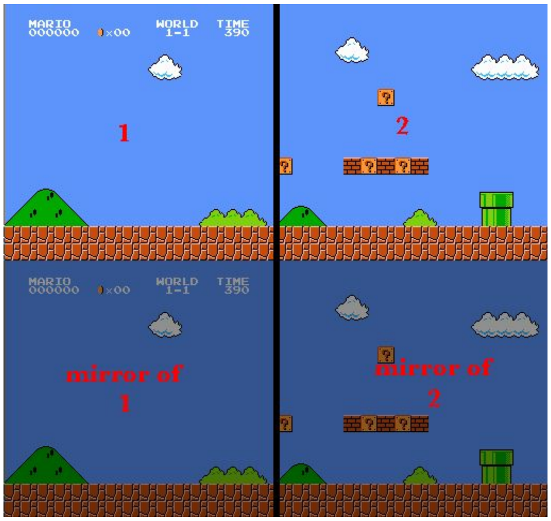

Mountain Climber VS

Antonio-Angel Medel | Juan Villasenor | Josh Logue | Andrew Manz

Overview:

Who can get to the top first? Beat your friend climbing the perilous mountain. Dodge hazards and leap from the safety of platforms to dangerous inclines and back again all while inflicting those same challenges upon your opponent.

A competitive split-screen multiplayer auto scrolling platformer, drawing inspiration from classic platformers like Super Mario as well as head-to-head competition featuring disruption similar to Puyo Puyo Tetris.

Technical Requirements:

- Programming will be done in C#
- Will utilize Unity
- Runs on Linux
- Inputs:
  - Keyboard: W,A,S,D, Arrow Keys, Jump button, Use button, Start, Select for both players on one keyboard
  - Gamepad: D-Pad, A, B, Start, Select: Emulating the NES Controller

Product Features:

- Art
  - 2-D
  - Sprite based

- Mario Style Platformer
- Two Player Competitive splitscreen
- Procedurally Generated platforms and walls
- Movement options
    - Four-Directional movement
    - Jump
    - Latch to wall with ice pick, and jump from there
- Two Modes: Race and Survival
    - Race: Whoever gets to the top of the mountain first wins
    - Survival: whoever stays on screen the longest wins
- Tracking of score
- Controller Support

Overall Goals:

- Making a vertical scrolling level
- Implement split-screen multiplayer
- Learn and gain experience using Unity
- Mountain climber tools use [ice pick, ziplines,... etc.]
- Powers to help you or hurt your opponent
- Implementation of sprites

Stretch Goals:

- Implementation of networks for online multiplayer matchmaking??
- More unique tools [jet pack...]
- vs Computer [AI...]
- Implementation of Enemies and Tool Boxes

Division of Labor:

- **Juan**
    - Score keeping: How high someone has gone in Survival mode, How quickly someone can finish the level in Race mode
    - Multiplayer interaction
- **Tony**
    - Player Functionality: Platforming Physics, Collsion Detection
    - Implementing tools/power ups use: Ice Pick Mechanic, Ziplines, etc.
- **Josh**
    - Spritework
    - Level design: Procedural Generation of platforms and Walls
    - Menus: Title screen, Mode Select, Options Menu, Character Select
- **Andrew**
    - Sound/Music/Spritework

- Controller Functionality: 2 controllers, with D-pad, A, B, Start, Select functionality
- Animation: Traditional sprite animation: replacing sprites during different parts of animation