Seminar for the subject Computer Graphics

Fast Parking

Group members:

Anastasija Djajkovska, Bojan Spasovski

ad9274@student.uni-lj.si

bs4567@student.uni-lj.si

Ljubljana 2024

Introduction

Embark on an exciting snowy adventure, navigating icy terrain and collecting cash for parking in this thrilling game. The challenge lies in mastering the art of icy driving to swiftly gather funds before parking. Combining racing, strategy, and precision driving, the game falls into the action racing genre. Developed with WebGL and JavaScript, it delivers a seamless and immersive gaming experience.

Will you conquer the icy roads and park your car with the maximum amount of cash in the shortest time possible?

1. Game Overview

The game falls into the action racing genre, combining elements of strategy and precision driving. It offers an easy level of difficulty suitable for all sorts of players. Targeted at a broad audience, the game is designed for those who enjoy the thrill of maneuvering a car in challenging conditions. Set in a snowy landscape, the game's scenario involves a driver racing against time, navigating on icy terrain, to secure funds before safely parking the car.

1.1. Game Overview

1.1.1 Overview

The game unfolds in a snowy environment, providing players with challenging terrain to navigate. The world is designed with a cartoonish, yet realistic aesthetic. Players interact with the game world in a 3D space, moving through a racetrack set in a snowy landscape.

1.1.2 Background

The background of the game features distant elements that contribute to the immersive snowy landscape, like snow trees and houses, as well as a night sky that enhances the overall visual experience.

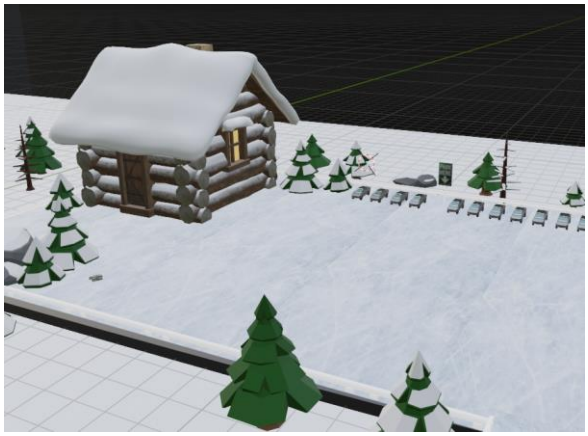

Image 1: Parking

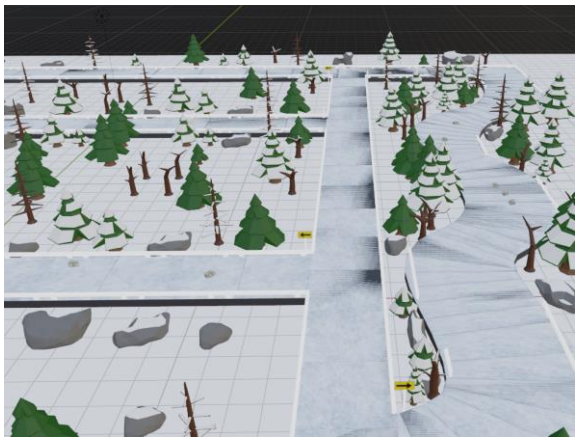

Image 2: Icy racetrack

1.1.3 Key Locations

Key locations include the start of the racetrack and the parking lot where the driver needs to park the car.

1.1.4 Size

The game world is designed at a small to moderate size, focusing on one large racetrack. The primary viewpoint encompasses the immediate surroundings visible by the light.

1.1.5 Objects

Players will encounter and collect scattered cash in the form of in-game currency. The focus is on gathering these resources to accumulate wealth for parking. The in-game currency and objects are strategically placed, requiring skillful navigation to acquire, because of the slippery racetrack.

Notably, all the objects, including the character, the money and the racetrack, have been crafted and designed using Blender, ensuring a bespoke and visually appealing experience for players as they engage with and collect these assets.

Some of the assets were made by us, notably the racetrack which was created using Bezier curves. The dynamic and fluid design of the track, with its twists and turns, has been meticulously shaped and perfected through Bezier curves. This technique allows for a seamless integration of curves, offering an engaging and visually appealing race track that challenges players with its intricacies. The utilization of Bezier curves not only adds a layer of sophistication to the track's layout but also contributes to the overall smoothness and aesthetic appeal of the in-game environment. This was done in Blender.

Some of the assets we took from free online sources, those were different kinds of snowy trees, rocks in different sizes, car (inspired by the tesla cybertruck), highway railings, sings and barriers, houses, money, etc.

1.1.6 Time

Time in the game progresses in real-time, mirroring our world's clock. The timer starts at 0, and the fastest completion time is recorded as the player's score. This adds a competitive edge as players aim to complete the mission in the shortest possible time.

1.2. Game Engine and Technologies Used

In the development of our game, we utilized WebGL and JavaScript as the primary technologies. WebGL provides a powerful framework for rendering 3D graphics in web browsers, while JavaScript served as the scripting language to implement game logic and interactivity.

