

# Game Proposal: False Alarm

CPSC 427 - Video Game Programming

Winter 21/22

## Team Members:

Shiyu Chen, #79146593

Yi Xuan Qi, #95411336

Zipeng Liang, #28699023

Kehong Liu, #38971404

Andy Chen, #88923800

Jingyang Sun, #48346902

## Story:

*False alarm* is a single-player stealth/survival and puzzle game. The player wakes up in the morning and soon realizes the exam is today and not next week. There is no way the student can pass the exam so he/she decides to pull the fire alarm to delay the exam (We do not recommend this, this is just a game). The goal of the game is to escape from camera and security guards and solve puzzle games along the way to unlock paths as the difficulty level increases. The player will eventually reach the final exam hall and pull the fire alarm. The ending of the game is that the entire game is just a dream, the student wakes up and finds himself sitting in the middle of the final exam.

## Technical Elements:

*Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI, physics.*

- Rendering:
  - pixel art for entities and backgrounds
- assets (geometry, sprites, audio, etc.)
  - The game will feature 2D assets
  - There will be intractable components in the map (doors, cabinets, alarm, locks etc.)
  - There will be different set of assets for mini-games (used for unlocking the door to the next level)
- 2D geometry manipulation
  - Top-down perspective
  - Multiple rooms and players need to advance through the room to the final room to pull the fire alarm
- gameplay logic
  - Character pathfinding: up/down/left/right or w/s/a/d key can be used to control the direction of movement of the main character
  - Damage: The player initially has three points. Each time the player gets caught by the security/security cameras, the player loses a point and gets a warning to go home.
  - Death: After two warnings, the player will die (sent to the dean's office) the next time the player gets caught.
  - Walking past an items will automatic pick the item, each character can have at most 3 items at the same time
  - Items have special effects that can heal the player from death, camouflage, etc.
- AI
  - Enemies will follow a predetermined path to patrol around the hallway.
  - Security cameras will change FOV with a predetermined pattern at fixed locations.
  - There are traps in the game where if the user steps on it, it will make sound and attract security.
- Physics
  - Characters will need to detect when they collide with walls and stop.
  - Enemies will need to detect when a player collides with their FOV and terminate the game if it does.

## **Advanced Technical Elements:**

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

- Physics element: some items can have the ability to deflect light from the security guards' flashlights.
  - Impact: removing this element reduces the variety of game mechanics and potentially make the game more difficult as the player has one less move against the enemies
  - Alternative: player simply avoids the flashlight beams, instead of deflecting them.
- Level selection: later level only unlocks upon completing the preceding levels
  - Impact: removing this element will make the learning curve steeper as the game difficulty and number of game mechanics increases level by level
  - Alternative: allows the player to freely choose levels
- Game saves: save the player's progress within a level (least priority)
  - Impact: removing this element will make the game less flexible and player would have to complete the levels in one sitting
  - Alternative: reset the player's progress to the beginning of the level every time they leave the game
- Security guards use a shortest-path algorithm to try to get closer to the player once alarm is triggered
  - Impact: removing this element can make the gameplay easier
  - Alternative: security guards can walk along a random path to try to catch the player

## **Devices:**

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

We support keyboard input and different keys are matched with different action of the character:

