Oct 14th, 2022

CMPT 276 Project Phase 1
Requirement Engineering and Design

Maze to the Queen Bee

Group 16

Marco Lanfranchi – 301433830
Sana Dallalzadeh Atoufi – 301399601
Satvik Garg – 301452798
Jinshuo Zhang – 301385111

Contents

Game Description

Inspired by an old TV show, Takeshi's Castle, our group came up with the "Maze to the Queen Bee" game. In this game, the player's objective is to move the main bee character to its destination, the Queen Bee, through a honeycombed-shaped maze filled with locked doors, enemies, traps, rewards, and bonus rewards. The player will start at the top of the maze and make their way to the Queen bee located at the bottom by moving up, down, left, or right and navigating the rooms.

As you go through the hexagonal rooms that make up the honeycomb maze, you may encounter enemies who are beekeepers and are always moving towards the main bee to catch it. It is important that the player avoids running into beekeepers and getting caught, or else the game is over, and the player loses.

Some rooms will contain bee traps. Like beekeepers, a player should avoid running into these traps. Instead of terminating the game, running into a bee trap will reduce the player's points by the amount of the trap. If the player's score goes below 0, the game is over, and the player loses.

Rooms will also contain rewards in the form of honey drops. The player must collect all the droplets on the map to finish the game in addition to reaching the queen bee. Honey drops also increase the player's score by a certain number of points. To collect one, the player must simply move their bee character onto the cell containing the reward.

Throughout the game, other rewards will appear and disappear on the map in the form of honeypots. These rewards are bonus rewards, and unlike honey droplets, a player does not need to collect all of them to finish. They simply increase the player's score for the round. The reward amount for a honeypot is substantially higher than that of a honey droplet.

The player is restricted to move the bee around the honeycomb in multiple ways. Firstly, the player cannot go outside the walls of the entire honeycomb. Secondly, the player must move around by moving from room to room in the attached, hexagonal-shaped rooms. Some edges of the hexagon will stop the bee from moving into the hexagon on the other side of that edge, while others will allow the bee to enter the room from the other side. The only way to make this distinction between the edges of the hexagons, which are the doors, is by attempting to guide the player through them.

The player will start with a score of 0 that is displayed in a panel on the game screen along with the total time elapsed for that round. The player must keep their score above 0 to survive by collecting regular and bonus rewards and avoiding bee trap punishments. The player will also have a limited view of the map during gameplay; a circle view slightly larger than one hexagonal room overtop of the character's current position will be all the player can see, allowing them to have a peek at what is in the nearby rooms and full view of the room they are in. After collecting all honey drop rewards and reaching the queen bee, the player wins, and their final score and time are displayed.

For this phase and the next 3 phases of the project, we plan to continue to work together while planning, coding, and testing. For phase 2, we will start by working together to implement our abstract classes, and then divide up the classes that extend those and the other classes among group members to implement while committing their work to the repository regularly. For phase 3, we will each implement testing for the classes that others in the group designed, and for phase 4 we will work together on the documentation.

Use Cases

Use case 1:	Start Round
Primary actor:	Game player
Goal in context:	To start a round of "Maze to the Queen Bee"
Preconditions:	System has been programmed to not begin a round of the game until prompted to do so by the player
Trigger:	The gameplayer presses a specific key to start the round
Scenario:	1. Gameplayer: observes the start screen 2. Gameplayer: presses a specific key to begin round 3. Gameplayer: observes the game map, characters, time counter, and player score to indicate the round started
Exceptions:	Invalid key entered: gameplayer must enter the correct key for the round to begin
Open issues:	Should the start screen display the map with "press the spacebar to start" or a simple welcome screen with "press the spacebar to start"?

