



De La Salle University – Manila

Gokongwei College of Engineering

Electronics and Communications Engineering Department

# **CPE ELECTIVE 2**

## **LECTURE / LABORATORY**

### **CPECOG2 / LBYCPH3**

## **Project Documentation**

Pepe Maze Craze Game Design Document

by

BAYETA IV, Reginald Geoffrey L. - 11811269

PANES, Michael Ryan B. - 11813911

TIU, Emerson Karl M. - 11838000

CPECOG2 – EQ2 / LBYCPH3 – EQ4

September 24, 2021

## I. Game Overview

The game is entitled **Pepe Maze Craze** which follows the player traversing three (3) stages of original classic mazes, where the player takes the form of the character Pepe the Frog, a popular meme icon that originates from Matt Furie's *Boy's Club*. In every stage, the player's starting position is in a green area, and the finish line is represented by a red area. By using arrow keys, the player can move in four directions ( $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$ ) except if there is a wall in front, and the center key is used to reset the position of the player to its original coordinates with respect to the current stage. In addition, the player can interact with two objects, shrink (-) and grow (+) sprites, where the former makes the player's sprite smaller, and the latter does the opposite, provided that the current size of the player is opposite of sprite they are trying to interact. Moreover, in traversing the maze, the player has limited vision due to the presence of fog of war which makes the game more challenging and fun.

## II. Game Specifications

### a. Characters

Table 1. *Player main sprite (small): pepeSmall*






Facing right	
Facing left	

Table 2. *Player main sprite (big): pepeBig*

Facing right	
Facing left	

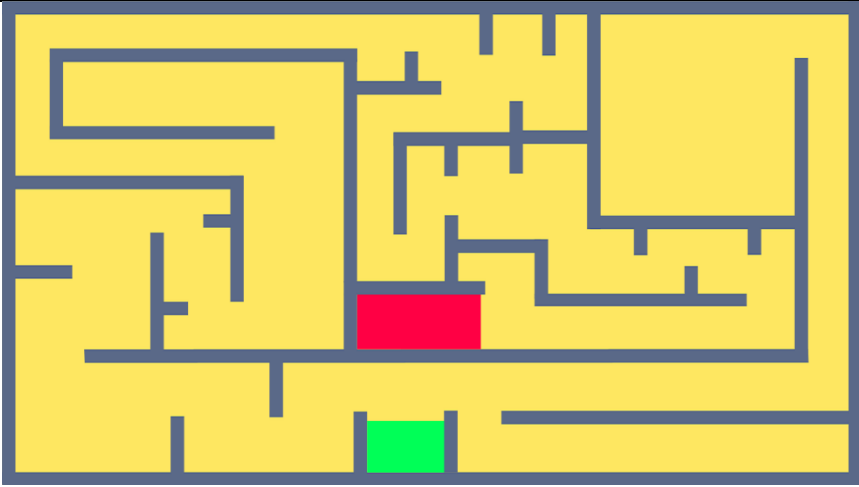
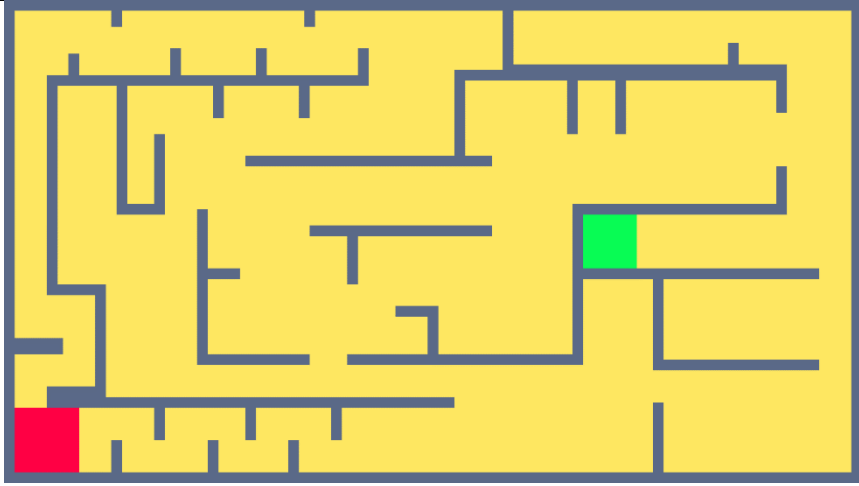
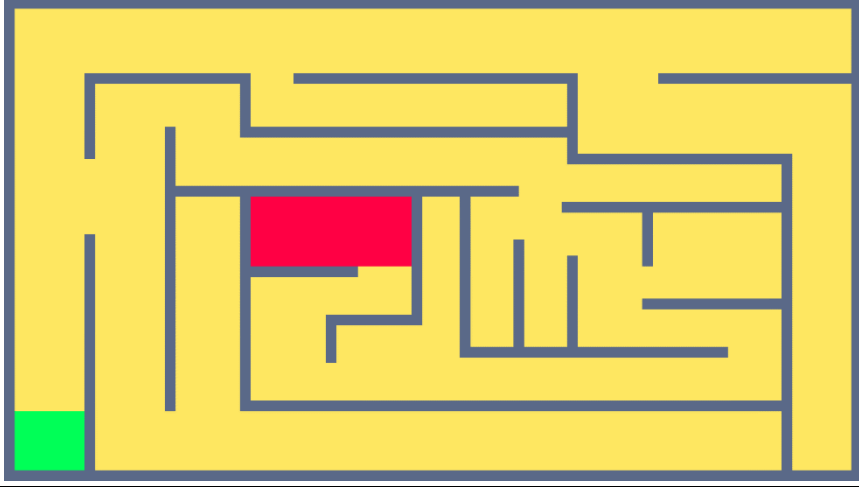



### b. Interactable Objects

Table 3. *Grow and Shrink sprites*

Grow	
Shrink	

### c. Stages

Table 4. *Stages*

Stage 1	
Stage 2	
Stage 3	
Legend	
	Starting Position
	Finish Position
	

d. Gameplay

i. Controls

- BUTTON\_LEFT - Move sprite in west direction
- BUTTON\_RIGHT - Move sprite in east direction
- BUTTON\_UP - Move sprite in north direction
- BUTTON\_DOWN - Move sprite in south direction
- BUTTON\_CENTER - Move back to starting position of the current level

ii. Selecting Level



Figure 1. *Game menu and select stage*

At the start of the game, the player will be opted to choose from three (3) stages: *Stage 1*, *2*, or *3* by traversing the character to a specific stage's location. In the hardware version of the code, unlike the one that utilizes *MiniFB* library, the EXIT function is not supported.



