**# Game Design Document: Snowboard Rush**

## Technical Requirements

- Unity Version: Unity 6000.0.38f1

- Target Platform: Windows

## Team Members

- Nguyễn Quang Sơn

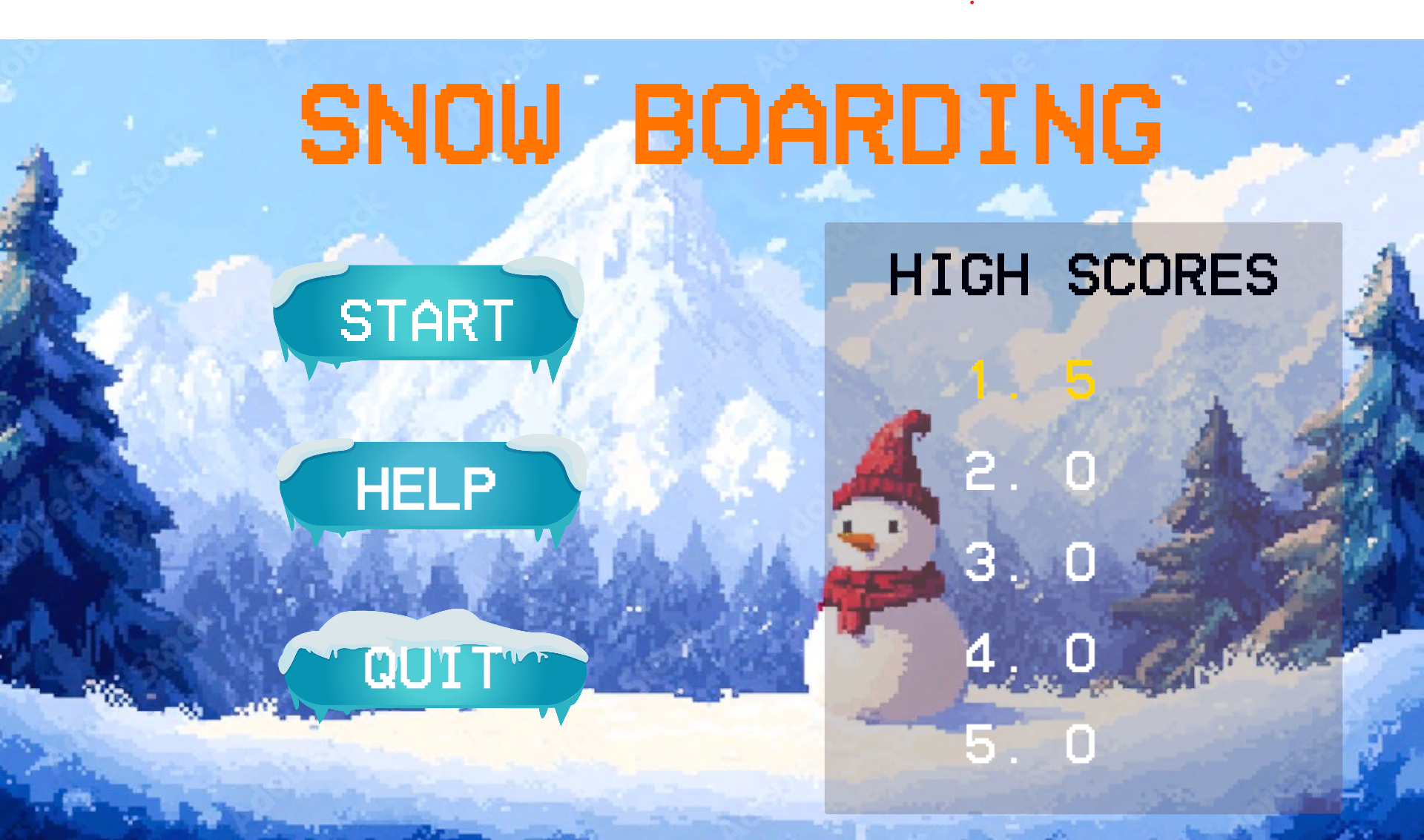
- Võ Lê Đức Anh

- Phan Đức Hậu

- Nguyễn Minh Thành

# # 1. Game Overview

Snowboard Rush is a 2D arcade-style game built in Unity where players control a snowboarder skiing down a snowy mountain. The game features a dynamic terrain system, obstacle avoidance, trick execution, and a scoring system that rewards skill and exploration.



