VR Shuffleboard Game

Zin Yamin Tun
Department of Electrical Engineering
Stanford University
zytun@stanford.edu

Abstract

The rapid advancement of virtual reality and haptic feedback technologies has opened new opportunities to experience traditional games in immersive digital formats. This project proposes the development of a virtual shuffleboard game built in Unity, capturing the essence of the physical game while enhancing interactivity through digital simulation. The initial version will use mouse-based input, laying the groundwork for future integration with a custom-designed 2-DOF pantograph device. This haptic interface will simulate physical sensations such as the inertia of the puck's mass and frictional forces, providing a more realistic and engaging user experience.

1. Introduction

Traditional games like shuffleboard offer intuitive game play and satisfying physical interactions, but are often limited by space and cost. Recent advances in virtual reality (VR) and haptic feedback technologies present new opportunities to recreate these experiences in immersive, interactive digital environments.

This project explores how shuffleboard can be brought into the virtual space using Unity. The primary goal is to simulate realistic puck's physical dynamics, including sliding friction, elastic collisions, and falling off the edge, while providing players with an intuitive interface using mouse input. Although shuffleboard games have previously been developed for mobile platforms, mouse-based control more closely aligns with the physical action of holding and launching a puck, offering a greater sense of presence and realism.

In addition, the system is intentionally designed to support future integration with a custom-built 2-degree-of-freedom (2-DOF) pantograph device. This haptic interface will simulate tactile sensations such as puck inertia and friction resistance, providing a more immersive experience that bridges digital simulation and physical interaction.

2. Game Description

The virtual shuffleboard game supports two players: red and blue. Each player has three turns to slide a puck across the board. Players may aim strategically not only to score points but also to displace the opponent's puck by knocking it off the board. After all pucks have been played, scores are tallied on the basis of their final positions in the scoring zones. The player with the highest total score is declared the winner.

3. System

3.1. Unity Setup

The 3D models of the shuffleboard table and puck were designed in Fusion 360 and imported into Unity. These assets were textured and scaled appropriately to fit the virtual environment. The Unity physics engine was used to simulate realistic interactions, including friction, collisions, and boundary detection.

3.2. Mouse Control

The game accepts two primary inputs: the position of the mouse on the screen and the state of the left mouse button. When the button is held down, the system interprets this as the player holding the puck. Releasing the button launches the puck.

The launch velocity is calculated based on the most recent mouse movements immediately preceding the release, using the formula $v = \Delta x/\Delta t$. Upon release, an impulse force is applied to the puck using Unity's built-in physics engine. To simulate realistic sliding behavior, a friction force is applied during motion. The puck's mass is set to 0.345 kg, with a dynamic friction coefficient of 0.8.

A boundary detection function continuously monitors the puck's position. If the puck moves beyond the predefined board limits, gravity is enabled to simulate it falling off the edge. Collisions between pucks are managed using Unity's sphere colliders and continuous collision detection to ensure accurate responses during high-speed inter-

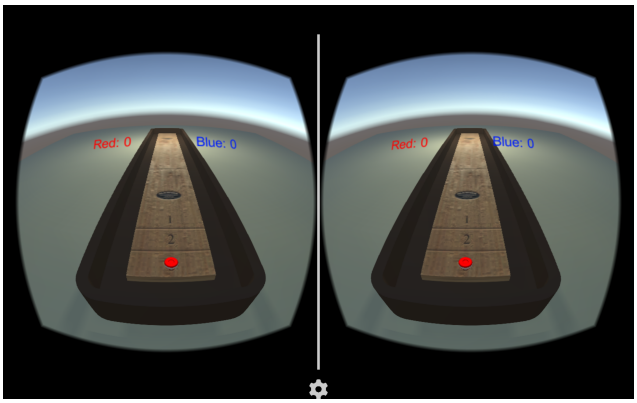

Figure 1. Stereo rendering view of the shuffleboard game interface at the start of gameplay.