Figure 2. *Player selecting Stage 1*

The way select-level works is that it checks if the player's current position is in the area of the stage color indicator (see Appendix B for colors used). If this condition is met, the stage selected will be loaded, and the player is ready to play the game.

### iii. Fog of War

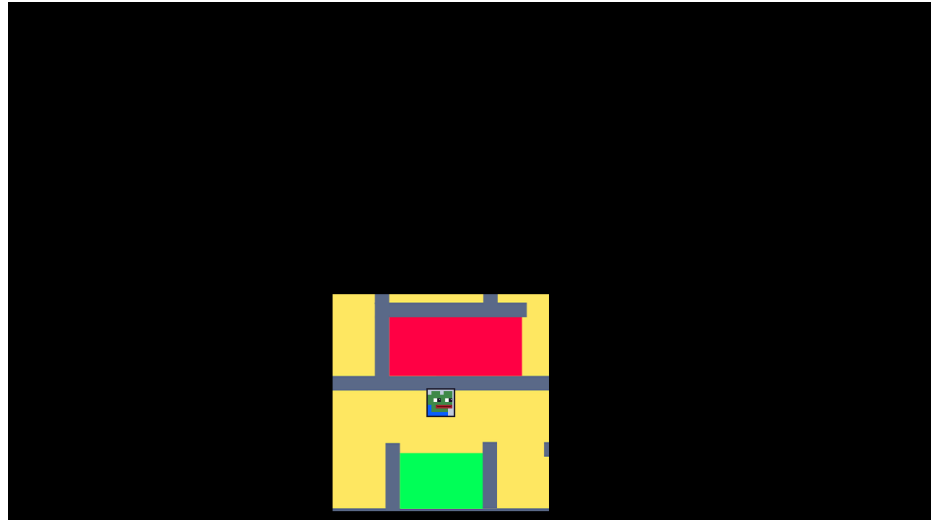


Figure 3. *Actual gameplay: maze traversal with fog of war*

After the player selects a stage, they are tasked to go from the starting position (green area) towards the finish position (red area) to clear the stage. As seen in Figure 3, however, the player will now have limited vision due to the presence of fog of war.

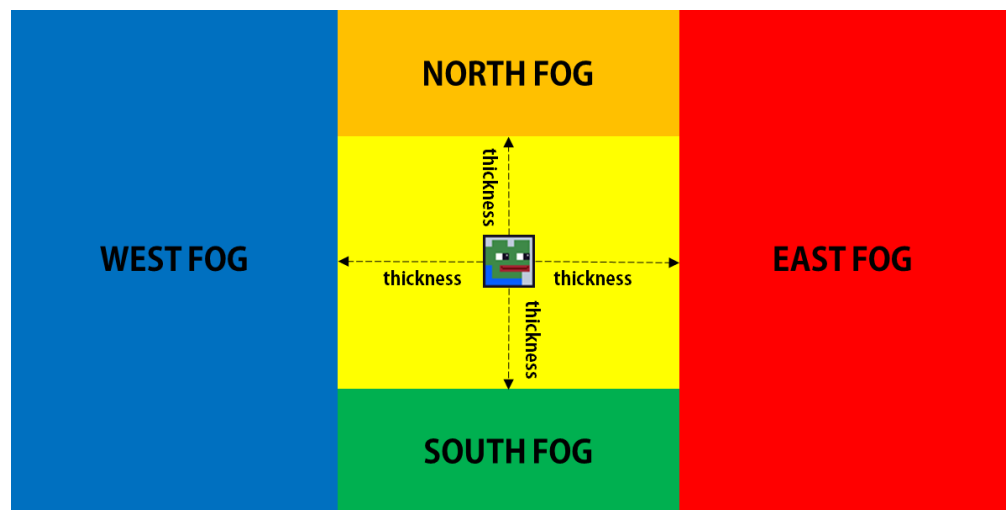


Figure 4. *Structure of fog of war*

Figure 4 details the initial structure of the fog of war in the game. Essentially, it is represented by four rectangles within four directions of the current

player position, wherein the fog thickness represents the field of vision of the player. **For the purpose of this documentation, fogs are colored differently, but it should be noted that the fog in the actual gameplay is colored black** as seen in Figure 3. The actual gameplay equivalent of this structure is seen in Figure 5.

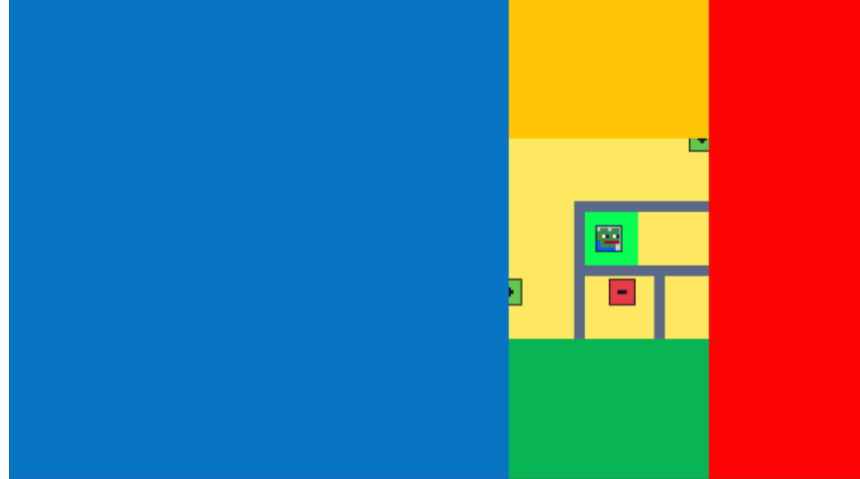


Figure 5. *Actual gameplay equivalent of Figure 4*

As the player moves in a given direction, the fog display will also update wherein the head direction will be updated as background, and the tail will be updated as fog as shown in Figure 6. In this implementation, only the head and tail will be updated when the update fog function is called, and **not the fog as a whole**, provided that the area to be updated is within the boundaries of the screen. The head and tail portions of the fog that will be updated are relative to the direction of player movement. The actual gameplay equivalent of this process is highlighted in Figure 7. Refer to Appendix D for all theoretical and actual gameplay demonstration of fog update for all directions.