Use case 2:	Move Player
Primary actor:	Game player
Goal in context:	Move player through maze
Preconditions:	Will not move until keys pressed; Within the boundary of the map
Trigger:	The gameplayer presses arrow-keys or WASD keys
Scenario:	1. Gameplayer: presses the top arrow key or the W key and moves up 2. Gameplayer: presses the right arrow-key or the D key and moves right 3. Gameplayer: presses the left arrow key or the A key and moves left 4. Gameplayer: presses the down arrow key or the S key and moves down
Exceptions:	Invalid key entered: Player will not move if the arrow-keys or the WASD keys are not pressed Move towards a closed door: Player will not move if they try to move towards a closed door.
Open issues:	N/A

Use case 3:	Player collects reward
Primary actor:	Game player
Goal in context:	Player collects a reward and receives the points of that rewards amount
Preconditions:	Maze contains honey drop rewards/honey pot bonus rewards

Trigger:	The gameplayer walks onto a cell that contains a reward
Scenario:	1. Gameplayer: observes a reward on the map 2. Gameplayer: player moves to the cell containing the reward 3. Gameplayer: player observes their points increase by the amount of the reward 4. Gameplayer:
Exceptions:	N/A
Open issues:	If a player moves to a bonus reward at the same tick of the game that that bonus reward disappears on, should the player still collect that reward?

Use case 4:	Player goes through unlocked door
Primary actor:	Game player
Goal in context:	Player moves to a new cell moving through an unlocked door
Preconditions:	Some doors of the hexagonal rooms are set to be unlocked and allow the player to go through
Trigger:	Gameplayer walks towards a cell that is on the other side of an unlocked door
Scenario:	1. Gameplayer: tries to walk up, down, left or right towards a cell 2. Gameplayer: has an unlocked door between their current position and the cell they wish to move to 3. Gameplayer: Moves to the desired cell through the unlocked door
Exceptions:	N/A
Open issues:	N/A

Use case 5:	Player tries to go through locked door
Primary actor:	Game player
Goal in context:	Player remains idle when trying to move into a locked door
Preconditions:	Some doors of the hexagonal rooms are set to be locked and act as barriers
Trigger:	Gameplayer tries to walk towards a cell that is blocked by a locked door
Scenario:	1. Gameplayer: tries to walk up, down, left or right towards a cell 2. Gameplayer: has a locked door between their current position and the cell they wish to move to 3. Gameplayer: Remains idle in the initial position
Exceptions:	N/A
Open issues:	N/A

Use case 6:	Player Enters Bee Trap Room
Primary actor:	Game player
Goal in context:	To decrease points when player goes into bee trap

Preconditions:	The door leading to the trap room is not locked
Trigger:	The gameplayer walks into a bee trap room
Scenario:	1. Gameplayer: opens door and moves into red highlighted room 2. Gameplayer: loses points by the amount of the bee trap 3. Gameplayer: if points go below 0, game is over, player loses
Exceptions:	N/A
Open issues:	N/A

