

Game Proposal: Lumin

CPSC 436D - Video Game Programming Spring 2018/19

Story:

The background story for our 2D puzzle platformer is the main character trying to make it through the dark realm safely. Each level will have a door for our character to switch on and reach in order to travel between the levels. In each level, our character needs to use their gift of light and their ability to manipulate light in various ways to avoid dangers and cleverly complete the level.

Technical requirements:

- **Rendering:** Appropriately rotate/scale large black triangles to create shadows where the player's light doesn't reach. Some obstacle blocks or paths will be rendered when the light is exposed to it and some will only appear when in the dark.
- **Geometric/Sprite/Other assets:** The texture for characters, objects such as buttons, switches, doors, etc, and walls will all be loaded via sprites. Animations for character movements i.e. jumping, idle, walking will be assets as well.
- **2D Geometry manipulation:** Player sprite will need to be translated to move around levels, will also need to detect collisions by the player with walls/floor/ceiling.
- **Gameplay logic:** Need logic to move the player around and allow it to jump. Also will need logic doing something similar to raycasting in order to determine whether interactable objects are in the player's line of sight.
- **Physics:** Characters will need to detect when they have collided with walls/floor/ceiling and stop when they have, will also need some rudimentary physics to deal with gravity/jumping
- **Advanced:** AI will be used for various mechanics powering idle characters to help solve the level which will require pathfinding.
- **Audio:** Sound effects might be used when the door has been switched on and some background music.

Advanced Technical Requirements:

List the more advanced and additional technical elements you intend to include in the game prioritized on the likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative

- Mirrors to bounce the light from the player will be used to solve puzzles but skipping this feature would not affect our gameplay
- Fireflies that are drawn to light and themselves light up the surrounding
- NPCs as enemies of the dark but if they are not included, we will have more than enough mechanics for our levels

Devices:

Explain which input devices you plan on supporting and how they map to in-game controls.

- Will support keyboard input
- W,A,S,D for movement
- J,K,L to switch between different filters for light (none, flashlight, laser)
- R to reset the level
- Mouse to direct the lights when filtered

Concepts:



Basic layout of the game. The switch toggles the light for the door, indicating to the player that they can progress to the next level.



Example of one of the different lights that can be used in the game. A directed light to show obstacles (blocks). Some blocks (white) allow light to pass through and some (grey) obstruct light.

Tools:

- No other libraries have been chosen besides OpenGL

Development Plan:

Tasks for each week. Account for testing, potential delays and alternative options (Plan B)

Week: February 1 - **Skeletal Game**

- Basic character movement (wasd, jump)
- Designs for some puzzles
- Basic collision (player with walls / floors)
- Sprites for a few key elements (player, common objects like blocks, switches, door)
- Basic level with static walls, switches (doesn't have to do anything yet)

Week: February 8

- Basic platforming
- Basic lighting system from player (aura of light around player, flashlight, laser) but doesn't necessarily interact with any other objects
- Design for more puzzles, designs combining several puzzles into parts of levels
- Test level (to show off interactions between light and different objects, walls)
- 1-2 individual playable puzzles

Week: February 15

- Design for level selection
- One/Two basic (tutorial-ish) levels completed
- Working prototype for light reflection / shadow (really basic, does not have to be pretty or robust)
- Objects should detect if they are being hit by light, could be in a basic way

Whispers say it's reading week...

Week: February 22 - **Minimal Playability**

- All "light types" roughly implemented - should have 3 basic levels completed
- Object can detect if they are being hit by light
- Core mechanics implementation complete - mirrors, switches, movement between levels, collisions with re-appearing blocks
- Start work on implementing AI characters - fireflies, characters that do specific tasks, any enemies
 - This can always be slightly delayed or further simplified if other tasks are taking longer than expected

Week: March 1

- User testing on the completed levels
- Design more levels based on user testing
- Start discussion and design Start Screen
- Add moving walls / floors
- Start making animations for player movement, jumping, falling, idle
- Start making/finding sounds fx

Week: March 8 – **Playability**

- Set of puzzles that rely on different "light types"
- Object can detect if they are being hit by light, taking into account blocking walls / shadows

- Core mechanics should be free of obvious bugs
- AI implementation complete and tested
- Simple sounds when things happen (light mode switching, switch getting lit, door unlocking)

Week: March 15

- Level selection implemented
- Transitions between scenes
- Implement Start Screen
- Finish player animations

Week: March 22

- Nail down level progression to feel good to player while unlocking “light types”
- AI implementation bug-free
- Add BGM soundtrack

Week: March 29 – **Robust Game**

- Shadows look good
- Objects can detect if they are being hit by light in a robust and in a way that feels natural
- Light and shadows bug-free
- At least one level that fully uses every mechanic in the game
- Complete user flow from start screen to level select to levels
- Level designs that incrementally and intuitively teach game mechanics
- Fun puzzles

Week: April 5

- Finalize art and music, robust user testing

Week: April 12

- Final touches to fix any minor bugs

Week: April 19 – **Grand Finale**

- FOR REAL Controller support
- Investigate third party engines / tools that support extra lighting options (ambient light, bloom etc.)