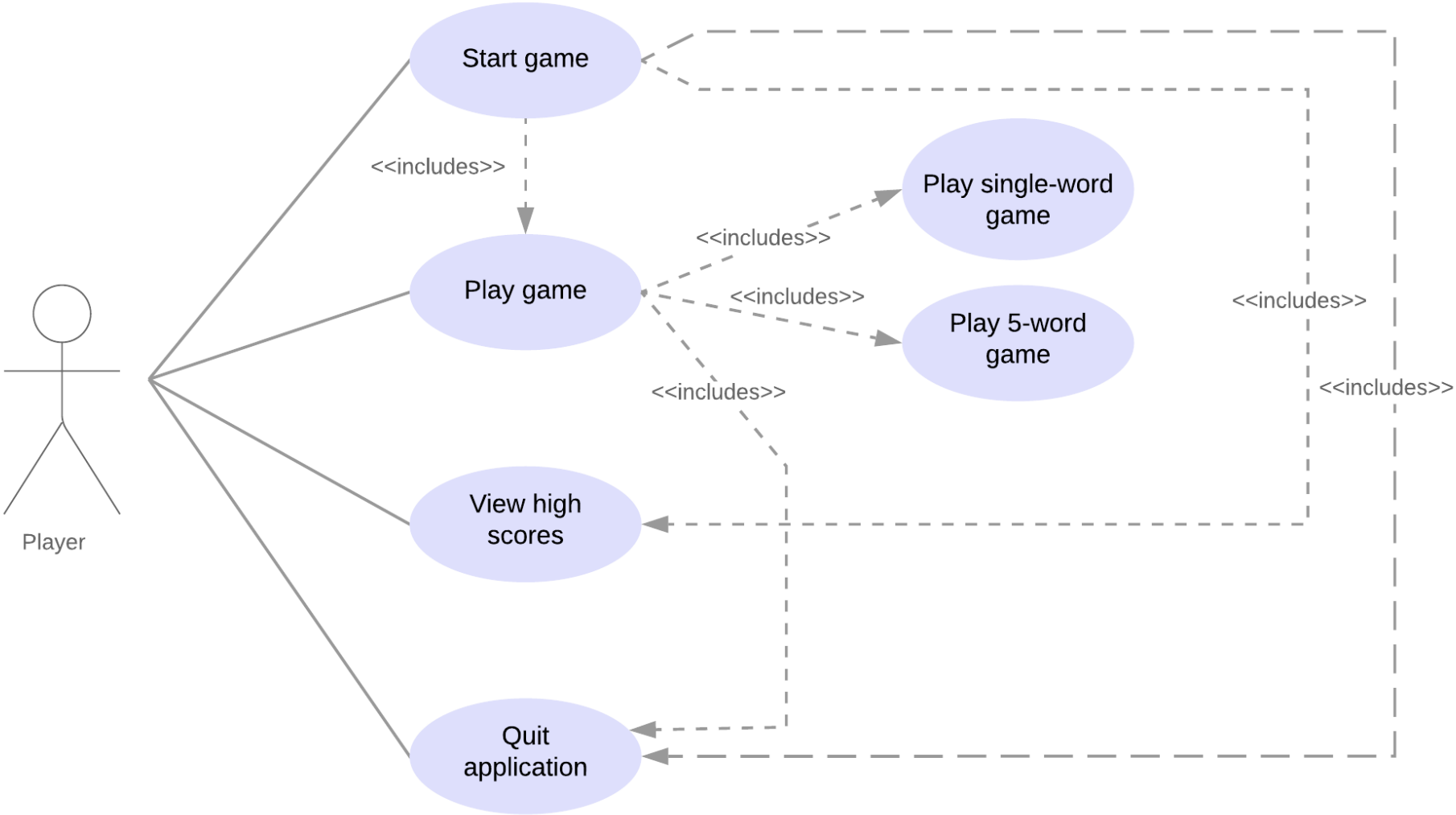
Time Log

Task	Anticipated time (hours:minutes)	Real time (hours:minutes)	Comments
Reading theme 2 book chapters and miscellaneous study material	12:00	20:30	• Severely underestimated the time it would take to write down notes for these many chapters • Underestimated how much I would have to complement book chapters with online tutorials and other reference material
Watching recorded Q&A lectures	02:00	03:45	
Sketching Use Case diagram	00:30	01:45	
Adapting Use Cases and writing fully dressed UC2	01:30	01:45	
Building "Play game" state chart	02:00	04:15	• Had not taken into account the amount of time required to search examples and understand how to use certain UML components
Implementing game	06:00	15:45	• Have both underestimated the implementation's difficulty and overcomplicated the game's implementation at this stage • Lost a lot of time implementing confirmation by the user • Did not anticipate multiple errors using Visual Studio Code for Java, nor the necessity to change to Eclipse in the middle of the implementation
Building class diagram	00:30	00:45	