# # 2. Game Elements

## ## 2.1. Player (PlayerController)

- Description: A 2D snowboarder that the player controls with smooth movement mechanics and physics-based interactions.

- Functionality:

- Movement:

- Left/Right arrow keys for rotation (torque-based turning)

- Spacebar for jumping

- Physics-based movement influenced by:

- Surface type (regular snow vs. ski lines)

- Slope angle

- Speed boosts after landing

- Trick System:

- Automatic flip detection during airtime

- Rotation-based trick scoring

- Points awarded for completed flips (360° rotations)

- Landing requirement to complete tricks

- Visual Effects:

- Snow trail particle effects when grounded

- Audio feedback for sliding and ski lines

- Visual tilting based on terrain angle

- Collision Detection:

- Ground collision handling

- Ski line collision detection

- Special handling for landing on surfaces

## ## 2.2. Terrain System

### ### 2.2.1. Terrain Generator

- Description: Procedurally creates the snowboarding environment with dynamic segment spawning.

- Functionality:

- Segment Management:

- Object pooling system for terrain segments

- Dynamic segment spawning ahead of player

- Segment cleanup behind player

- Segment connection system for seamless transitions

- Spawn Logic:

- Distance-based spawn trigger (50 units ahead)

- Random segment selection from prefab pool

- Initial segments for game start

### ### 2.2.2. Terrain Segments

- Description: Individual terrain pieces that combine to form the complete course.

- Functionality:

- Connection Points:

- Start and end connection points for proper alignment

- Automatic positioning based on previous segment

- Visual debugging with gizmos in editor

- Content Management:

- Obstacle placement within segments

- Collectible placement with randomization

- Difficulty progression through segment design

### ### 2.2.3. Ski Lines

- Description: Special paths that provide different physics properties for the player.

- Functionality:

- Physics Properties:

- Custom friction multiplier (affects player speed)

- One-way collision detection (top-only)

- Surface normal calculations for player orientation

- Visual Representation:

- Line renderer for visual display

- Edge collider for physics interaction

- Customizable line color and width

- Creation Methods:

- Bezier curve generation for smooth paths

- Support for straight lines

- Multiple control points for complex shapes

## ## 2.3. Collectibles & Obstacles

### ### 2.3.1. Coins

- Description: Collectible items that provide score bonuses.

- Functionality:

- Scoring:

- Award points on collection

- Audio feedback

- Score manager integration

- Placement:

- Positioned within terrain segments

- Random activation based on probability

### ### 2.3.2. Obstacles (Spikes)

- Description: Hazardous elements that can end the player's run.

- Functionality:

- Collision:

- Triggers game over on player contact

- Works with CrushDetector system

- Placement:

- Strategic positioning in terrain segments

- Difficulty-based distribution

## 

## ## 2.4. Visual Elements

### ### 2.4.1. Farthest Point Marker

- Description: Visual indicator showing the player's record distance.

- Functionality:

- Visual Features:

- Bobbing animation

- Optional rotation

- Text display showing record distance

- Placement:

- Positioned at the player's farthest achieved distance

- Updates in real-time when new records are set

### ### 2.4.2. Scrolling Background

- Description: Creates a sense of movement with a parallax effect.

- Functionality:

- Movement:

- Continuous horizontal scrolling

- Speed-based adjustment

- Material texture offset manipulation

# # 3. Game Systems

## ## 3.1. Score System (ScoreManager)

- Description: Handles point tracking, high score management, and UI updates.

