**Project Name: Crowd Surge**

**What is “Crowd Surge”**

Our project "Crowd Surge" is a 2D immersive hybrid-casual game that offers a unique blend of strategic depth and casual accessibility. At its core, players lead and grow a dynamic crowd through a series of imaginative levels, each presenting its own set of challenges and obstacles.

Players need to using left and right keys to navigate through multiplier gates to expand their crowd and overcome the red enemy mobs. Beyond the core gameplay, "Crowd Surge" introduces a pseudo PvP element, creating an illusion of competing against other players, which amplifies the thrill and competitive spirit.

In future, we may also build more metagame features, like involving base building and resource management, adding layers of goals and satisfaction. With its vibrant graphics, intuitive controls, and a balance of quick-to-learn mechanics with long-term engagement strategies, "Crowd Surge" promises a fun and rewarding experience for both casual gamers and those seeking a more involved gaming session.



Figure 1 concept example

**Basic Functions:**

**Goal of the game**

Players' main goal is to guide a growing crowd through a series of increasingly challenging obstacles, enemies and collective objects. The ultimate aim is to successfully navigate crowds while growing the crowd sufficiently to defeat enemy mobs and reach the end. Reaching this goal requires skillful management of the crowd, utilizing multiplier gates to grow the crowd, and avoiding or overcoming obstacles and enemies.

**Rules of the game**

Crowd Management

The game's core rule involves using multiplier gates/plus gates strategically. These gates increase the number of crowd members, but choosing the right gate is crucial for survival and success. Players must also avoid obstacles that can decrease the crowd size or lead to failure.

Obstacle Navigation

Game contains a variety of obstacles and enemy types. Some obstacles are static, while others are dynamic, requiring timing and strategic movement. Enemies have specific behaviors and patterns that players must learn and counter.

**Challenge of the game**

Increasing Difficulty:

The game is designed with an escalating difficulty curve. Early game period introduces basic mechanics, while later periods feature more complex layouts and challenging enemy configurations. The difficulty is not just in the level design but also in managing a larger crowd and making quick strategic decisions.

Strategic Decision-Making:

Players are constantly faced with choices: which path to take, which gate to go through, when to confront enemies, and when to avoid them. These decisions impact the size and strength of the player's crowd and their ability to complete the level.

**Interaction of the game**

Player and Environment:

Interaction with the game environment is constant. Players must navigate through levels, interact with multiplier gates, avoid or dismantle obstacles, and engage with enemy mobs. The environment is dynamic, with elements that can change based on the player's actions or game progression.

Player and Game Mechanics:

Players interact with various game mechanics, including crowd control, resource management, and strategic planning in both the core gameplay and the metagame.

The game offers a variety of interactive elements, from simple movements for crowd control to more complex strategies about how to choose gates and how to face enemies.

**Special Features:**

**Crowd Mechanics:**

Players guide a crowd through an infinite long game until the crowed vanished, with the crowd growing or shrinking based on interactions with **collectible characters , enemy and obstacles.** The game will feature a fluid and responsive control system, allowing players to navigate complex environments.



Figure 2 collectible characters and obstacles example

**Level Design and Progression:**

Game will offer unique challenges, including a variety of obstacles (like traps, moving platforms) and enemies. With the game progressing, the background will vary in theme, difficulty, and layout, ensuring a fresh experience in each stage.

**Strategic Gameplay:**

integrates strategic elements into its core gameplay. Players must make **tactical decisions** about when and where to grow their crowd, how to maneuver through obstacles, and the best way to defeat enemy forces.

**Enhanced Pseudo PvP Experience:**

While being a single-player game, our game will simulate a PvP (player vs. player) environment by developing **enemy behavior strategies**. This feature enhances the competitive feel of the game, making each level not just a challenge against the game but also a competition against seemingly real opponents.

**Performance Targets:**

"Crowd Surge" is being developed using the robust and versatile C++ programming language, complemented by video and audio libraries provided by the course, ensuring a consistent experience across different operating systems.

Graphics and Audio: The game will feature stylized 2D graphics with top-down view and an engaging soundtrack, enhancing the overall player experience.

**Market Identification:**

**Target User Group:** "Crowd Surge" targets a broad audience, including teenagers and young adults, particularly those active on streaming platforms like TikTok and Twitch. The game will also be a hit with casual strategy and puzzle enthusiasts, offering short yet captivating play sessions.

**Size of the User Group:** In research from Silvija Lazanin “Mob Control Analysis: The Anatomy of a Hybrid-Casual Hit” The same categories hyper-casual game Mob Controlamassed over 50 million downloads in a single year and ranked**#1 on the top-grossing chart** for 2023 (AppMagic).

**Competing Products:**

Direct Competitors: Games like "Mob Control, Pigeon Pop, Helix Jump" and other popular crowd-based strategy games.

Indirect Competitors: Other casual and strategy games on mobile and desktop platforms that capture similar user demographics.

**System Description:**

1. Level Design and Development (People 1 and 2)

**Level Architecture Development:** Constructing the game's levels by programming the placement of obstacles, enemies, collectibles and environment. Balancing technical constraints with gameplay needs to maintain challenge and playability.

1. Crowd Dynamics Development (People 3 and 4)

**Crowd Behavior Algorithms:** Creating and refining algorithms for crowd behavior, including movement patterns, formation logic, and interactions with game elements.

**Character Interaction Coding:** Scripting the interactions among different character types within the crowd to ensure cohesive group dynamics.

**Environmental Interaction Logic:** Programming the crowd's logical responses to environmental factors like obstacles and power-ups.

3. Graphics and Rendering (People 5 and 6)

**2D avatar and Animation Coding:** Creating 2D avatars and animations for characters, obstacles, and environments within the game's engine.

**Rendering Optimization:** Implementing and optimizing rendering to ensure a consistent frame rate and quality across different hardware.

**Visual Effects Implementation:** Programming visual effects, such as particle systems and lighting, to enhance the game's visual appeal.

6. Enemy behavior Development (People 7)

**Enemy Behavior Programming:** Developing the behaviors of enemy characters. This includes developing algorithms that dictate enemy behaviors, movement patterns, and reactions to player actions. The goal is to create challenging and varied enemy that enhances the game's difficulty and engagement.

**Assembly Rules of Functional Components**

Begin by establishing a foundational architecture that allows for the integration of all game components. This should include a core game loop, resource management systems, and a modular design that facilitates independent development and testing of each component.

**Level Architecture Integration:**

The level architecture will be the first layer upon this foundation. The placement of obstacles, enemies, and collectibles will be integrated into the game's engine, ensuring that level data is well-structured and can dynamically adjust based on gameplay.

**Crowd Dynamics Integration:**

Crowd dynamics must interact seamlessly with the level architecture. The crowd behavior algorithms and character interactions need to respond to the level design, meaning the crowd's actions should be influenced by the environment and the obstacles placed by the level designers.

**Enemy Behavior Coordination:**

Enemy behavior programming, managed by People 7, needs to be closely coordinated with both the level design and crowd dynamics. The enemies' AI should be aware of the level layout and the crowd's state to provide a responsive and challenging experience for the player.

**Graphics and Rendering Synchronization:**

The visual components created by People 5 and 6, including 2D avatars, animations, and visual effects, should be synchronized with the gameplay mechanics. This ensures that visual feedback is accurate and timely, corresponding with the player's actions and the game's events.

**Holistic Testing and Iteration:**

Once all components are integrated, comprehensive testing is essential. This should include not just bug testing, but also gameplay balance and performance testing across different hardware configurations to ensure the game delivers the intended experience to all players.