Thoughts: I must take better into account the extra time it takes to take notes when studying the reference material and, especially, to complement the book and lectures with additional online resources. Also, regarding the +9h45 implementation time, do not implement more than is required for each iteration.

Use case diagram



UC 1 Start Game

Precondition: none.

Postcondition: the start menu is shown.

Main scenario

1. Starts when the Player wants to begin a session of the hangman game.
2. The system presents the main menu with a title, the option to play a new game, view the high score tables and quit the game.
3. The Player makes the choice to start the game.
4. The system starts a new game (see Use Case 2).

Repeat from step 2

Alternative scenarios

3.1 The Player makes the choice to view the high scores.

1. The system presents the high scores menu (see Use Case 3)

3.2 The Player makes the choice to quit the game.

1. The system quits the game (see Use Case 4)

4.1 Invalid menu choice

1. The system presents an error message.
2. Goto 2

UC 2 Play single-word game

Precondition: the start menu is shown.

Postcondition: a new game is running.

Main scenario

1. Starts when the Player wants to play a new game.
2. The system presents the new game menu, with a title, the option to play a single-word game, to play a 5-word game, or to go back to the main menu.
3. The Player makes the choice to start a single-word game.
4. The system starts a new single-word game, selects a random word to be guessed, and waits for player input.
5. The Player enters a letter which is contained in the word
6. The system checks the input, updates the displayed word revealing the guessed letter(s) and asks the Player for a new input

Repeat from step 5 until the player has no more tries left or the word has been guessed

Alternative scenarios

3.1 The Player makes the choice to start a 5-word game.

1. The system starts a new single-word game
2. Repeat from step 5
3. After each single game is finished, the system starts a new single-word game until a total of 5 single-word games has been played

3.2 The Player makes the choice to go back to the start menu.

1. The system returns to the start menu (see Use Case 1).

5.1 The Player enters a letter which is not a part of the word

1. The system decreases the number of tries by 1, updates the hangman drawing and asks for new user input
2. Goto 5

5.2 The Player enters an invalid character

1. The system exhibit a message and asks the Player for new input
2. Goto 5

5.3 The Player enters the word “quitgame”

1. The system asks the Player for confirmation
2. The Player confirms
3. The system terminates (see Use Case 4)

5.4 The Player enters the word “resetgame”

1. The system asks the Player for confirmation
2. The Player confirms
3. The system returns to the start menu (see Use Case 1)

5.5 The Player tries to guess the hidden word and fails

1. The system reduces the number of tries remaining, updates the hangman drawing and asks the user for a new input
2. Goto 5

5.6 The Player tries to guess the hidden word and succeeds

1. The system displays a message informing that the user has won and the options to go back to the start menu (see Use Case 1) or quit the application (see Use Case 4)

6.1 The Player has run out of tries

1. The system displays a message informing that the user has lost and the options to go back to the start menu (see Use Case 1) or quit the application (see Use Case 4)

UC 3 View High Scores

Precondition: the start menu is shown.

Postcondition: a high scores table is shown.

Main scenario

1. Starts when the Player wants to view the previous high scores.
2. The system presents the high scores menu, with a title, the option to view high scores for single-word games, view high scores for 5-word games, or to go back to the main menu.
3. The Player makes the choice to view high scores for single-word games.
4. The system presents a table containing the top 5 high scores for single-word games and the option to go back to the start menu.

Alternative scenarios

3.1 The Player makes the choice to view high scores for 5-word games.

1. The system presents a table containing the top 5 high scores for 5-word games and the option to go back to the start menu.

3.2 The Player makes the choice to go back to the start menu.

2. The system returns to the start menu (see Use Case 1).

3.3 The Player enters an invalid input.

1. The system presents an error message.
2. Goto 2

UC 4 Quit Game

Precondition: The game is running.

