FINAL PROJECT REPORT

Problem

The central challenge motivating the development of "**Touchdown Maze Adventure**" was the perceived limitations in existing maze games. Traditional maze games often lack depth and engagement, with static environments and predictable gameplay. My objective was to address this shortfall by crafting an immersive gaming experience that seamlessly integrates the excitement of American football with the intricacy of navigating a dynamic 3D maze.

Existing maze games lack realism and dynamism. I aimed to break away from this trend by incorporating advanced features, such as dynamic vehicle control and realistic physics simulations. The challenge was to strike a balance between the immersive nature of the game and the playability that users expect. We sought to create a game that not only challenges players with maze navigation but also provides a visually stunning and interactive environment.

In summary, the problem I set out to solve was the need for a maze game that transcends the conventional, offering players a novel experience through the fusion of Georgia football theme and dynamic gameplay.

Approach

- Choice of Elements:

The project commenced with a careful consideration of key elements. The Bulldog vehicle was chosen as the player's means of navigation, introducing an engaging and dynamic aspect to maze exploration. To efficiently & succinctly present a 3D modelled maze, a glb formatted maze structure was imported in the project. This choice ensured a balance between visual fidelity and optimal performance. The inclusion of a touchdown point added a goal-oriented dynamic, aligning the game with the American football theme.

- Integration of Three.js:

To achieve visually stunning graphics and an immersive 3D environment, Three.js was used as the foundational library. Leveraging its capabilities, the project successfully rendered realistic textures on maze structures and floor elements. Responsive camera controls were implemented to provide players with an engaging perspective as they navigated the maze. The seamless integration of Three.js played a pivotal role in setting the visual tone & captivating user experience.

- Maze Structure:

The maze structure presented in the project is basically a glb file imported in the JavaScript environment using GLTFLoader. The maze was scaled & positioned accordingly to fit the arena for a prefect gameplay experience. Three.js in-built fog element was used to provide depth & realism to the maze. An integral part of building maze structure was the application of Physics properties to the maze by using Cannon.js. Cannon.js is a physics engine for JavaScript, specifically designed for 3D physics simulations. Initially, I wanted to create a dynamic methodology which can we used to apply physics properties like collision detection to any

desired maze of my choice at runtime. However, the limitations of working with an imported glb made it difficult. This basically limited my choice of using different randomly selected maze structures for every new game instance. Cannon.js basically uses a bounding box algorithm to associate collision properties to elements in a 3D scene. With the help of cannon-debugger, I was able to associate collision detection properties to all the walls in the maze structure. This process was quite meticulous & involved mathematical calculations. Notably, the use of half-extents for specifying box dimensions in cannon was quite a surprise, on research I realized it helped in simplifying calculations.

- Implementation of Vehicle Mechanics:

Central to the project was the implementation of dynamic control for the Bulldog vehicle. A combination of box & sphere geometries was used to generate the visibly realistic structure of Bulldog vehicle. The hinge constraint feature of Cannon.js enabled joining of wheels & the body of the Bulldog vehicle. This hinge constraint on the wheels provided a motor which facilitated rotational feature providing the required mobility to the vehicle. Forward & sideways thrusting was used to allow movement in all directions. Realistic driving experiences were achieved through meticulous adjustments to acceleration, braking, and steering mechanisms. Finally, to facilitate user inputs for vehicle control, multiple event listeners were associated with keyboard-based interactions. Integration with Cannon.js facilitated accurate collision detection, preventing the vehicle from unrealistically passing through maze structures. This dynamic interaction enhanced the overall gameplay experience.

- Texturing and Wave Motion:

Texture was applied to the floor elements at both the beginning and end points of the maze, enhancing visual appeal, and adding a layer of realism to the maze environment. The introduction of a dynamic wave motion for the flag further elevated the immersive quality of the game. The flag's movement was determined by a sine wave that depends on the initial positions of the flag vertices and the elapsed time. Texturing on the flag further enhanced its realism. These visual enhancements were crucial in creating a vibrant and engaging atmosphere, enhancing the overall aesthetic experience for the player.