"a" - character moves in direction left

"s" - character moves in direction down

"d" - character moves in direction right

"w" - character moves in direction up

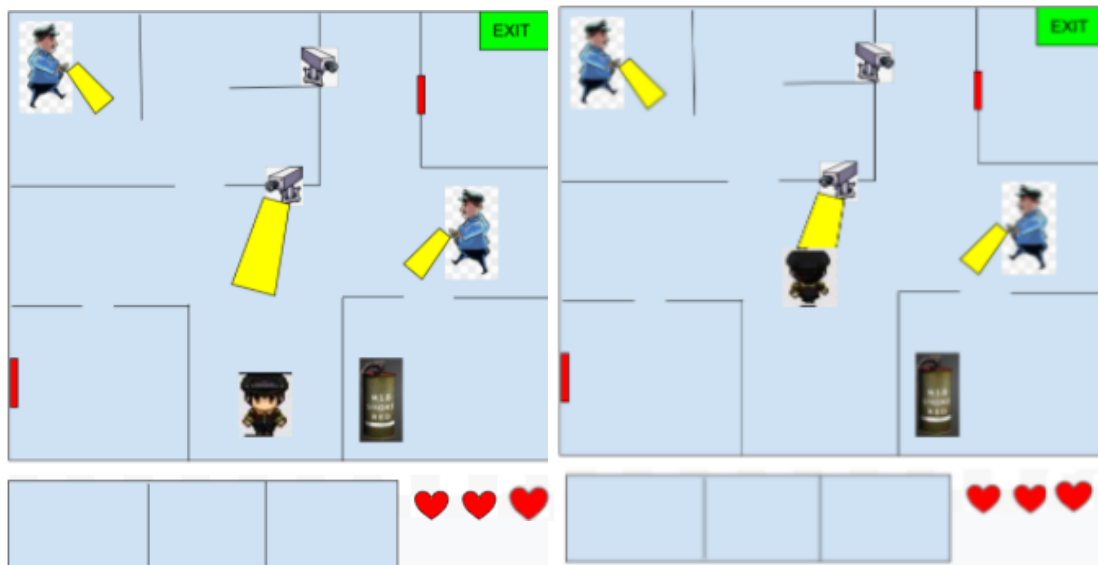
"j" - use game items we picked up

We also support mouse input for navigating through the menu, change basic setting like turning on and off the music and so on

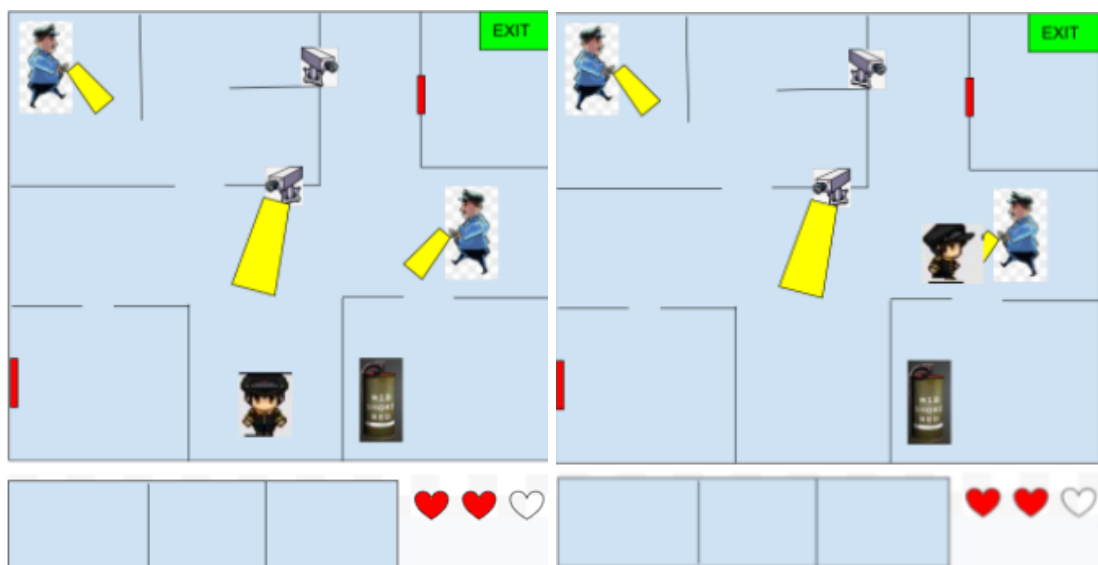
## Concepts:

*Produce basic, yet descriptive, sketches of the major game states (screens). These should be consistent with the game design elements, and help you assess the amount of work to be done.*