Postcondition: The game is terminated.

Main scenario

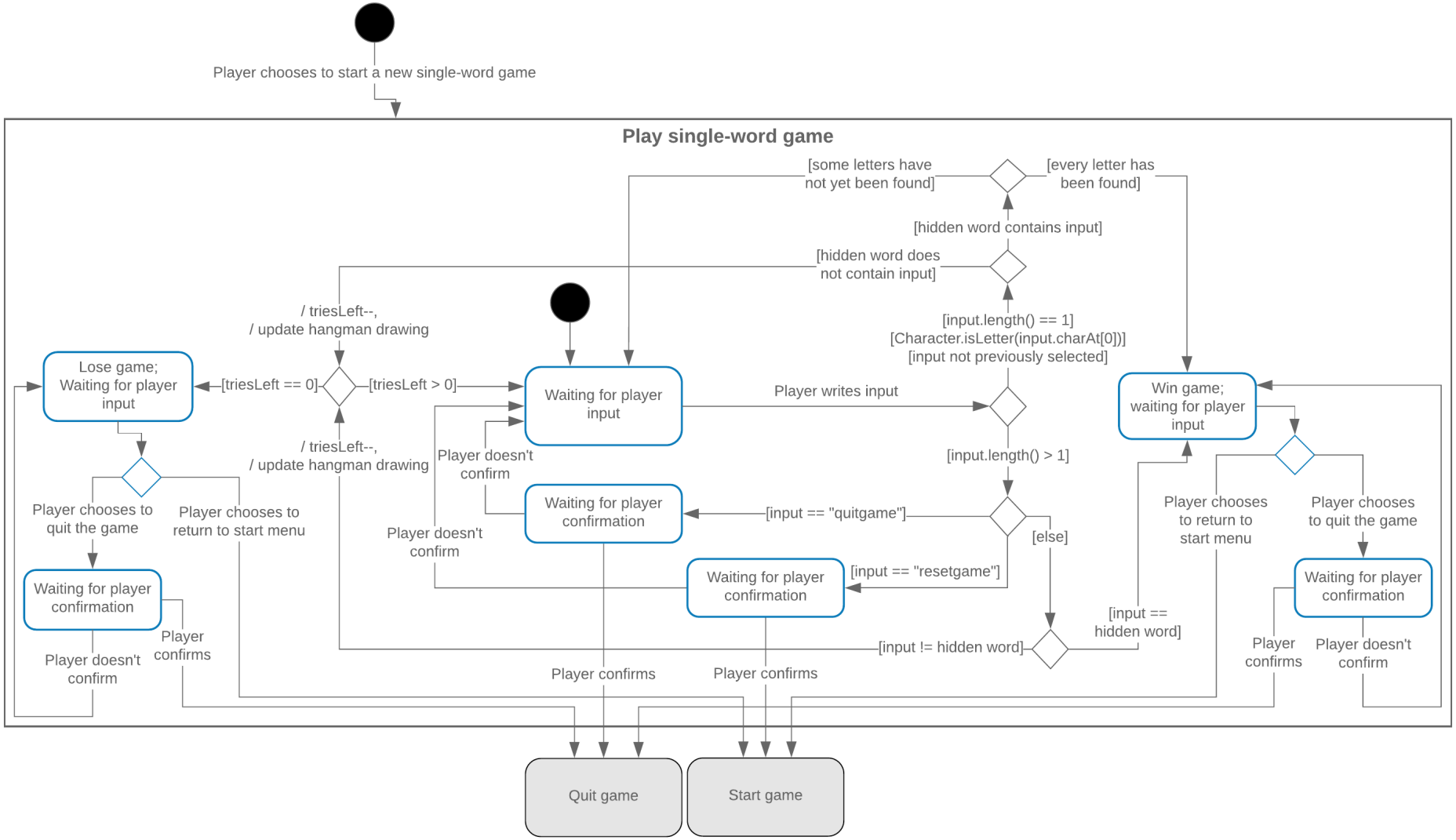
1. Starts when the Player wants to quit the game.
2. The system prompts for confirmation.
3. The Player confirms.
4. The system terminates.

Alternative scenarios

3.1. The Player does not confirm

1. The system returns to its previous state

”Play single-word game” state chart



Class diagram