ICEBERG BRAWL

Game Design Document

A GDD ON THE PORTFOLIO PIECE: ICEBERG BRAWL"

Author:

Jeppe WILLATZEN

June 30, 2024

Contents

1 Game Overview

1.1 Game Concept

- **Title:** Iceberg Brawl

- **Genre:** Multiplayer Party Game

- **Platform:** PC

- **Target Audience:** Casual gamers, ages 10 and up

1.2 Game Summary

Iceberg Brawl is a fast-paced multiplayer party game where 2-4 players compete to stay on a slippery iceberg while trying to shove their opponents into the freezing water. The last player remaining on the iceberg wins the round.

1.3 Unique Selling Points

- **Dynamic slippery mechanics:** The core mechanic of slipping on ice adds a unique and challenging twist to the gameplay, making every movement a strategic decision. The physics-based interactions provide a fresh and engaging experience not commonly found in typical party games.

- **Fast-paced multiplayer action:** Iceberg Brawl offers intense and quick rounds, perfect for both casual and competitive play. The ability to play with friends locally or online makes it accessible and engaging for a broad audience.

- **Simple yet deep gameplay:** The basic controls and straightforward objective make the game easy to pick up and play, but mastering the slippery mechanics and outmaneuvering opponents adds depth and replayability.

- **Vibrant and playful visuals:** The cartoonish art style and colorful characters appeal to a wide range of players, from children to adults. The lively animations and humorous shoving interactions enhance the fun and light-hearted atmosphere.

1.4 Player Experience and Game POV

1.4.1 Player Experience

- **Who is the player?** The player is one of the comical, vibrant avatars, each with their own personality and style, competing to be the last one standing on the slippery iceberg.

- **What is the setting?** The game is set in a whimsical, frozen world with various themed icebergs, each presenting unique challenges and visual styles. Players find themselves in a fantastical arctic arena filled with dynamic and interactive elements.

1.4.2 Emotions and Engagement

- **What emotions should the player feel?** The game aims to evoke emotions of excitement, joy, and laughter. The chaotic nature of slipping and shoving, combined with the pressure of being the last one standing, creates a thrilling and entertaining experience. Players should feel a mix of exhilaration from close calls and satisfaction from successfully shoving opponents into the water.

- **What keeps the player engaged?**

  - **Competitive edge:** The competitive nature of the game, with its fast-paced rounds and the drive to be the last one standing, keeps players engaged.

  - **Social interaction:** Playing with friends, whether locally or online, adds a layer of social fun and camaraderie.

  - **Variety and customization:** Different characters, icebergs, and environmental hazards keep the gameplay varied and interesting. Players can look forward to unlocking new customizations and arenas.

  - **Replayability:** The simple yet deep mechanics encourage players to keep improving their skills and strategies, ensuring that no two matches are ever the same.

2 Gameplay

2.1 Core Mechanics

- **Player Movement:** Players can move in all directions on the iceberg. The surface is slippery, causing players to slide.

- **Shoving:** Players can shove each other by pressing a button, applying a force to push opponents away.

- **Falling Off:** If a player slides off the edge of the iceberg, they fall into the water and are out of the round.

- **Winning Condition:** The last player remaining on the iceberg wins the round.

2.2 Multiplayer Features

- **Lobby System:** Players can host or join multiplayer games.

- **Network Synchronization:** Player movements and interactions are synchronized across the network to ensure smooth gameplay.

- **Matchmaking:** Basic matchmaking to connect players to games.

2.3 Game Modes

- **Free-for-All:** All players compete against each other. The last player standing wins.

- **Team Mode (optional):** Players are divided into teams, and the last team with members on the iceberg wins.

3 Level Design

3.1 Iceberg Level

- **Shape:** Circular iceberg to encourage constant movement and interaction.

- **Surface:** Low friction to create a slippery effect.

- **Size:** Small enough to keep players in close proximity but large enough for strategic movement.

3.2 Environmental Hazards (optional)

- **Cracks:** Parts of the iceberg that crack and break away over time.