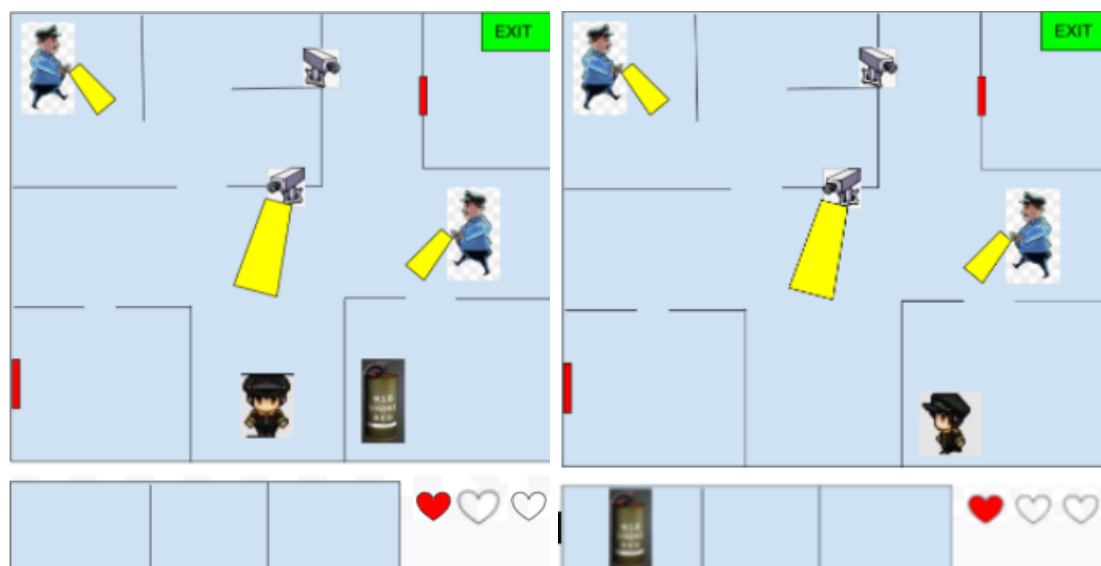
Initial Game State



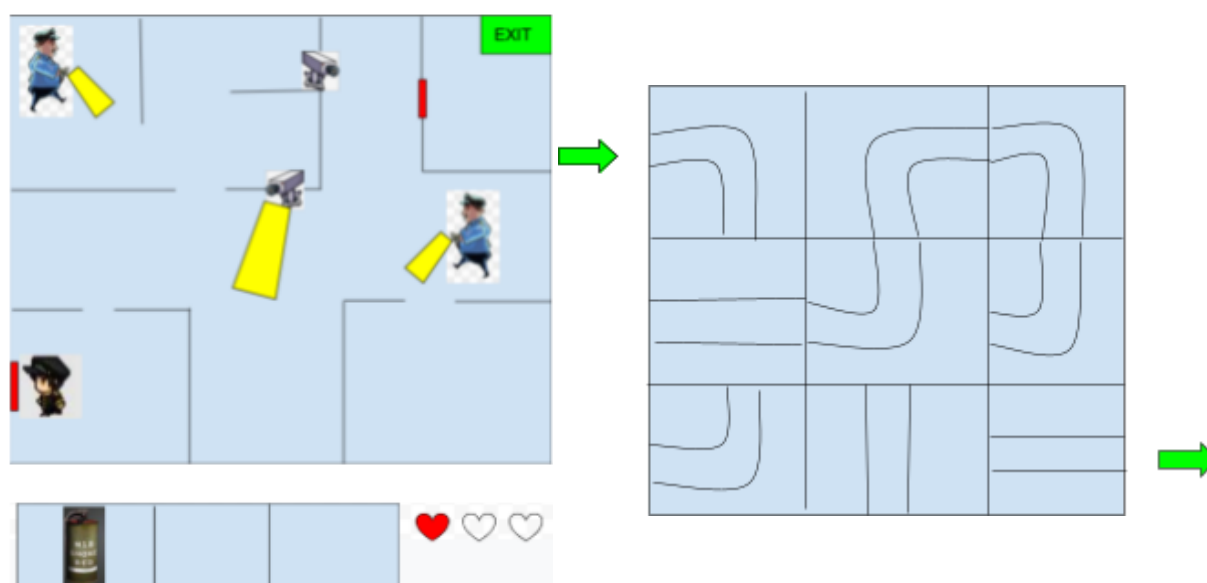
Decrease health by 1, back to starting position



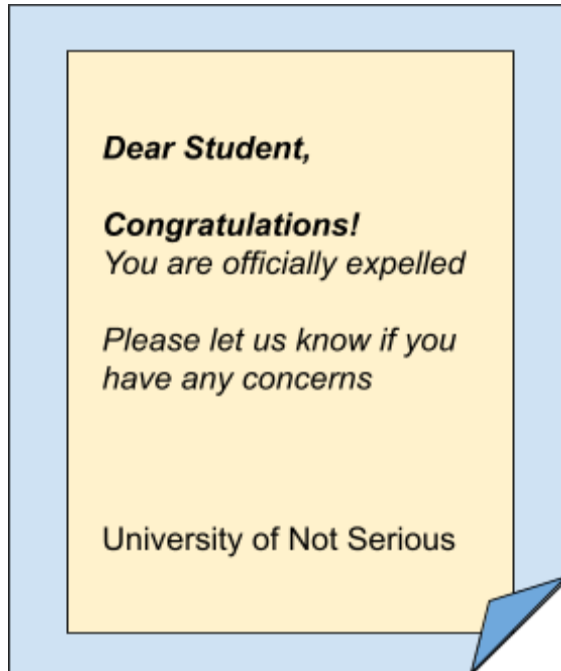
Decrease health by 1, return to starting position



