Minimally Playable Game: Stealthy Sam

CPSC 436D - Video Game Programming

Spring 2018/19

Team Members:

Caroline Chang	20044146	d6g0b
Sam Chow	27453142	i6x9a
Mah-dir Sheikh	36718154	p4u0b
Sam Veloso	33140154	m6o0b
Sam Goldwax	13433157	u9k0b
Nicolas Gagnon	26528142	l4f0b

Milestone Requirements:

Sustain progressive, non-repetitive gameplay using all required features for 2 min or more (assume that you can provide users with oral instruction).

- Player is capable of playing the first three rooms consecutively and restart the game after collision with enemy. In the first room, the player has to avoid the enemies in order to reach the key in the chest. After acquiring the first key, the player has to continue to avoid enemies and proceed to the second room. Once there, there will be a key situated on the dinner table to be collected. The player will then continue onto the final room, the 'kitchen', where a key will be hidden in one of the cupboards. Any collisions with an enemy will result in the player's death and they will have to restart from the first room. Once all 3 keys are collected, then the player will be able to continue to through the locked gate and the minimal playability will stop there. This should last approximately 2 minutes and will be expanded upon on future iterations.

You should implement state and decision tree driven (possibly randomized) response to user input and game state (create a decision tree data structure and reuse it for multiple entities)

- Our enemies have state and decision tree driven data. The ghosts are initially patrolling. Once the user is on the left side of the screen, the ghosts start chasing him. And if he hides in a closet, the ghosts stop chasing him and return to their original patrols.

Provide extended sprite and background assets set as well as corresponding actions.

- There are a lot more sprites from different open source sprite sheets that were added to match our original ideas for each of the rooms.

Provide basic user tutorial/help.

- There is a text box telling the user they can use WASD to move, which disappears once they start moving. In addition, once that disappears, there is text saying they can press E to interact with objects. And when they die, there is text showing they can press R to restart.

Stable game code supporting continuing execution and graceful termination

- Our code can be continually executed without bugs and we are able to terminate the program without any problems.

Creative:

You should implement one or more additional creative elements. These can include additional assets, rendering effects, complex gameplay logic, or pre-emptive implementation of one or more features from subsequent milestones.

- We have implemented a light dimmer that makes the player's surroundings more darker.
- Extensive sprites to match our original level designs.
- Refactored the original code to model the ECS design.
- Added additional interactable closets in each of the levels.

Your submission should align with your proposed development plan: Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies.

In our original plan, we had these features planned for completion for this milestone:
Week: February 22 - **Minimal Playability**

- World shader
  - We have this implemented, and our character successfully travels with and takes a hole in the world shader with him
- Start implementation on interactables
  - We have currently keys implemented, and closets you can hide in for interactables.
- Level layout
  - We have 3 rooms laid out of the level.

We made all of our major goals for the minimal playability milestone.