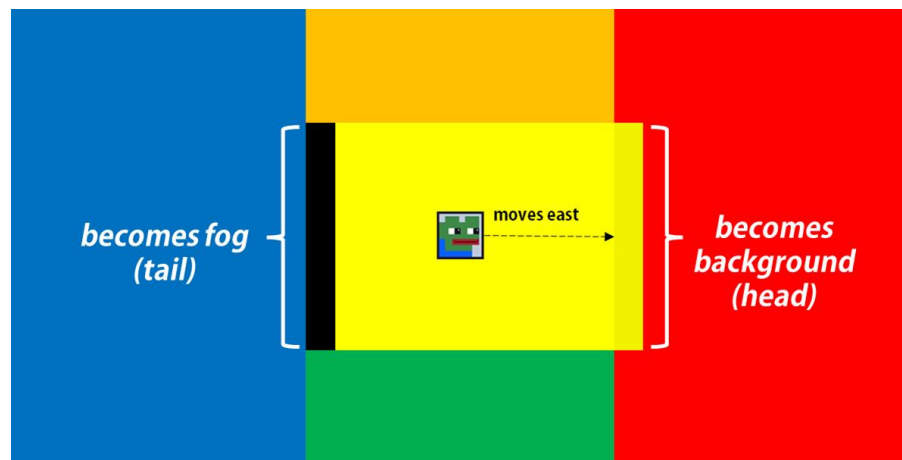


Figure 6. *The theory behind updating fog of war*

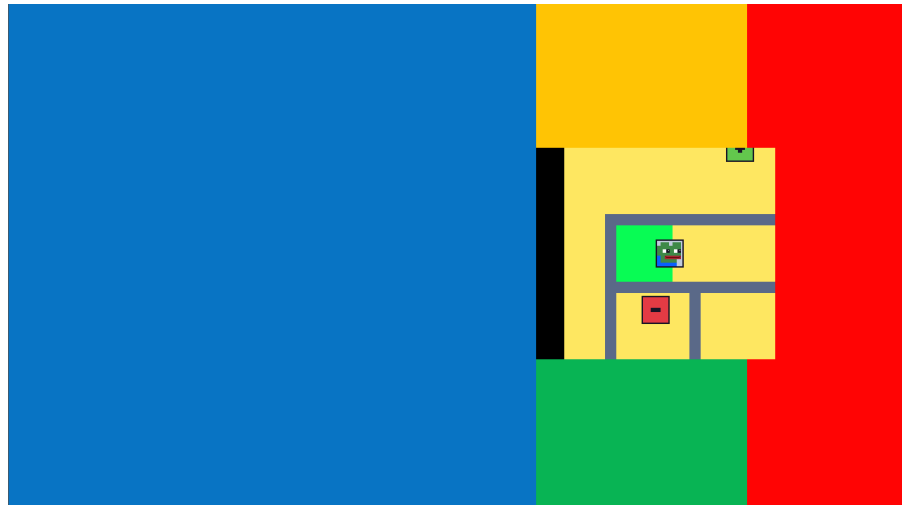


Figure 7. Actual gameplay equivalent of Figure 6

e. Object Collision / Detection

For the object collision and detection of the game, two different types of approach is implemented. For walls, the finish line and the different stages in the game menu, the colors of the objects to be detected are utilized. On the other hand, for grink and shrow sprites, the locations of these sprites are used.

i. Wall Collision

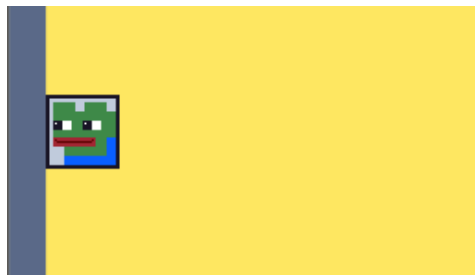


Figure 8. Wall Collision (no space)

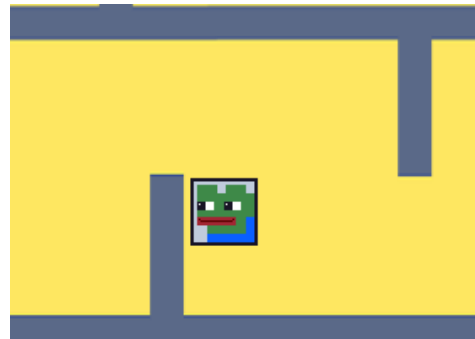


Figure 9. Wall Collision (with space)

The wall collision uses the color of the wall and the future position of the main sprite to detect if the player is able to move. If there are no hex color of the

wall present in the future position of the player, then the player sprite will move to the desired direction. Given that the movement speed of the player in the *MiniFB* is 5 pixels, if the gap between the player and the wall is divisible by 5, the player will be able to touch the wall as seen on Figure 8. However, if the distance is not divisible by 5, a gap which is equal to the remainder of the distance divided by 5 as shown in Figure 9.

## ii. Finish Line Detection

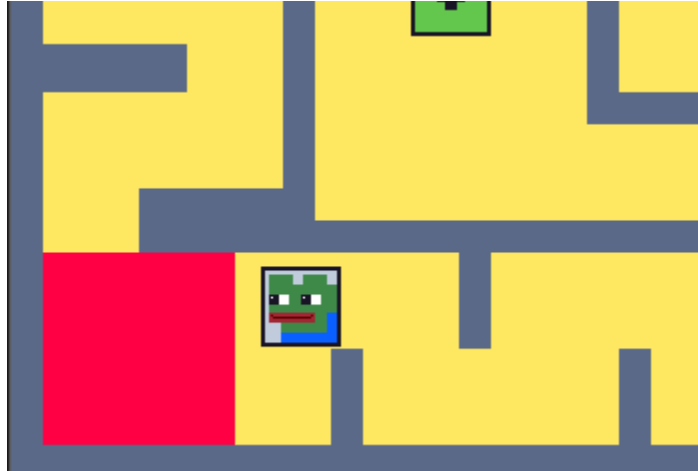


Figure 10. *Finish Line Detection*

Similar to the wall collision, the finish line detection in the game uses the color of the finish line and compares it to the future location of the player if a key is pressed. However, unlike the wall collision where there are cases that a gap between the player and the wall is unavoidable, the player is immediately teleported to the next stage after touching the finish line.