## Collect items



Reach game puzzle to open the gate to exit



You are lost

## Tools:

*Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.*

C++, OpenGL for game development and Pixel art editor for making art assets

## Team management:

*Identify how you will assign and track tasks, and describe the internal deadlines and policies you will use to meet the goals of each milestone.*

We divided the task into mainly four parts. sprite/item design, map design, user interaction design, puzzle game design

For the Skeletal game milestone, we will focus on the first 3 designs. In the first week, it will be mainly sprite design, map design, and user interaction design (approx. 2 people each). We will set internal deadlines to match our development plan. Each member agrees to complete the assigned task on the internal deadline and meet virtually on the day after at 2pm.

We will assign tasks and make adjustments on the progress at the beginning of each milestone. The task assignment for the Skeletal game milestone are set as follows:

### **Internal DDL: Jan. 28**

Shiyu: Basic collision detection (player with walls/floors), Basic character movement (wasd)

Kehong: Basic character movement (wasd), Design for control menu (game level selection)

Yi Xuan: Basic collision detection (player with walls/floors), Basic level map with static walls

Zipeng: Spirit with key element, Basic level map with static walls

Summer: Spirit with key element, Basic lighting system

Andy: Spirit with key element, Basic lighting system

### **Internal DDL: Feb.3**

Shiyu: Set simple path for security guards and collision detection with player

Kehong: Set simple path for security guards and collision detection with player

Yi Xuan: Camera-control, First-level (tutorial-ish) complete

Zipeng: Camera-control, First-level (tutorial-ish) complete

Summer: Design for control menu, Working Lighting reflection/shadowing

Andy: Design for control menu, Working Lighting reflection/shadowing

## **Development Plan:**

*Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code).*

### **Skeletal Game**

Week 1 January 28

- Basic character movement (wasd)
- Basic collision detection (player with walls/floors)
- Spirit with key element (player, security guards/camera, door, fire alarm)

- Basic level map with static walls

Week 2 February 4

- First-level (tutorial-ish) complete
- Camera-control (extra basic feature)
- Set simple path for security guards and collision detection with player (extra advance feature)

## **Minimal Playability**

Week 1 February 11

- Basic enemy AI (enemy going with fixed path)
- Design puzzle game to unlock each level
- Intro-level maps complete
- Start to make Sprite animation
- All light object (player, security, camera) implemented
- Basic lighting system from security guards/camera
- Working Lighting reflection/shadowing
- Design for control menu (game level selection)

Week 2 February 18

- Designs for more puzzle
- Medium level map in progress
- Apply texture to most background components
- Basic main menu screen
- At least two levels complete

Week 3 February 25

- User testing on completed levels
- Designs for more level at puzzle base on testing
- Sprite animation complete
- Add background music

## **Playability**

Week 1 March 4

- Design usage of game items
- Handle unexpected user input



- Check resolution for and aspect ratio consistency
- Start implement game items
- Improve player/enemy animation to be more smooth

#### Week 2 March 11

- Implement search algorithm for security guards to chase the player
- More game levels complete
- Time-stepping based physical simulations for jagger waves
- Finish implements game items
- Implement more complex geometry in game items/light to detect collision

#### Week 3 March 18

- Test memory usage and efficiency
- Locate runtime bottlenecks and resolve lags
- Save game state for user
- Game items implementation are bug free
- Support continuing execution and graceful termination

### **Final Game**

#### Week 1 March 25

- Finalize different background music and sound effects of interaction
- Fix all bugs identified in prior milestones
- Support continuing execution and graceful termination, test unexpected behavior
- Add additional levels, new maps and bonus animations
- Add tutorial option in the menu

#### Week 2 April 1

- Finish all game levels
- User testing and make changes based on feedbacks
- Integrate with external libraries to improve gaming experience
- Implement advanced 2D dynamic shadows

#### Week 3 April 8

- Make trailer video for the game
- Implement advanced gameplay, when the character make noise, securities will chase the character in an optimal path base on search algorithm
- Add audio feedback, for example alarm when securities notice the character, and more intense background music when the user are closer to their destination