In summary, the approach to developing "Touchdown Maze Adventure" was characterized by a meticulous selection of elements, the strategic integration of Three.js for advanced graphics, the implementation of realistic vehicle mechanics, and the addition of visual elements for a more immersive gameplay experience.

Experiments & Results

I conducted experiments to test the gameplay mechanics, visual elements, and collision detection accuracy. Results showed that the integration of Three.js and Cannon.js significantly improved the overall gaming experience. The gameplay was successful in providing responsive controls, realistic vehicle dynamics, and visually appealing textures. Cannon-debugger played a pivotal role in managing the collision detection boxes. Collision detection turned out robust, preventing the vehicle from passing through maze walls. The dynamic wave motion for the flag added a touch of realism to the environment.

Reflection

- Successes:

The approach taken in developing "Touchdown Maze Adventure" yielded quite a success. The implementation of dynamic vehicle control proved to be a key highlight, providing players with a realistic and responsive driving experience. The integration of Cannon.js for collision detection was particularly successful, ensuring that the Bulldog vehicle interacted seamlessly with maze structures. This accurate collision response added a layer of realism and challenge to the gameplay, preventing unrealistic penetrations through walls.

The visual elements, including texturing and the dynamic wave motion for the flag, were successful in enhancing the overall aesthetic appeal of the game. Realistic textures added depth, creating a visually engaging environment. The dynamic wave motion added a subtle but impactful layer of dynamism, contributing to the immersive quality of the game world.

Responsive camera controls provided an engaging perspective, allowing users to navigate the maze with ease and enhancing the overall user experience.

- Challenges and Learnings:

Despite the successes, the project was not without its challenges. Dynamic application of physics properties to an abstract imported element was the biggest hurdle. Fine-tuning the balance between realism and playability in vehicle mechanics required iterative adjustments. Striking the right sensitivity in thrusting, braking, and steering proved to be a delicate process. However, this challenge also provided valuable insights into the complexities of creating a realistic driving experience within a virtual environment.
The integration of dynamic elements, such as the waving flag, presented challenges in achieving the desired visual effects without compromising performance. Iterative refinement and optimization were necessary to strike a balance between realism and maintaining a smooth gaming experience.

Future Enhancements:

Looking ahead, the successes and challenges experienced during the project have laid the foundation for exciting future enhancements. The exploration of multiplayer support is a natural progression, adding a competitive edge to maze navigation. Expanding maze variations and touchdown scenarios would provide diversity to the game. The introduction of scoring mechanisms and difficulty levels is a logical next step, adding gamification elements to elevate the overall experience.
Continued development beyond the class is a definite consideration. Future iterations could explore additional visual enhancements, new maze structures, and advanced gameplay features to keep the gaming experience fresh and captivating.

a) Relevance to the Class:

The project directly aligns with the objectives of the class by implementing advanced concepts from Three.js and Cannon.js. It demonstrates the application of 3D graphics rendering, physics simulations, and interactive game development, showcasing a practical understanding of the course material.

b) Level of Effort:
The project required a significant level of effort, equivalent to 1.75 times that of Project 3. Implementing dynamic vehicle control, responsive camera controls, and realistic physics demanded thorough research & hard work.

c) Time Investment:
The project was initiated right before the thanksgiving break, and approximately 70 hours were invested in its development. The extended effort reflects the complexity of implementing advanced graphics and physics simulations.

In conclusion, "Touchdown Maze Adventure" successfully addressed the problem of creating an engaging maze game with realistic physics and dynamic gameplay. The application of Three.js and Cannon.js, coupled with careful fine-tuning, resulted in a project that not only meets class objectives but also lays the groundwork for future enhancements and continued development.

References:

- **https://threejs.org**
- **https://3dexport.com**
- **https://www.freecodecamp.org/news/three-js-tutorial/**
- **https://pierfrancesco-soffritti.medium.com/how-to-organize-the-structure-of-a-three-js-project-77649f58fa3f**

EXECUTION INSTRUCTIONS:-

NPM & WEBPACK BUNDLER WERE USED FOR DEVELOPMENT EASE.

Use the following script to execute:- (needs npm package manager installed)

- npm run bulldogs

if encountered any errors execute the following script to install correct dependency packages

- npm i

DEMO LINK:-

- https://youtu.be/E_ne_KXlOBA