understand and enjoy.

In order for the motorcycles to move, event listeners were added to the keys to let players turn to the left and right. Player 1 uses the left and right arrow keys to move their motorcycle while Player 2 uses the “a” and “d” letter keys to move theirs. Both motorcycles will go straight by default. They can change their directions by pressing their left and right keys. At first, the motorcycles were turning at a 20 degree angle. When we did this, however, we thought the game would be more fun if players had more control over their turns. For this reason, we changed the turning angle to be 5 degrees. This allows players to have more control over how much they can turn and move.

In this game, players lose when they collide with their own motorcycle’s trail, the opponent’s trail, or the wall. To do this, we checked if the player’s bounding box collided with the trails’ meshes. If it collided, the player would lose. We designed the game so that the player would die when they crash into the wall instead of it bouncing back. We believe this makes the game more challenging for players, which will allow it to be more engaging.

Before the players start the game, there is a page with the directions explaining the rules of the game. When both players are ready, they can click begin. A count down will appear on the screen to notify when the game will start. If any of the player collides with a path or a wall, the game will end. A message will pop up notifying which player won. They will have an option to play again if they want.

## Results

The end product created is a simple web browser game that two people can participate in. The initial screen displays the instruction to the game. The players will then click begin to start playing. Player 1 can use the left and right arrow keys while Player 2 can use the ‘a’ and ‘d’ keys to move. When one of the players lose, there will be a black screen stating who won.

The most important metric of success for our game was the frame rate. Since we limited the length of the trail, the game is less laggy than previously. The frames per second (fps) fluctuate a bit but it mainly stays at around 60 fps. This was measured by recording the time at the beginning and end of the animation frame handler and getting the reciprocal of the elapsed time. Collision detection is consistent, but it can happen preemptively. Since collisions are detected with a bounding box on the vehicle, the collisions appear to happen just before the bike would actually touch a trail or while a bike is parallel to a trail. This approach was more efficient than checking to see if each cube in the trail was intersecting with the bike itself by using the points which define the mesh.

## Discussion

### Mechanics

In order to improve performance of the game, the main mechanic to change would have to be how trails are represented. Since they are currently represented as several trailing cubes, the cubes can slow down the game. This problem is mitigated by restricting how long the trails can be. For older computers, however, there is still the possibility that this will slow the game down. An ideal solution would be if there was a simple and effective method for expanding a mesh to be the trail in addition to a quick way to calculate collisions.

### Future Works

Adding sockets to PrinceTron Legacy would allow for changes to the game not otherwise possible. First, it would allow each player to have their own camera without the necessity of a split screen. Having a camera follow the player would enable players to have a more 3D experience. The new camera angle will also make the game more challenging. This is because the player may need to shift their strategy on how they would avoid colliding into the motorcycle paths and walls. Since there will only be a camera following their path, they may need to retain a mental map of what they cannot see. A shift in camera angle would also permit more movement options, such as jumping. These movements are difficult for players to use and have when there is a static camera angle.

Sockets would also make a multiplayer experience possible. There can be multiple players on different devices who can enjoy the game together. The same rules would still apply but instead of having just one opponent, players would continue until there is only one player left.

In addition to sockets, including items that occur throughout the gameboard can enhance the game. The items can possibly give the player special powers that increases their speed or trail length for a certain time period. Furthermore, the game can be enriched by presenting more vehicle options for the players to choose from. Each vehicle can have different properties with varying strengths and weaknesses. The incorporation of more obstacles and stage hazards can make the game more dynamic.

### Visuals

PrinceTron Legacy is inspired by the movie Tron Legacy. Thus, we believed the minimalistic design would best capture Tron's virtual environment. In the movie, Tron's main colors consist of blue and orange. Everything else is black. The colors of the bikes and buildings pop out visually because they are glowing in the black atmosphere. This is what we recreated in our game. Our game's design is easy on the eye and sets a mysterious and futuristic mood. The game board has an orange boarder. The two motorcycles glow and leave behind blue and pink trails respectively. We had to play around with the code to see what

would look best. One of the challenges we faced was figuring out what to add to the game board to make it more complicated but also remain simple. We played around with different shapes and size.

The visuals can be extended by adding some form of terrain to the game board. This will allow the bikes to interact more with their surrounds and provide more friction to the players path. This approach, however, will diverge from Tron Legacy's visuals. It will stray away from the motorcycles going quickly and smoothly. It might add depth perception to the game though.

## Conclusion

PrinceTron Legacy functions smoothly and achieves the goals we set. Two players are able to go to the website and begin playing with ease. They are able to use the keyboard to have a interactive game on the browser. An aspect of the game that can be improved in the future is making it more dynamic. The next steps that can be taken to do this is by adding more features, like items that give players special powers and multiple car options with different properties, to the game including different types of terrain the players can choose from can increase the engagement between the players. Different terrains can vary in the level of difficulty, which can help keep the players excited about the game. It will also allow different age groups to participate in the game. Depending on the players' abilities, the level of difficulty can be altered so that the game is not too easy or too hard. In addition to new features, PrinceTron Legacy can be extended with sockets. Sockets will let more players participate in one game which can increase player engagement.

An issue that may need to be revisited is developing a more efficient solution to detect collisions and render bike trails. This is necessary because in order to implement more features and allow more players, there cannot be a significant compromise to the fps of the game. If this problem is not solved, the new features would exacerbate the issue with the collision detection and fps.

## Contributions

Kevin Finch worked on putting together the front end to the backend, made the code more efficient so it was less laggy, and helped refined the features of the game. Nebil Ibrahim worked on the data structures for the setup of the game. Alice Xue and Catherine Nguyen both worked the visuals and overall aesthetics of the game. Alice Xue worked on the collisions and movement of the motorcycles.

Special thanks to June Ho Park for advising us in this project as well as the other TAs of COS 426 for general advice.

## References

- [1] Slither. <https://slither.io>.
- [2] Gradients in three.js, March 2016. <https://darrendev.blogspot.com/2016/03/gradients-in-threejs.html>.
- [3] Catherine Nguyen Alice Xue Kevin Finch, Nebil Ibrahim. Princetron legacy, May 2020. [https://github.com/KevinMFinch/Cos426\\_final](https://github.com/KevinMFinch/Cos426_final).