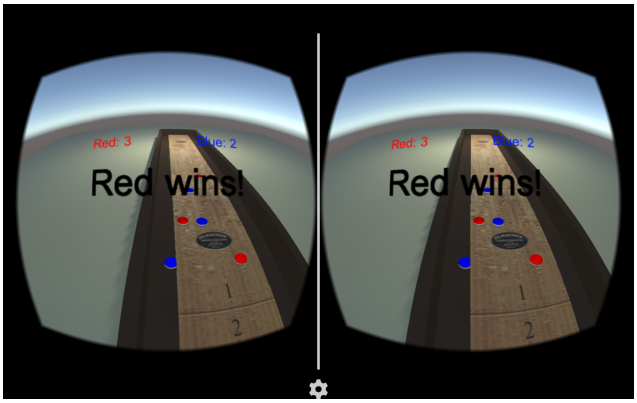

Figure 2. End-of-game screen showing final scores and the winning player.

actions. Physical parameters and launch force settings were carefully tuned to produce a natural and visually convincing experience.

A video demonstration of the game can be viewed at the following link: `https://youtu.be/c6RBVFmPa-A`.

3.3. Virtual Reality

The Unity project was integrated with a starter framework provided by the EE 267 course instructors to enable stereo rendering for virtual reality (VR). This framework supports head-tracked visualization using inertial measurement unit (IMU) data transmitted from the head-mounted display (HMD). A head-and-neck kinematic model was used to interpret IMU readings, allowing the user to rotate their head naturally and experience immersive 3D visual feedback within the virtual shuffleboard environment.

3.4. Haptic Feedback Device

A redesigned pantograph-style haptic device was developed to provide force feedback. The mechanical components were modeled using CAD software and fabricated through 3D printing and laser cutting. The system incorporates two DC motors, two magnetic rotary (MR) encoders, and two custom Arduino Uno boards provided by the ME327 course staff.

The MR encoders track the joint angles of the pantograph, enabling computation of the 2D end-effector position (x, y) and velocity. A button state is also monitored to determine whether the user is "grasping" the puck. These values—position, velocity, and button state—are transmitted from the Arduino to Unity via serial communication.

In Unity, the incoming position data is mapped to on-screen pixel coordinates to control the puck. When the button is held down, the system interprets this as the puck being held; when released, the puck is launched using the transmitted velocity, which is scaled appropriately to produce natural motion in the virtual environment.

Due to timeline differences between the two projects, the haptic version was not presented in the Virtual Reality course demo. However, a demonstration video of the haptic-enabled version is available at the following link: `https://youtu.be/apxXAllH7X4`.

4. Conclusion

This project demonstrates a promising direction for developing physically interactive virtual games. The shuffleboard simulation successfully combines intuitive control with realistic physical responses, providing an engaging user experience. During the demo session, I received positive feedback noting that the puck's motion and interactions felt natural and responsive.

In the future, this system could be extended to explore more creative variations on traditional games. For example, the board's shape could be modified to include bumps or valleys, and different surface materials—such as water, sand, or even a zero-gravity "space" environment—could be simulated to alter puck dynamics. These enhancements would open new opportunities for entertaining, customizable, and physically immersive game experiences.

5. Code Overview

Due to the large size of the full project folder, only selected components have been uploaded as follows:

- **MouseControlScripts**: Unity scripts for the virtual reality version.
- **HapticUnityScripts**: Unity scripts for the haptic feedback device version.

- **Shuffleboard_Arduino**: Arduino code for force feedback, and for sending position, velocity, and button state data.
- **VRShuffleboard.unitypackage.zip**: The VR demo project exported as a Unity package (zipped).

Below is an overview of key script files included:

- `CameraFollow.cs`: Controls the main camera to follow the currently active puck.
- `PlayerController.cs`: Implements overall game logic, including puck turn order and game termination conditions.
- `PlayerMovement.cs`: Handles physics, interaction, and puck behavior based on user input.
- `ReadUSB.cs`: Reads and parses incoming Arduino data.
- `Restart.cs`: Provides restart functionality; supports restart via button in the haptic version and via the 'R' key in the VR version.
- `ScoreManager.cs`: Calculates and manages the game scoring system.

The full code repository is available at: GitHub Repository