1.3. View and Camera

The game employs a third-person perspective to offer players a comprehensive view of the snowy landscape and their car. The camera is strategically positioned to follow the car's movements, providing a dynamic and engaging visual experience.

The *FirstPersonController* class manages the camera's behavior in the game. It dynamically follows the player-controlled car, offering an engaging view of the snowy landscape.

Here's an overview of the camera implementation: The update method governs camera movement and rotation, adjusting the car's velocity based on user input (W and S keys). The camera's position is updated relative to the car, and mouse movements control yaw and pitch. The implementation ensures a responsive and immersive camera experience, enhancing the player's control and navigation through the icy terrain.

The user interface (UI) elements are carefully designed to be intuitive and unobtrusive, ensuring that players can easily monitor essential information such as time, collected cash, and any in-game prompts.

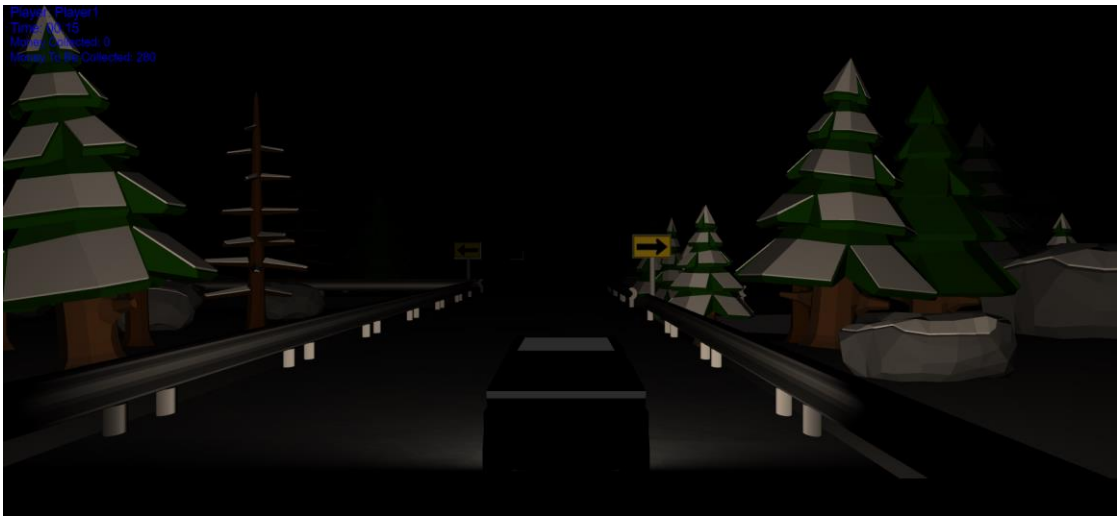

Image 3: The position of the camera

2. Characters

In our game, the primary character is the player-controlled car. Users have complete control over the car's movements, including acceleration, deceleration, turning left, and turning right.

The primary action the character can take involves collecting scattered cash on the slippery ice. Once all the cash is collected, players must navigate the car to the parking area to complete the game.

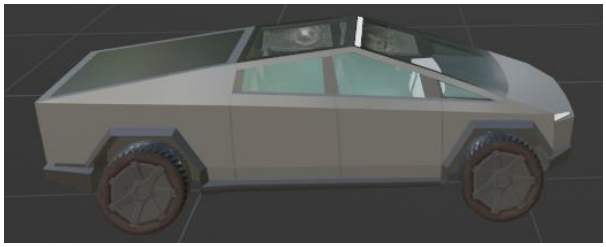

Image 4 and 5: The car the user controls

3. User Interface

The user interface (UI) in our game is designed to be intuitive, providing players with essential information and controls for a seamless gaming experience.

UI Elements:

1. Start Menu: Displays the game title and credits for acknowledgment, the game controls and the current high score.
2. HUD (Head-Up Display): Shows the player's name, timer, cash collected and the remaining cash to be collected.
3. Pause Menu: Displays the game controls, and buttons for resuming or exiting the game.
4. End Menu: Displays the player's name, the time it took to finish the game and buttons for restarting or exiting the game.

The UI features a clean and minimalist design, ensuring that vital information is easily readable during gameplay. The UI is implemented using HTML and CSS for structure and styling, and JavaScript for dynamic elements and interactions.