- **Ice Floes:** Moving ice platforms that players can use to avoid being shoved off.

3.3 Powerups (optional)

- **Speed Boost:** Temporarily increases the player's movement speed.

- **Super Shove:** Enhances the player's shoving power, making their shoves more effective.

- **Sticky Boots:** Temporarily negates the slipperiness of the ice, providing better traction.

4 Visual and Audio Design

4.1 Visual Design

The visual design of *Iceberg Brawl* is crucial in creating an engaging and enjoyable experience for players. The design should be vibrant, fun, and visually appealing to cater to a wide audience, including children and casual gamers. Here's a breakdown of the visual design elements:

4.1.1 Art Style

- **Cartoonish and Colorful:** The game will feature a cartoonish art style with bright and vibrant colors to make the game visually appealing and accessible to all ages.

- **Character Design:** Each character will have a unique and playful design with exaggerated features to enhance the cartoonish feel. Characters should be distinguishable by their colors, shapes, and costumes.

- **Environment Design:** The icebergs and surrounding environment will have a whimsical, frosty look with playful elements like icicles, snowflakes, and animated sea creatures in the water.

4.1.2 User Interface (UI)

- **Main Menu:** A simple, intuitive menu with large buttons and playful fonts. Background animations could include floating icebergs or falling snow.

- **In-Game HUD:** Minimalistic and unobtrusive, showing only essential information like player scores, power-up timers, and match time. Use bright colors and icons to keep the UI visually consistent with the game's art style.

- **Power-Up Indicators:** Clear and distinct icons for each power-up, with brief animations when a power-up is collected or used.

4.1.3 Animations

- **Character Animations:** Smooth and exaggerated animations for movement, shoving, falling, and power-up usage. These animations should be

playful and reflect the game's cartoonish style.

- **Environmental Animations:** Icebergs should have subtle movements and interactions with the water. Additional environmental effects like splashes when players fall into the water, or small ice shards when shoves occur, can add to the immersion.

- **Special Effects:** Use particle effects for power-ups, shoving impacts, and other interactions. These should be visually distinct and match the playful aesthetic.

4.1.4   Lighting and Shading

- **Lighting:** Bright and dynamic lighting to enhance the colorful environment. Use light sources like the sun reflecting off the ice and water.

- **Shading:** Soft and smooth shading to complement the cartoonish art style. Avoid realistic shading techniques to maintain the playful atmosphere.

4.2   Audio Design

The audio design should complement the visual style and enhance the overall gameplay experience. Here's a detailed plan for the audio elements:

4.2.1   Background Music

- **Theme:** Upbeat and cheerful music to match the light-hearted and fun nature of the game. The music should be dynamic, with variations that reflect different in-game situations like the start of a round, intense moments, and victory celebrations.

- **Loops:** Seamless looping tracks that keep the energy up without becoming repetitive or annoying.

4.2.2   Sound Effects

- **Movement and Interaction:** Sounds for walking on ice, slipping, shoving, and falling into the water. These should be exaggerated and playful to match the cartoonish visuals.

- **Power-Ups:** Distinct sound effects for each power-up, both when collected and when used. These should be short, satisfying, and easily recognizable.

- **Environmental Sounds:** Ambient sounds like wind blowing, water splashing, and subtle iceberg creaking. These should be present but not overpowering, adding to the immersion without distracting from the gameplay.

4.2.3 Voice Overs and Character Sounds

- **Character Reactions:** Playful and varied voice lines or sound effects for characters when they shove, get shoved, collect power-ups, or fall off the iceberg.

