

Retro Game Jam GDD Template

Game Title

GlitchCycle

Team Name

Group 30 (G 30)

Team Members

- Jeff Paller (21677646)
- Mridul Nohria (74243379)
- Raunak Khanna (89443253)
- Muhammad Ahsan Kalam (86275237)

Core Concept

Our game reimagines the classic Tron: Light Cycle (1982) arcade experience, where competing riders leave brightly colored trails behind them and must avoid colliding with any trail. Our twist introduces a lateral jump mechanic that allows players to vault over their own or an opponent's trail, creating opportunities for strategic escapes and maneuvers. Players must carefully time these jumps to transform what would be certain collisions into skillful evasions, resulting in fast-paced, tactical gameplay that builds upon the neon-dueling action of the original Tron.

Core Gameplay

In each round, players control light cycles that continuously leave glowing trails behind them, requiring careful navigation to avoid collisions with any line—whether their own or an opponent's. The core loop centers on strategic movement and the lateral jump ability that allows players to vault over trails when cornered. Because colliding with any trail results in immediate knockout, every decision carries high stakes. Visual and audio feedback enhance the experience through distinctive jump and crash animations, as well as jump and crash sound effects.

Player Controls

Player 1 Controls

- A = Turn Left
- D = Turn Right
- Q = Jump Left
- E = Jump Right

Player 2 Controls

- J = Turn Left
- L = Turn Right
- U = Jump Left
- O = Jump Right

Note: In Tutorial mode, Player 2 is controlled by an AI opponent.

Game Modes

The game features two distinct modes:

Tutorial Mode

A single-player experience where players compete against an AI opponent designed to teach the fundamentals of gameplay. The AI demonstrates basic strategies while providing a manageable challenge for beginner players to learn the controls and mechanics.

2-Player Mode

A competitive mode where two players face off on the same device using the control scheme detailed above. This mode allows friends to challenge each other in direct competition across multiple arenas.

Level & Progression

Game Structure

Our game consists of five handcrafted levels, each set in a uniquely designed arena featuring different obstacles and wall patterns. These levels are available in 2-Player mode, ensuring variety in gameplay environments.

Scoring System

GlitchCycle employs a persistent scoring system that tracks the total number of rounds won by each player throughout the entire play session. Each time a player wins a round (by being the last cycle standing), their score increments. This running tally continues until players choose to exit the game, providing an ongoing competitive experience without predetermined match endpoints. The scoring display remains visible after every round, allowing players to track their performance history across multiple rounds.

Timeline & Milestones

Week 1: Core Mechanics & Tutorial Mode Implementation

1. Set up a basic Unity project:
 - Initialize the project, configure version control, and establish project settings.
 - Assigned to: Jeff Paller

2. Implement player input and core mechanics:
 - Program the light cycle's movement using turn-based controls, ensuring continuous forward motion and trail generation.
 - Assigned to: Jeff Paller

3. Implement lateral jump ability:
 - Code the lateral jump mechanism with proper cooldown logic for both players.
 - Assigned to: Mridul Nohria

4. Implement scoring system:
 - Develop a persistent scoring system that tracks total rounds won throughout the play session.
 - Assigned to: Raunak Khanna

5. Develop a basic UI:
 - Create initial UI elements to display cumulative scores, jump cooldown indicators, and control information.
 - Assigned to: Muhammad Ahsan Kalam

6. Implement tutorial AI opponent:

- Create a simplified AI that demonstrates basic movement patterns and uses jumps occasionally to teach players the game mechanics.
- Assigned to: Jeff Paller

Week 2: 2-Player Mode, Levels & Polish

1. Implement 2-Player mode:

- Set up controls for both players and ensure proper gameplay mechanics for competitive matches.
- Assigned to: Jeff Paller

2. Design and implement five levels:

- Handcraft five unique arena maps with varying obstacles and wall patterns for both game modes.
- Assigned to: Everyone

3. Add visual and audio polish:

- Integrate particle effects and animations for jumps and trail interactions, along with appropriate sound effects and background music to evoke a retro neon aesthetic.
- Assigned to: Mridul Nohria