- Functionality:

- Score Tracking:

- Real-time score updates

- UI display integration

- Score persistence between scenes

- High Score Management:

- Stores top 5 scores

- Persistent storage using PlayerPrefs

- High score display in main menu and completion screen

- Scoring Rules:

- Point rewards for collecting coins

- Point rewards for performing tricks

- Score saving on game completion



## ## 3.2. Distance Tracking System (DistanceTracker)

- Description: Tracks and records the player's traveled distance.

- Functionality:

- Distance Calculation:

- Real-time distance measurement in meters

- Conversion from Unity units to meters

- UI display integration

- Record Management:

- Tracks farthest achieved distance

- Persistent storage using PlayerPrefs

- Visual notification when records are broken

- Farthest Point Marker:

- Visual indicator at record distance

- Dynamically updated when records are broken

- Text display showing record distance

## ## 3.3. Game State Management

### ### 3.3.1. Main Menu

- Description: Entry point for the game with various options.

- Functionality:

- Navigation Options:

- Play Game button

- How to Play button with instruction panel

- Quit Game button

- High Score Display:

- Shows top 5 high scores

- Persistent data from ScoreManager

### ### 3.3.2. Pause Menu

- Description: In-game pause functionality.

- Functionality:

- Game State Control:

- Pause/resume game functionality

- Time scale management

- Canvas group-based UI visibility

- Menu Options:

- Resume Game

- Restart Game

- Quit to Main Menu

### ### 3.3.3. Game Completed Manager

- Description: Handles the end-game screen after a run.

- Functionality:

- Score Processing:

- Displays final score

- Compares with high score

- Shows notification for new high scores

- Distance Records:

- Displays distance traveled

- Compares with farthest distance record

- Shows notification for new distance records

- Navigation Options:

- Play Again button

- Back to Main Menu button

# # 4. Technical Systems

## ## 4.1. Collision Detection

### ### 4.1.1. CrushDetector

- Description: Handles game-ending collisions and transitions.

- Functionality:

- Collision Detection:

- Monitors player collisions with ground and death zones

- Special handling for spike obstacles

- Prevents multiple trigger activations

- Game Over Handling:

- Plays crush sound effect

- Disables player physics

- Saves distance and score records

- Transitions to game completed scene

### ## 4.2. Save System

- Description: Persistent data storage across game sessions.

- Functionality:

- Stored Data:

- High scores (top 5)

- Farthest distance record

- Last score and distance for game completed screen

- Implementation:

- PlayerPrefs for data persistence

- String serialization for score lists

- Float and int storage for records

### ## 4.3. Audio System

- Description: Handles game sound effects and audio feedback.

- Functionality:

- Sound Categories:

- Player movement (sliding, ski line)

- Record breaking notification

- Coin collection

- Crush/impact effects

- Implementation:

- AudioSource components

- Dynamic pitch adjustment based on speed

- Volume adjustments based on terrain

# # 5. Control Summary

- Movement: Left/Right arrow keys

- Jump: Spacebar

- Pause: ESC key



# # 6. Level Design Philosophy

- Progressive difficulty through terrain segments

- Strategic placement of coins and obstacles

- Balance between challenge and rewards

- Record systems to encourage replayability

- Visual feedback for player actions and achievements

# # 7. Appendix: Class Relationships

## ## Core Gameplay

- PlayerController: Handles player movement, physics, and tricks

- TerrainGenerator: Manages terrain creation and pooling

- TerrainSegment: Individual terrain pieces with start/end points

- SkiLine: Special terrain paths with unique physics properties

## ## UI and Game State

- ScoreManager: Handles scoring and high scores

- DistanceTracker: Records and displays travel distance

- MainMenu: Manages main menu UI and navigation

- PauseMenu: Handles in-game pausing

- GameCompleteManager: Processes end-game state

## ## Collectibles and Obstacles

- Coin: Collectible items for scoring

- CrushDetector: Handles fatal collisions

## ## Visual Elements

- FarthestPointMarker: Visual indicator for record distance

- RecordNotification: UI notification for new records

- ScrollingBackground: Creates parallax background effect

- SnowTrail: Manages snow particle effects