

Game Proposal: Minos' Monster Maze

CPSC 436D - Video Game Programming
Spring 2018/19

Story:

A 2D platformer game where you play as Theseus and have been transported into a digital Greek labyrinth hounded by classic monsters from Greek mythology (including the infamous Minotaur). You must escape the various levels of the maze before the Minotaur catches you! Basic keyboard input controls include moving left/right and jumping. A twist to the game is that you can rotate the screen to get to areas you may otherwise not be able to reach.

Technical Elements:

The game will use geometry for the platform, sprites for characters, and sound bites for the different interactions within the game. The design of the mazes for each level will be hardcoded into the game. Transformations will be used to move characters and rotate the platform. The rotations will be about the player character's z-axis. During rotation, the characters will be frozen until the rotation is complete.

There will be collision detection between the sprites and the platform, and between the sprites themselves. The monsters will be chasing the player. Upon game update, monsters recalculate distance to hero and determines the shortest path to get to the hero. The monster speed increases for each level succession. Game physics include gravity for jumping and falling objects/sprites.

Advanced Technical Elements:

Player has free rotational control of the map (highest priority)

- *impact: players will feel restricted, less control of the game*
- *alternative: player can only rotate map by 45 degrees*

Enemy AI uses a search algorithm to find the shortest path to the hero upon every game update call. After rotation, enemies' path finding will be updated and then move in the direction that the algorithm specifies (left or right). Difficulty can be scaled between levels either by increasing enemy speed and/or reducing frequency of random idling times.

- *impact: this would result in slightly easier gameplay for the user with less advanced enemies*
- *alternative: only have monsters that move back and forth on predetermined path and a boss with simple AI behaviour/patterns*

Platform Surfaces

Addition of spikes, icy platforms, fading platforms

- *impact: less exciting elements to the game*
- *alternative: do not implement*

Saving Game State

If players decide to close the game, they can start from the level they just completed. After finishing a level, generate a code that players can input to get unlock and start from that level again

- *impact: players will be annoyed if they reboot the game and have to start all over*
- *alternative: players must start from the beginning of the game*

Physics to calculate whether a character can keep walking up a platform that is rotated or if it will slide down

- *impact: excluding this from the game will not impact gameplay by much, but will simply make character movement less realistic and will make different platform surfaces not possible to implement.*
- *alternative: a threshold will be hardcoded so that characters can only walk up platforms that are slanted less than a specific degree*

Power-ups / Rotation Energy Gauge (Lowest priority)

Examples of power-ups: invincibility, ability to freeze enemies

Rotation Energy Gauge: based on how many points collected = how many times you can rotate

- *impact: less gameplay mechanics*
- *alternative: do not implement*

Devices:

Keyboard controls player movement and menu selection.

left arrow = move left

right arrow = move right

spacebar = jump

z = rotate clockwise

x = counterclockwise

h = open and close help/instructions menu

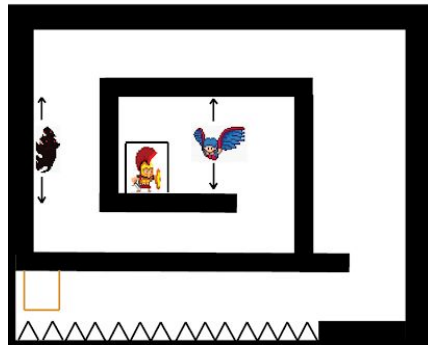
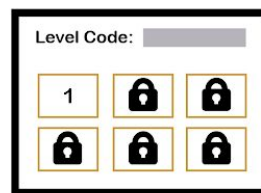
esc = pause game (with option to go back to main menu screen)

Concepts:

Main Menu

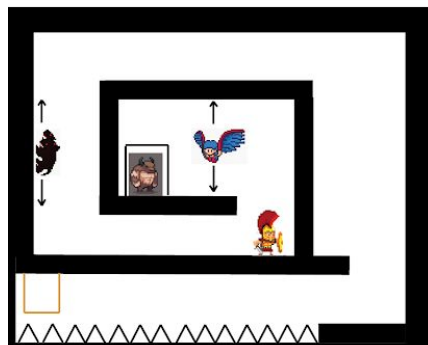
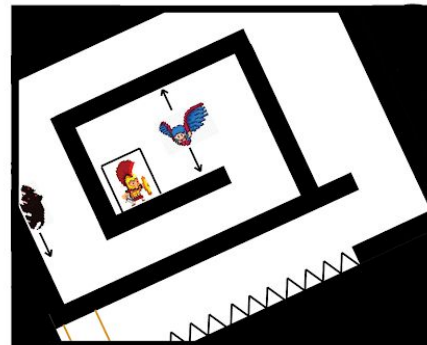