4. Refine the UI and menus:

- Enhance the UI with clear on-screen indicators for cumulative scores and jump cooldowns, and implement menus (main menu, mode selection, level selection, pause screen).
- Assigned to: Raunak Khanna

5. Balance and fine-tune gameplay:

- Playtest and adjust the tutorial AI's difficulty (speed, jump frequency) and ensure both game modes provide an engaging experience.
- Assigned to: Jeff Paller

6. Final testing and packaging:

- Conduct final tests, fix remaining bugs, record a gameplay demo video, update documentation (README), and package the final build for submission.
- Assigned to: Everyone

Assets

Models & Art:

Provide links or sources for sprites, models, and animations.

- <https://opengameart.org/content/grid-hud> Good map/grid like background for all levels.
 - Found on OpenGameArt
- For the bike/Player, we can use: <https://opengameart.org/content/spaceship-building-kit>
 - Found on OpenGameArt
- For the walls/obstacles on the map we can simply use Unity's shape tool to design a thin line following behind the player, and also to make the simple obstacles that will be for differing levels (rectangles, circles, etc).
- Jump animation for when jumping laterally: <https://opengameart.org/content/warp-effect-2>
 - Found on OpenGameArt.
- Simple explosion animation for when crashing <https://opengameart.org/content/bubble-explosion>
 - Found on OpenGameArt

Sound & Music:

Provide links or sources for music and sound effects.

- <https://freesound.org/people/CJspellsfish/sounds/727650/> RETRO JUMP SOUND (for when using lateral jump movement).
 - found on FreeSound.
- <https://freesound.org/people/squareal/sounds/237375/> CRASH SOUND EFFECT (might need to be clipped but still good for when player hits wall).
 - found on FreeSound
 - Also, clip the beginning of this for a good turning sound effect.
- <https://freesound.org/people/Timbre/sounds/561191/> MAIN MENU MUSIC, good loop-able music for a TRON vibe while selecting the level you want to play/upon start-up.
 - Found on FreeSound.
- <https://freesound.org/people/Bertrof/sounds/131658/> SELECTION SOUND EFFECT, sound used for when a player selects say a level or something else in the main menu, as feedback confirmation that they clicked something/something was selected.
 - Found on FreeSound
- <https://freesound.org/people/furbyguy/sounds/331876/> Good in game music (good music for a level to use)
 - Found on FreeSound
- <https://freesound.org/people/Timbre/sounds/496186/> Good in game music (good music for a level to use)
 - Found on FreeSound
- <https://freesound.org/people/3ag1e/sounds/745852/> Good in game music (good music for a level to use)
 - Found on FreeSound
- <https://freesound.org/people/zagi2/sounds/223475/> Good in game music (good music for a level to use)
 - Found on FreeSound
- <https://freesound.org/people/danlucaz/sounds/497109/> Good in game music (good music for a level to use)
 - Found on FreeSound

UI:

For our UI, we aim to create a retro-futuristic interface that complements the Tron-inspired aesthetic. Here's what we plan to include:

- **Fonts:**

- Retro Font to use: <https://fonts.google.com/specimen/Press+Start+2P>
 - Publicly given by Google Fonts
- <https://fonts.google.com/specimen/VT323?query=Vt> This font compliments the above if we decide to use more than one.
 - Publicly given by Google Fonts

- **Icons and UI Elements:**

- **Icons:**

- If we need any icons in particular, this collection seems to have all the basic things: <https://opengameart.org/content/grayscale-icons>
 - Found on OpenGameArt

- **Main Menu:**

- A custom-designed main menu featuring minimalist panels and neon-styled buttons.
- Options will include:
 - **Play:** Launches the game and then prompts a level selection screen.
 - **Settings:** Opens a settings menu for adjusting game options like sound, controls, etc.
 - **Quit:** Exits the game.
- If we can't think of anything for the main menu, we can always use this: <https://opengameart.org/content/game-icons-and-buttons>
 - Found on OpenGameArt

- **In-Game HUD:**

- Custom-designed UI elements to display round scores and ability cooldowns with a retro neon aesthetic (likely simple text at top of screen in bright text).
-