## iii. Shrink and Grow



Figure 11. *Small Pepe Sprite (40x40)*



Figure 12. *Big Pepe Sprite (80x80)*



Figure 13. *Grow Sprite*





Figure 14. *Shrink Sprite*

In order to make the game more interesting, shrink and grow sprites are added to the game which changes the size of the player depending on the type of sprite they touch. These sprites are non-consumables which means the user can infinitely use the shrink or grow given that their current size is the opposite of the sprite they will come into contact with. This means that if the player is currently small, then touching the grow sprite will make the player big but touching the shrink sprite while small will not do anything. The same rule applies if the player is currently big. Unlike the other object detection methods, the collision detection of the grow and shrink sprites utilizes its location and creates a hitbox on that location, if any pixel of the player is within that hitbox, the shrink or grow sprite will activate and immediately take into effect.

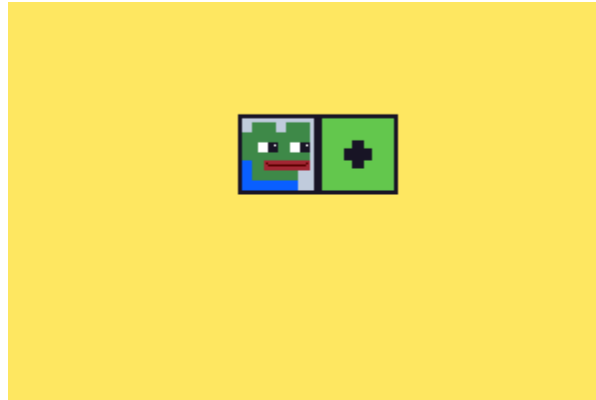


Figure 15. *Small Pepe beside Grow Sprite*

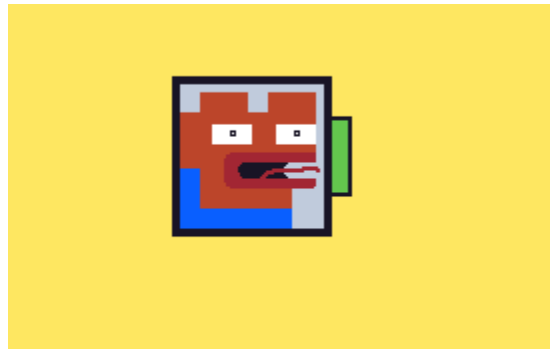


Figure 16. *After Picking up Grow Sprite*

Picking up the grow sprite will double the size of the player where the growth will take affect from the center of the player. This means that by picking up a grow sprite as small pepe, the player will increase their size in all directions as shown in Figure 16. This behavior will cause pepe to get stuck if the player picks up a grow sprite near a wall, they will get stuck and require to reset the stage which is part of the game.

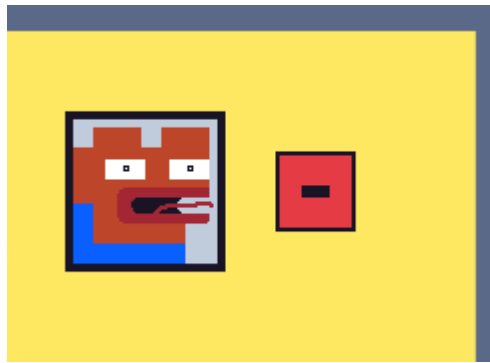


Figure 17. *Big Pepe beside Shrink Sprite*

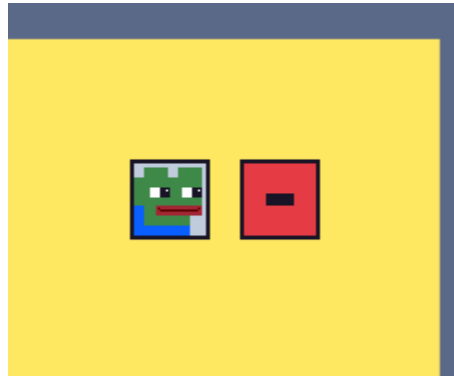


Figure 18. *After Picking up Shrink Sprite*

Similar to the grow sprite, the shrink sprite will shrink the player with respect to the player's center. This causes the player to be separated from the shrink sprite after picking it up as shown in Figure 18. Unlike for the growth feature, the shrink feature is used by the player given that there are some paths in the maze that require small Pepe in order to pass. Refer to Appendix E for the placement of all grow and shrink sprites in all levels of the game.

f. Game Over / Credits



Figure 19. Credits Stage

After the player has successfully cleared Stage 3, the game will now display the credits page that details the names of the developers. In addition, the player can opt to go back to the main menu by going to the MENU display as shown in Figure 19.

### III. Appendix


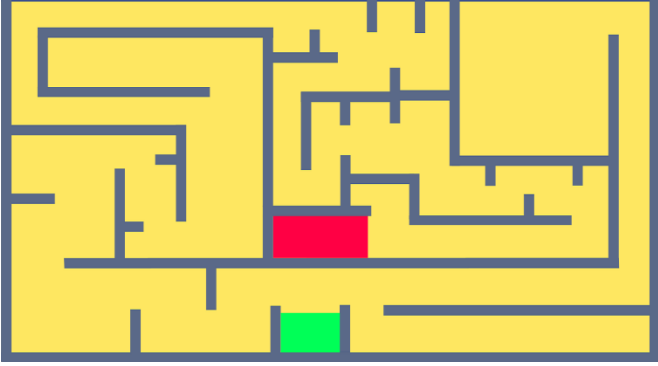
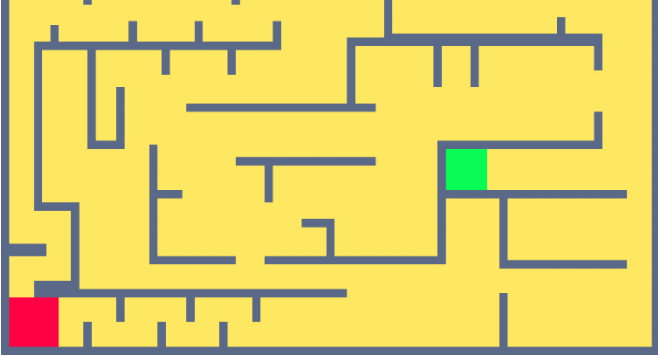
a. Memory map of game objects