Continue Screen

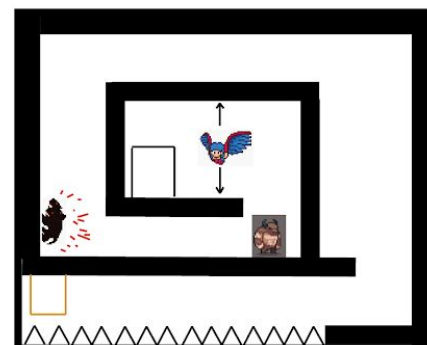


level start - simple enemies patrol back and forth

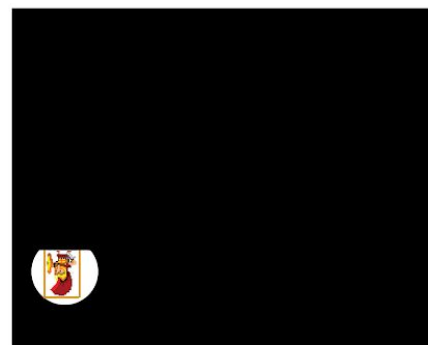
rotate
→



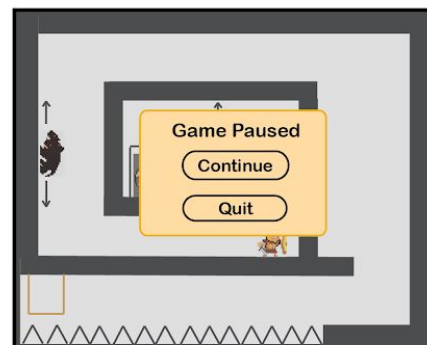
minotaur chases player after a certain period of time / the player has reached a certain point of the map



death by encounter with enemy (player explodes into bits)



circle of visibility shrinks to player when player reaches the exit



Game paused state

Tools:

We plan on using a Pixel art editor and colour palette tool for generating art assets.

Also, to maintain a consistent aesthetic to the game, we plan on using a colour scheme chooser (such as <https://color.adobe.com/>)

We might use Audacity if needed to mix sound clips (if we need to do anything too complex to handle in code).

Development Plan:

Week: February 1 - **Skeletal Game**

- Layout basic class hierarchy/structure
- First level of game (maze design)
- Player controls
- Basic collision detection
- base sprite assets

Week: February 8

- Add simple 'ghost-type' enemy AI (not individually controlled, can go through walls)
- State diagrams for enemy AIs
- Basic physics (gravity)
- One advanced maze designed and coded (basic platforms)

Week: February 15

- Implement rotation controls
 - Background music
- Whispers say it's reading week...*

Week: February 22 - **Minimal Playability**

- Basic enemy AI (following the exact path as the hero or, alternatively, going on a fixed path)
- Tutorial (press H for instructions)
- At least two levels of the game designed (difficult enough to sustain at least 2 min of game play)
- Sprite animation

Week: March 1

- Textures applied to most maze components
- Add rolling boulders (geometry)
- Basic main menu screen
- Plan for various platform surfaces + character's reactions to surfaces

Week: March 8 – **Playability**

- Platform surfaces (spikes)
- More sophisticated collision detection (against geometry)
- 3-4 playable levels (enough to sustain at least 4 min of game play)
- Background physics (exploding gore when player lands on spikes or collides with enemies), may push to following week if needed

Week: March 15

- Implement search algorithm for enemy AI
- Refactor if needed
- One more playable level

Week: March 22

- Physics to calculate if player can continue walking up a platform
- Vary efficiency of enemy AI between levels (make them faster/smarter)
- One more playable level
- Improve calculation efficiency if needed (for lag-free gameplay)
- Testing and bug fixing

Week: March 29 – **Robust Game**

- Tutorial level design
- 5-6 playable levels (enough to sustain at least 6 min of game play)
- Saving game state
- Menu screen that allows starting at any unlocked levels
- Sound effects for most interactions
- Check for and fix memory leaks

Week: April 5

- Dust effects
- Energy gauge for limiting rotation ability
- One more playable level
- Advanced moving and disappearing/fading platforms

Week: April 12

- Implement Power-ups such as invincibility and/or temporarily freezing enemies
- Potentially add lighting effects (from torch sprites in background)
- One more playable level
- Additional elements to platforms such as rolling boulders

Week: April 19 – **Grand Finale**

- Sound effects for all interactions
- Potentially different background music for different levels
- Potentially add prologue scene before first level and epilogue screen after game is beaten
- 7-10 playable levels (enough to sustain at least 10 min of game play)
- Intuitive “tutorial” level progression, levels should be designed so that the player gradually learns and masters their movement, rotation skill, and use of environment as he/she plays through each level.