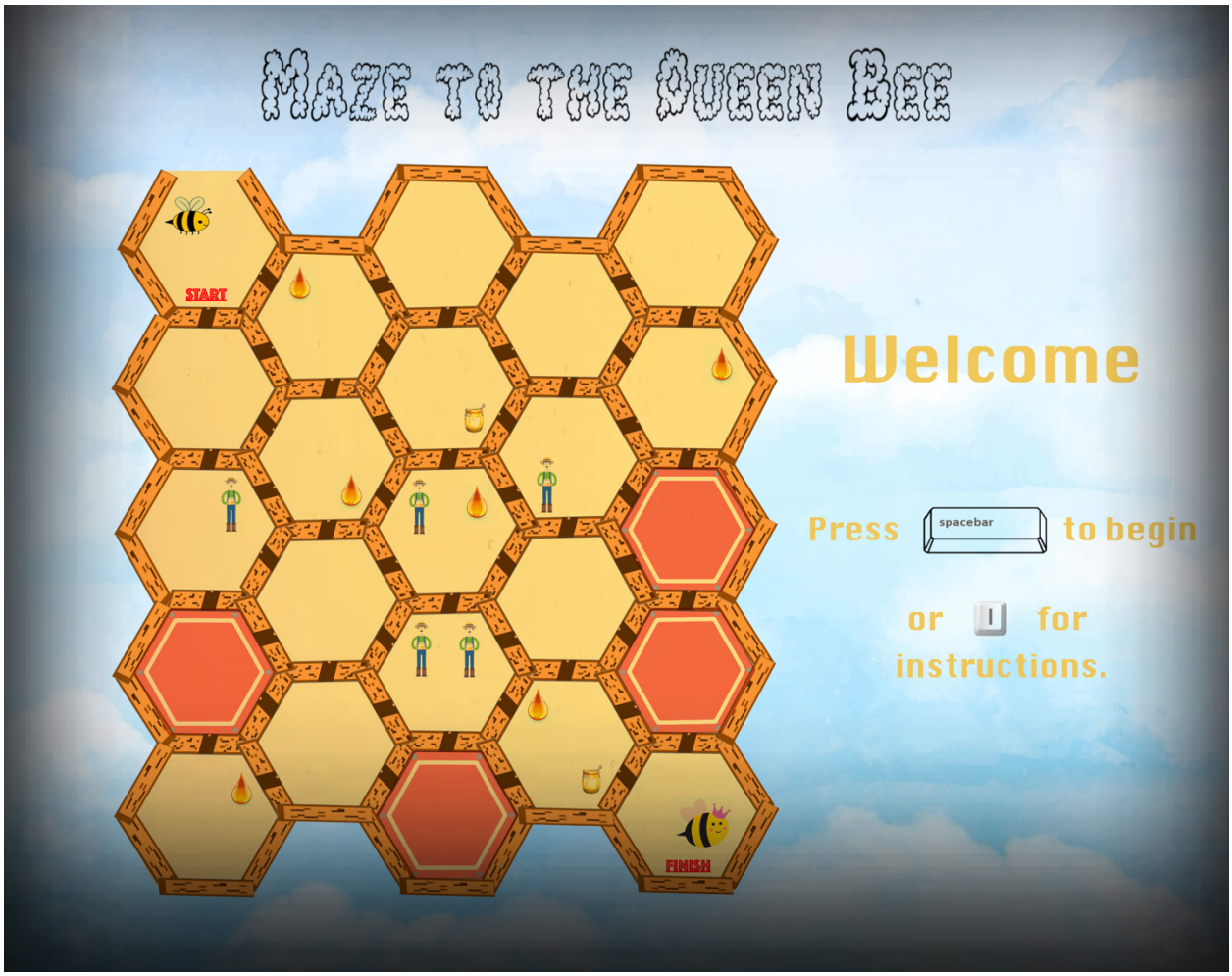
Use case 7:	Player encounters moving enemy (beekeeper)
Primary actor:	Gameplayer
Goal in context:	Player loses game after moving into an enemy or enemy moves into the player
Preconditions:	The enemies are programmed to move in the direction that gets them closest to the player
Trigger:	Either player or beekeeper make a move putting them onto the same cell
Scenario:	1. Gameplayer: moves into a cell that a beekeeper moves into at the same tick, or remains idle and a beekeeper moves into its current cell
Exceptions:	N/A
Open issues:	N/A

Use case 8:	Pause the game to do some other functions
Primary actor:	Gameplayer
Goal in context:	Stop and quit the game you are playing
Preconditions:	Player must be playing the game
Trigger:	The player decides to stop the game for a break or wants to quit the game
Scenario:	1. Gameplayer presses “ESC” on the keyboard or clicks “PAUSE” when playing the game. 2. The game stops and shows 3 options: “Resume”, “Rules”, “Quit” 3.“Rules” will show you the rules of the game. The player clicks the close button to go to step 2. “Resume” takes the player back to the game. “Quit” to quit the game.

User Interface Mock-ups

Before Starting a Round

This image is a mock-up of the user-interface in its initial state before the user starts a round of the game. It displays the game's name, the maze and all its components, and two simple instructions for the user.



During a Round

This image is a mock-up of the user-interface once the user starts a round of the game. The player's view of the maze is limited to everything within a certain radius of their bee character as well as a view of the destination room containing the queen bee. The players score, time, and number of 'honey drop' rewards collected are displayed beside the maze.

UML Class Diagram