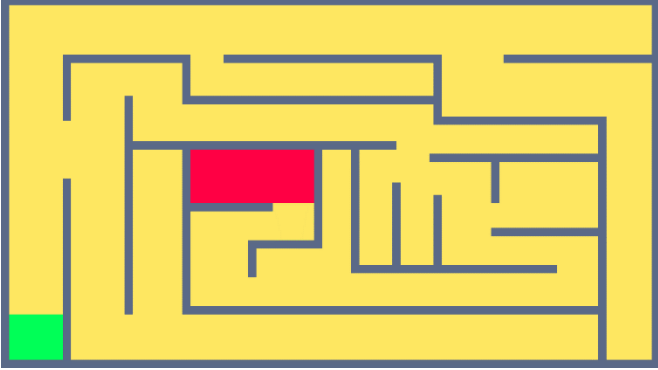







	Start Address	End Address
FRAMEBUFFER	0x10000000	0x108FFFFFFF
BACKGROUND MEM	0x10900000	0x10BFFFFFFF
MENU	0x10C00000	0x10EFFFFFFF
LEVEL1	0x10F00000	0x111FFFFFFF
LEVEL2	0x11200000	0x114FFFFFFF
LEVEL3	0x11500000	0x117FFFFFFF
CREDITS	0x11800000	0x11AFFFFFFF
PSMALLR	0x11B00000	0x11B01FFF
PBIGR	0x11B02000	0x11B06FFF
SHRINK	0x11B07000	0x11B08FFF
GROW	0x11B09000	0x11B0AFFF
PSMALLL	0x11B0B000	0x11B0CFFF
PBIGL	0x11B0D000	0x11B11FFF

b. Color used for interactions

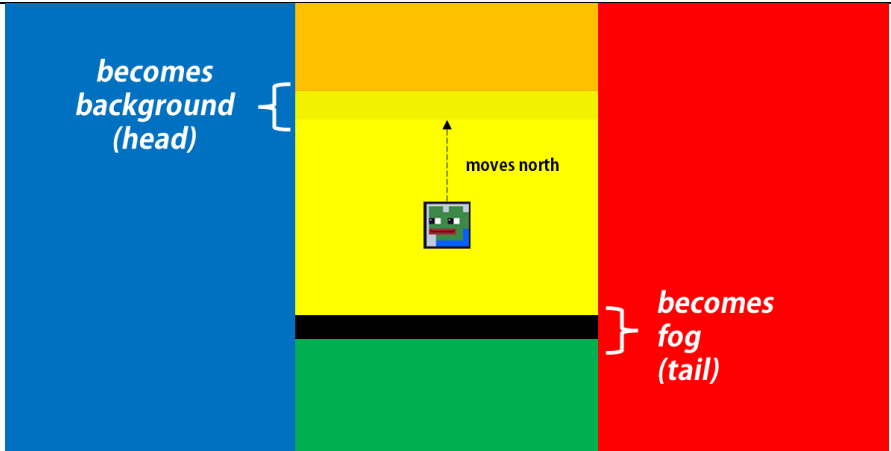
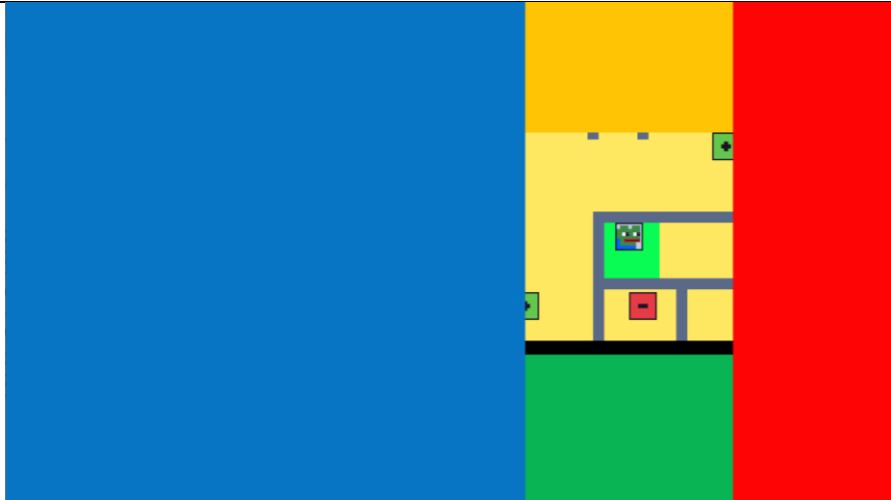
HEX Code	Color	Usage
0x480000		EXIT Program
0x0076ff		Go to Stage 1
0x7a009f		Go to Stage 2
0xf65058		Go to Stage 3
0xff0044		Go to Next Level
0x5a6988		WALL Color

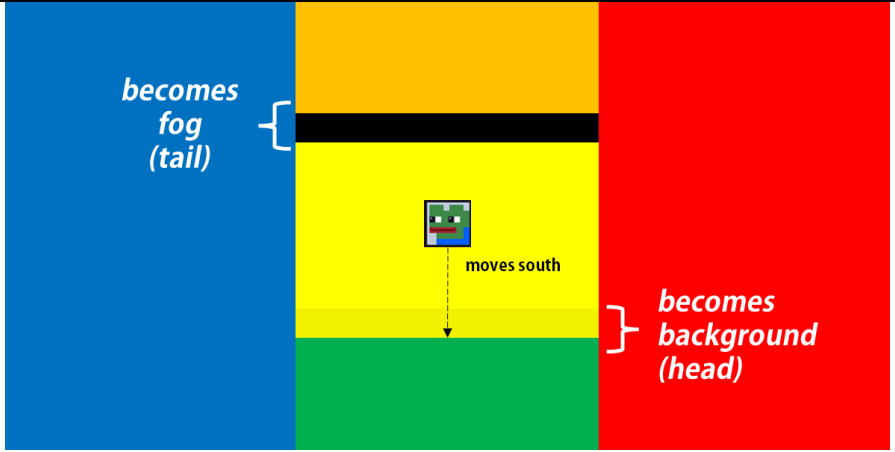
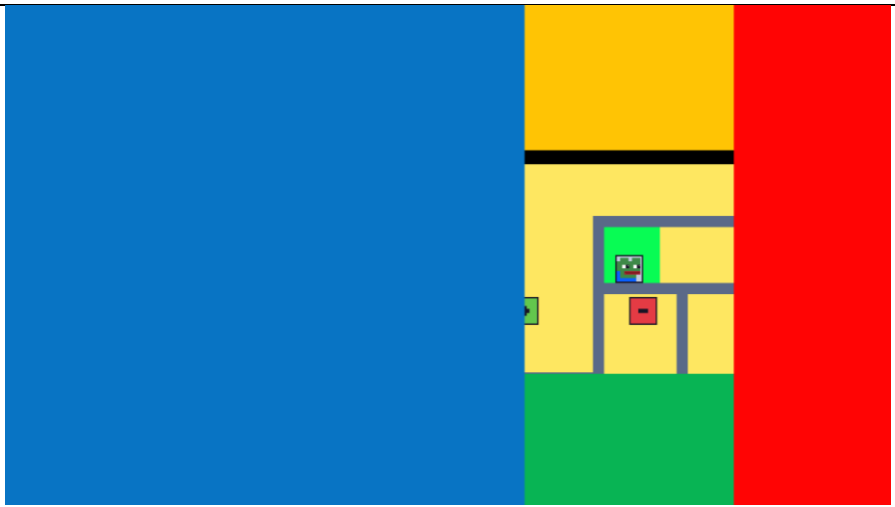
c. Assets