4.3 Character Examples

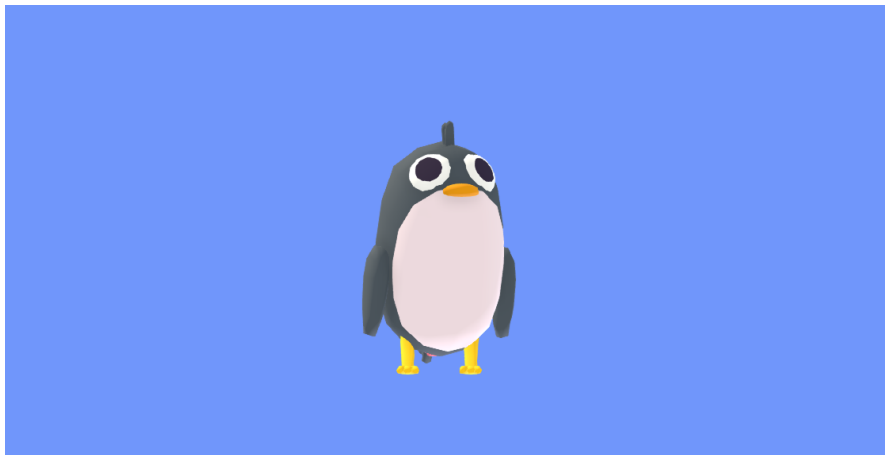
Figure 1: Polar Bear

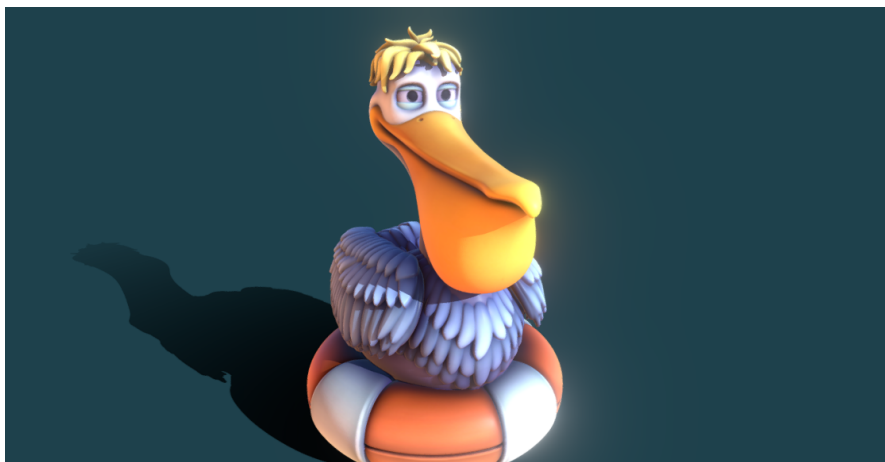Figure 2: Winter-themed Explorer

Figure 3: Penguin

Figure 4: Pelican

5   Technical Design

5.1   Development Tools

- **Game Engine:** Unreal Engine

- **Version Control:** Git

- **3D Modeling:** Blender

- **Audio Design:** FMOD

5.2   Game Environment

- **Dimensions:** The game is a 3D platformer, providing a fully three-dimensional environment for the players to interact with.

- **Camera:** Third-person perspective to allow players to see their character and surroundings clearly.

5.3   Controls

- **Movement:** WASD keys for movement.

- **Shoving:** Spacebar to shove other players.

- **Jumping (optional):** Key to be determined if jumping is included.

5.4   Asset Creation and Management

- **3D Models:** Created using Blender. Models will include player avatars, the iceberg, and any additional environmental elements or hazards.

- **Textures and Materials:** Created in Blender and imported into Unreal Engine. Textures will be designed to be colorful and cartoonish to match the game's aesthetic.

- **Animations:** Character animations for movement, shoving, and falling off the iceberg will be created in Blender and integrated into Unreal Engine.

- **Audio Assets:** Sound effects and background music created and managed using FMOD. This will include sounds for player actions, environmental effects, and ambient background music.

5.5 Networking

- **Multiplayer Framework:** Utilizing Unreal Engine's built-in networking capabilities.

- **Replication:** Ensuring all critical game data is replicated across the network to maintain consistent game state for all players.

- **Latency Management:** Implementing techniques to handle network latency and ensure smooth gameplay, such as client-side prediction and server reconciliation.

