Maze Game Documentation

Implementation

The features we implemented for the maze game are 5 items which need to be retrieved before the player can use the exit, a backpack which is able to store the items that the player retrieves, and controllers used to help move the player and game interactions.To implement the five items, require us a function called create_random_items which items in a list and uses a for loop to randomly assign each item to a random location. Next, In order to implement the backpack,simply have to assign it as a list in the player file, then when the player steps over the item, a function is called and adds the item to the backpack . This will happen to all five items that the player needs to find in order to win. Controllers are used to operate and run the game. The game controller is used to read input to move the player character by checking the user's input, then using a while loop to update the player's location relative to the amount of input made. The player will be able to, at any time, quit by pressing the 'q' button on the keyboard when the player is playing the game. The next features are the welcome controller which starts the game and end_game controllers which closes the game by raising a SystemExit.

Test Plan

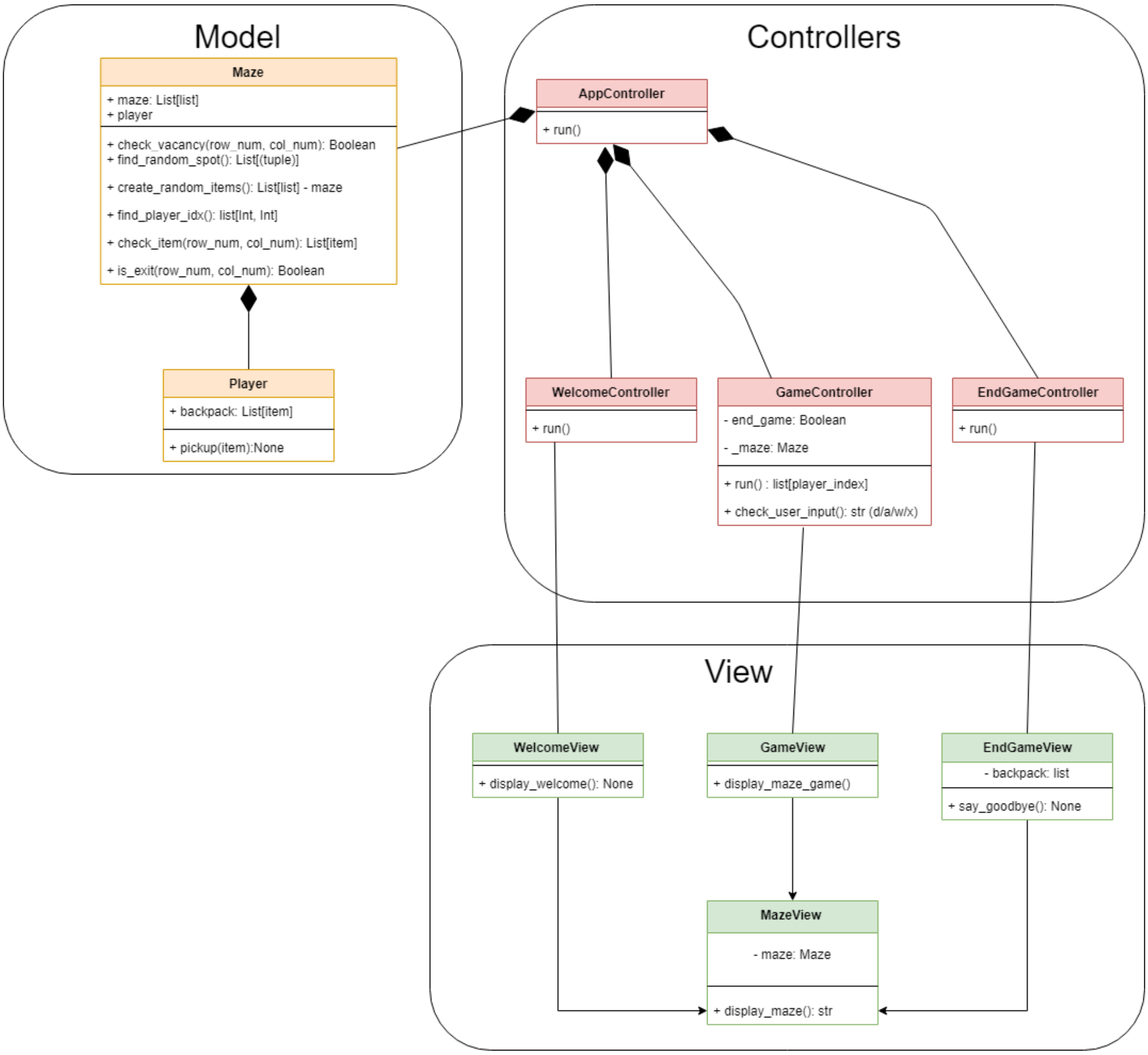
Class Maze

Method Tested	ID	Comment
check_item	001A	Checks if the player is on the item
find_random_spot	002A	Items currently appear in empty spaces
find_player_idx	003A	Returns the player's coordinates
pickup_item	004B	Adds and stores items

Class Player

Method Tested	ID	Comment
__init__	005A	Defines the player as a string and backpack a list

UML Class Diagram


Model
Maze
+ maze: List[list]
+ player
+ check_vacancy(row_num, col_num): Boolean
+ find_random_spot(): List[(tuple)]
+ create_random_items(): List[list] - maze
+ find_player_idx(): list[Int, Int]
+ check_item(row_num, col_num): List[item]
+ is_exit(row_num, col_num): Boolean
Player
+ backpack: List[item]
+ pickup(item):None
Controllers
AppController
+ run()
WelcomeController
+ run()
GameController
- end_game: Boolean
- _maze: Maze
+ run() : list[player_index]
+ check_user_input(): str (d/a/w/x)
EndGameController
+ run()
View
WelcomeView
+ display_welcome(): None
GameView
+ display_maze_game()
EndGameView
- backpack: list
+ say_goodbye(): None
MazeView
- maze: Maze
+ display_maze(): str