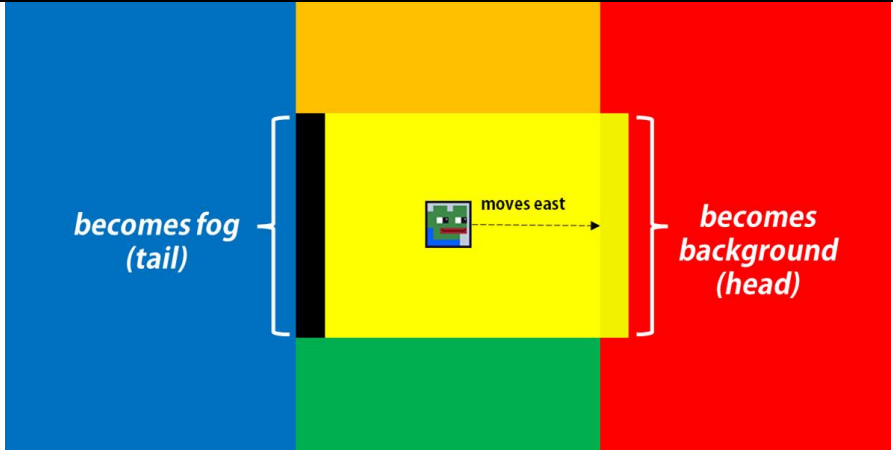
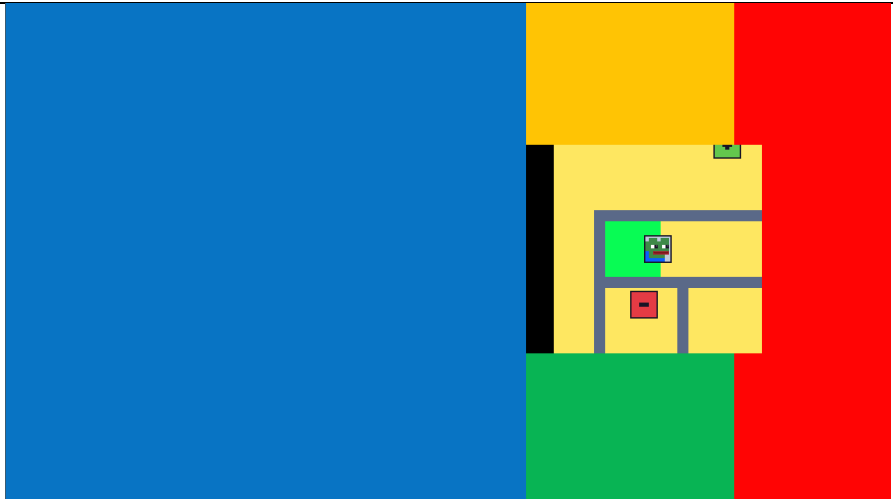
Filename	Dimension	Image
menu.bmp	1280x720	
level1.bmp	1280x720	
level2.bmp	1280x720	

level3.bmp	1280x720	
credits.bmp	1280x720	
pepesmall.bmp	40x40	
pepesmall2.bmp	40x40	
pepebig.bmp	80x80	
pepebig2.bmp	80x80	
shrink.bmp	40x40	
grow.bmp	40x40	

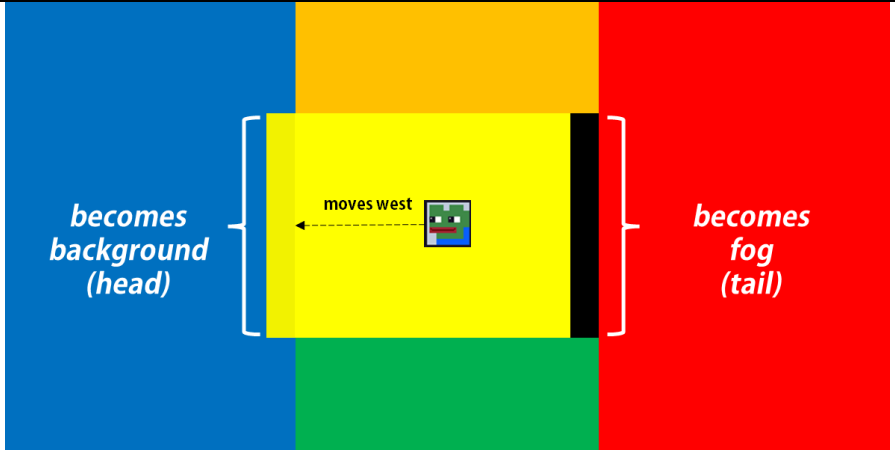
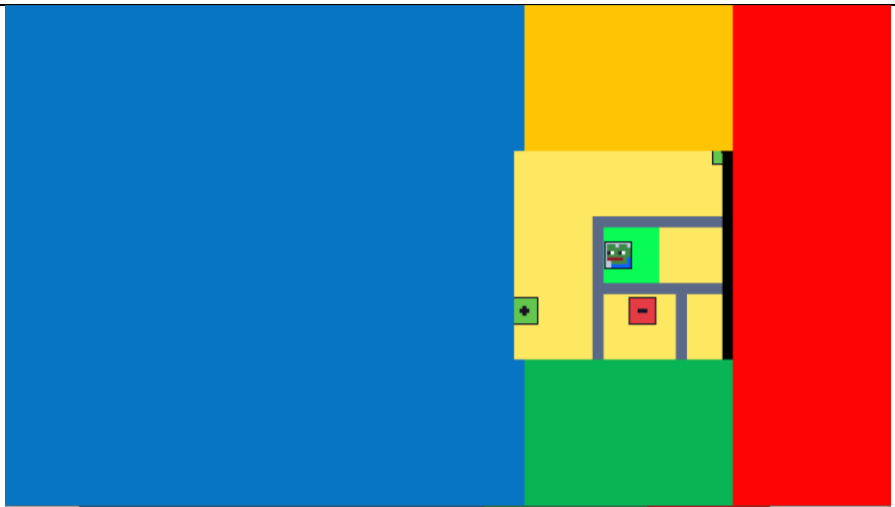
d. *Fog update for all direction*