5.6 Optimization

- **Performance Testing:** Regular performance testing to ensure the game runs smoothly on the target platform.

- **Asset Optimization:** Reducing polygon counts for 3D models, optimizing textures, and ensuring efficient use of resources.

- **Network Optimization:** Minimizing the amount of data sent over the network to reduce latency and improve responsiveness.

5.7 Tools and Libraries

- **Blender:** For creating and animating 3D models.

- **FMOD:** For creating and integrating audio assets.

- **Unreal Engine:** For game development, including physics, rendering, and networking.

- **Git:** For version control, allowing for efficient collaboration and tracking of changes.

Development Plan

Week 1: Setup and Basic Mechanics

- **Day 1-2: Set up Unreal Engine and create a new project.**

  - Install Unreal Engine and create a new project.
  - Set up source control (e.g., Git) to manage your project files.

- **Day 3-4: Implement player movement and basic mechanics.**

  - Create a basic character controller with movement.
  - Implement slipping mechanics on the iceberg.

- **Day 5-7: Design the initial iceberg level.**

  - Create a circular, slippery iceberg level with appropriate friction settings.
  - Implement basic collision detection.

Week 2: Networking and Multiplayer Setup

- **Day 8-10: Set up basic multiplayer functionality.**

  - Integrate Unreal's multiplayer framework.
  - Create a simple multiplayer lobby and connect multiple players to the same game session.

- **Day 11-12: Sync player movement and interactions across the network.**

  - Implement network replication for player characters.
  - Ensure smooth synchronization of player actions across all clients.

- **Day 13-14: Implement shoving mechanics.**

  - Create a system for players to shove each other.
  - Ensure shoving actions are synced and result in realistic physics reactions.

Week 3: Game Rules, Polish, and Testing

- **Day 15-17: Add scoring and win/loss conditions.**

  - Track which players are still on the iceberg.

  - Implement a scoring system to determine the last player standing.

- **Day 18-19: Polish visuals and add sound effects.**

  - Improve the visual design of the iceberg and player characters.

  - Add sound effects for movement, shoving, and falling into the water.

- **Day 20: Playtest the game and fix any bugs.**

  - Conduct thorough playtesting with multiple players.

  - Identify and fix any bugs or balance issues.

- **Day 21: Prepare the game for release and create a portfolio entry.**

  - Package the game for distribution.

  - Create a portfolio entry with screenshots, gameplay videos, and a description of the game.

Detailed Steps for Key Features

1. **Player Movement and Slipping Mechanics:**

   - Implement a slippery surface by adjusting the friction values of the iceberg.

   - Ensure player characters have momentum and slide realistically on the ice.

2. **Multiplayer Setup:**

   - Use Unreal Engine's multiplayer capabilities to create a lobby system.

   - Allow players to host and join games easily.

3. **Shoving Mechanics:**

   - Add a shoving action triggered by a button press.

   - Use physics forces to push other players, with varying strength based on conditions (e.g., running start).

4. **Game Rules and Scoring:**

   - Implement a system to detect when a player falls off the iceberg.

   - Track and display which player is the last one standing.

5. **Polish and Optimization:**

   - Add visual feedback for slipping and shoving (e.g., animations, particle effects).

   - Optimize network performance to ensure smooth multiplayer gameplay.

Tips for Success

- **Focus on Core Mechanics First:** Ensure player movement, slipping, and shoving feel good before adding additional features.

- **Regular Testing:** Test with multiple players frequently to catch and fix issues early.

- **Iterate on Feedback:** Use feedback from playtesting to improve gameplay and balance.

- **Keep It Simple:** Aim for a fun and playable game rather than overcomplicating mechanics or features.

Appendix

Additional Characters

Details about extra characters that might be included in future updates.

Concept Art & References

Figure 5: Polar Panic Level from Crash Bash

Figure 6: Polar Panic Level from Crash Bash

Figure 7: Polar Panic Level from Crash Bash

Test Results

Record the results of playtesting sessions, including feedback and bug reports.