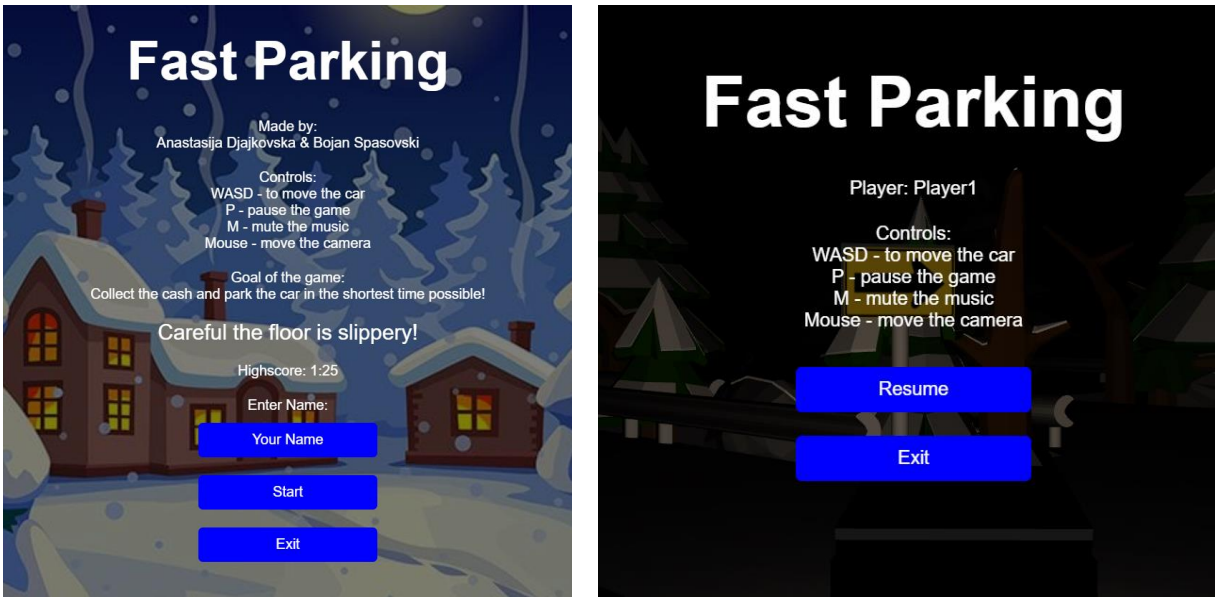

Image 6 and 7: The start menu and the pause menu

4. Music and Sound

In our game, the integration of music and sound enhances the overall player experience, adding depth and immersion to the gameplay.

Background Music: We have incorporated background music that complements the snowy setting and driving theme. The use of a continuous loop ensures a seamless auditory experience throughout the gameplay. The music can be muted.

Adaptive Music: We have implemented adaptive music elements that respond to specific in-game situations. For instance, we have included engine sounds that reflect the acceleration and deceleration, sounds for collecting money and celebrating the successful parking of the car.

The integration of music and sound is achieved through HTML5 audio elements with JavaScript.

All audio assets, including music and sound effects, are sourced from reputable free libraries.

5. Gameplay Overview

Start & Controls: Players navigate a car through a snowy landscape using W, A, S, and D keys. Additional controls include P for pause, M for mute, and mouse movement for camera adjustments.

Objective: Collect scattered cash on icy terrain swiftly. The goal is to park the car, and the faster it's done, the better the score.

In-Game Actions:

- Driving: Navigate through slippery roads.
- Cash Collection: Strategically gather scattered cash.
- Parking: Complete the game by parking the car.
- Timer & Scoring: Timer starts at 0; fastest time is the highest score.
- Collision Detection: The car interacts with the other objects of the game with collision detection.
- Dynamic Lighting: The car has headlights that light the path as the car moves.

In our game, collision detection is vital for realistic interactions between dynamic and static objects. Here's a concise overview of how we've implemented it:

- The *Physics* class manages collision detection and resolution between dynamic and static objects.
- Functions like *intervalIntersection* and *aabbIntersection* check for AABB intersections in each dimension.
- *getTransformedAABB* transforms local AABB vertices to global space.
- *resolveCollision* adjusts dynamic object positions upon collision detection.
- Static objects like parked cars, money cubes, end positions, and barriers are created and marked as static within the scene.

This system ensures realistic responses for dynamic objects, such as the player's car, enhancing the overall gameplay experience. The collision detection logic contributes to immersive gameplay as players navigate the environment, collect resources, and achieve objectives.

Dynamic lighting plays a crucial role in creating a realistic and immersive environment. Here's a concise explanation of how we've implemented dynamic lighting:

- The game features a dynamic lighting system where the world initially appears dark.

- The player's car is equipped with headlights that illuminate the surroundings as the car moves through the environment.
- The headlights dynamically light up the immediate area around the car, revealing the terrain and objects.
- This implementation mimics real-world scenarios where a moving vehicle illuminates its surroundings with its headlights.
- The dynamic lighting enhances the gameplay experience by creating a sense of realism and immersion.
- As the car moves, the dynamically lit environment allows players to discover and interact with objects strategically placed throughout the game world.

The game ends upon successful parking, displaying the final time. Players can replay for better times, mastering icy driving for optimal scores.

6. Conclusions and Possible Upgrades

The project enriched our skills in WebGL and JavaScript, providing hands-on experience in crafting engaging gaming environments.

Originally envisioned as a straightforward racetrack, the project took a more captivating turn. To enhance engagement, we introduced parking challenges and a currency system. Inspired by the season, we shifted the setting to a winter landscape with icy terrains, injecting an additional layer of excitement and strategy into the game.

Future enhancements may include improved graphics, additional levels, multiplayer integration, and customization features.

The project was a valuable learning experience, with potential for further improvements to elevate the overall gaming experience.