Fog update when player moves NORTH	
Theoretical	 <p>The diagram shows a 2D grid representing a game world. A player character is positioned in the center of a yellow square. A dashed arrow points upwards from the player, labeled "moves north". To the left of the player, a bracket indicates the area "becomes background (head)". To the right, a bracket indicates the area "becomes fog (tail)". The grid is divided into three main vertical sections: a large blue area on the left, a yellow area in the center, and a red area on the right. The yellow area is further divided into a top yellow section and a bottom green section by a horizontal black line.</p>
Actual Gameplay	 <p>The diagram shows the same 2D grid as the theoretical version, but with a more detailed player character and a more complex environment. The player is now a small figure with a head and body, standing on a yellow floor. The environment includes a green wall, a red door, and a green cross. The fog update is shown as a yellow area that expands from the player's position, covering the area between the blue and red sections. The blue and red sections remain solid colors, representing the background and fog respectively.</p>

Fog update when player moves SOUTH	
Theoretical	 <p>The diagram shows a 2D grid with four colored regions: blue on the left, yellow in the center, green at the bottom, and red on the right. A player character is positioned in the yellow region. A dashed arrow labeled "moves south" points from the player to the green region. A bracket on the left side of the yellow region is labeled "becomes fog (tail)". A bracket on the right side of the yellow region is labeled "becomes background (head)".</p>
Actual Gameplay	 <p>The diagram shows the same 2D grid as the theoretical diagram, but with a more complex layout. The blue region is larger and covers the left side. The yellow region is smaller and contains a player character. The green region is at the bottom. The red region is on the right. The player character is positioned in the yellow region. A dashed arrow labeled "moves south" points from the player to the green region. A bracket on the left side of the yellow region is labeled "becomes fog (tail)". A bracket on the right side of the yellow region is labeled "becomes background (head)".</p>

Fog update when player moves EAST	
Theoretical	 <p>The diagram shows a 2D grid with three main colored regions: blue on the left, yellow in the center, and red on the right. A black vertical bar is on the left edge of the yellow region. A small player icon is in the center of the yellow region, with a dashed arrow pointing right labeled "moves east". Brackets on the left and right edges of the yellow region are labeled <i>becomes fog (tail)</i> and <i>becomes background (head)</i> respectively.</p>
Actual Gameplay	 <p>The diagram shows a more complex environment with a large blue area on the left, a yellow area on the right, and a green area at the bottom. A black vertical bar is on the left edge of the yellow area. A small player icon is in the center of the yellow area, with a dashed arrow pointing right labeled "moves east". The yellow area is divided into several smaller yellow regions by black lines, and a red square is visible in one of these regions.</p>



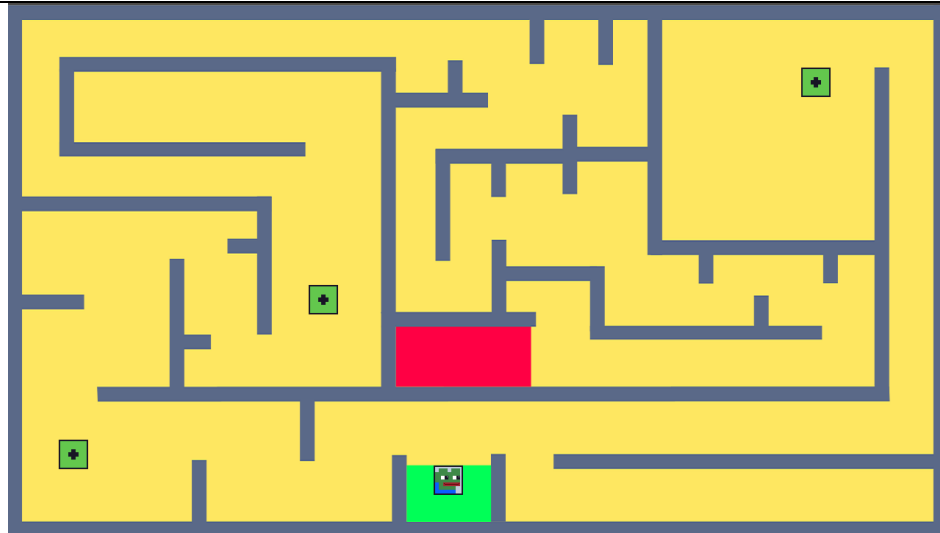
Fog update when player moves WEST	
Theoretical	 <p>The diagram shows a 2D grid with four colored quadrants: blue (top-left), yellow (top-right), green (bottom-left), and red (bottom-right). A player character is positioned in the yellow quadrant, with a dashed arrow pointing left labeled "moves west". A vertical black line is located at the boundary between the yellow and red quadrants. Brackets on either side of this line indicate that the yellow area "becomes background (head)" and the red area "becomes fog (tail)".</p>
Actual Gameplay	 <p>The diagram shows the same 2D grid, but the yellow quadrant is now a complex, irregular shape representing a maze. The player character is located within this maze. The blue, green, and red quadrants remain solid blocks of color. The black line is still present at the boundary between the maze and the red area.</p>

e. Location of all grow and shrink sprites

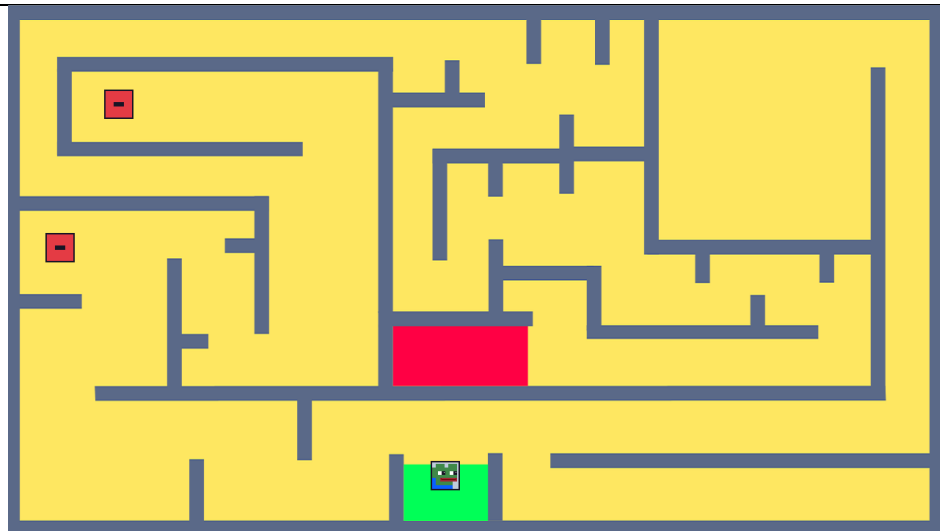
Menu	
Grow	
Shrink	

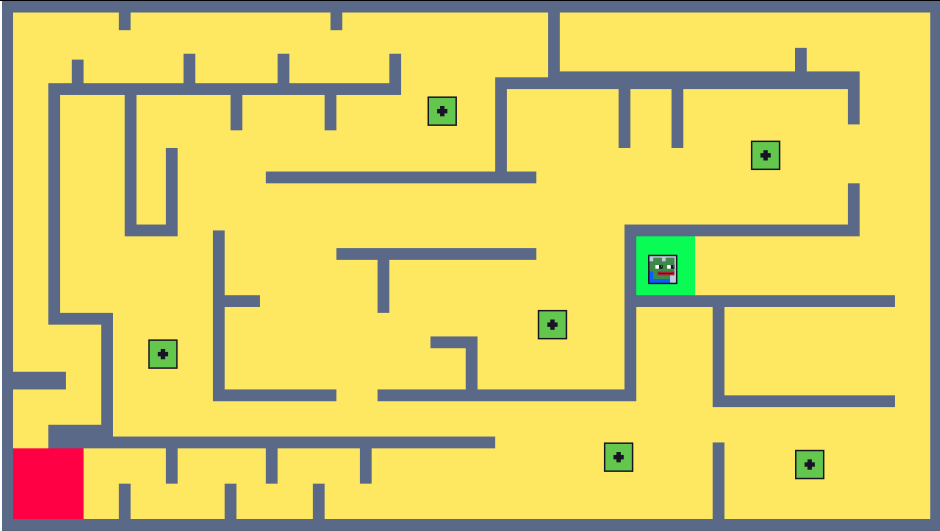
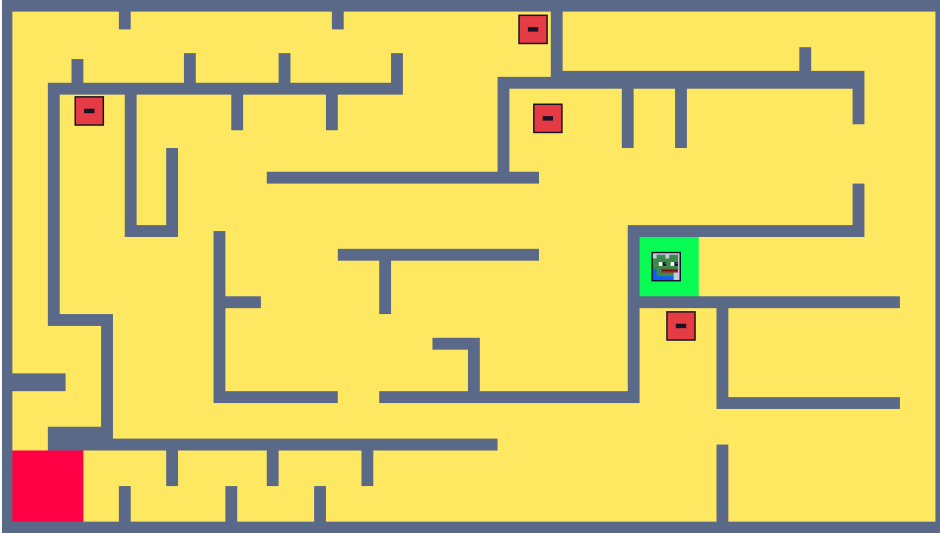
Level 1

Grow



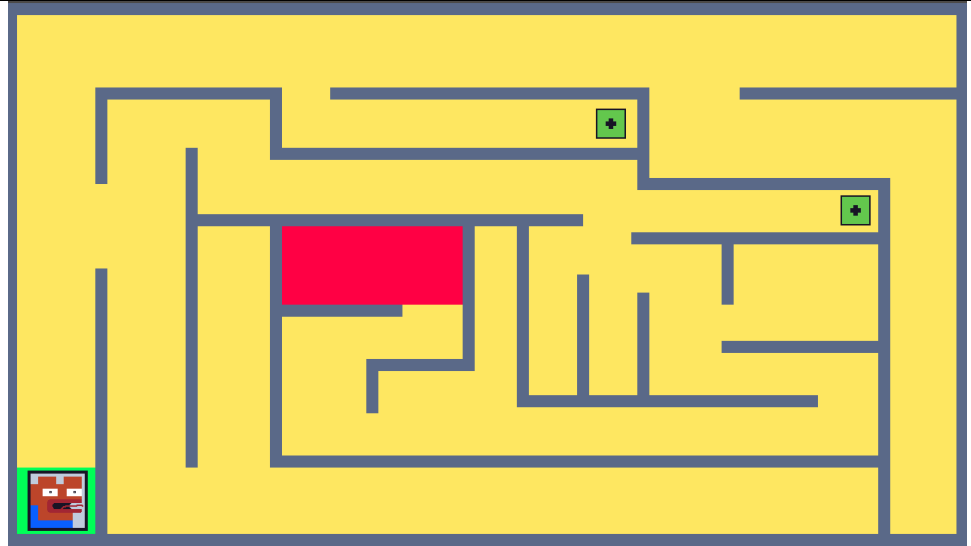
Shrink



Level 2	
Grow	
Shrink	

Level 3

Grow



Shrink

