

Innehållsförteckning

1. Planning iteration 2.....	2
2. Use Case Diagram	3
3. User Cases.....	4
UC 1 Start Game	4
UC 2 Play Single Player Game	4
UC 3 Play multiplayer game	5
UC 4 Play game	6
UC 5 Quit Game	7
4. Extended state machine	8
Extended hangman state chart	8
Play single player version state chart (User Case 2)	9
Play multiplayer version state chart (User Case 3)	10
Play game state machine (User Case 4)	11
5. Class Diagram of implementation	12
6. Time Log for iteration 2.....	13
7. Implementation comments	13

1. Planning iteration 2

Deadline: 22/2-2019

Read chapters: 4, 5, 20, 6, 7 and 15 in *Software Engineering 10th ed. by Ian Sommerville*. This is 200 pages and should take in total 7-8 hours. This should not be read in a streak but distributed through iteration 2. Lectures on this subject should also be attended. There are 7 lectures which is estimated 2hours each which result in 14 hours.

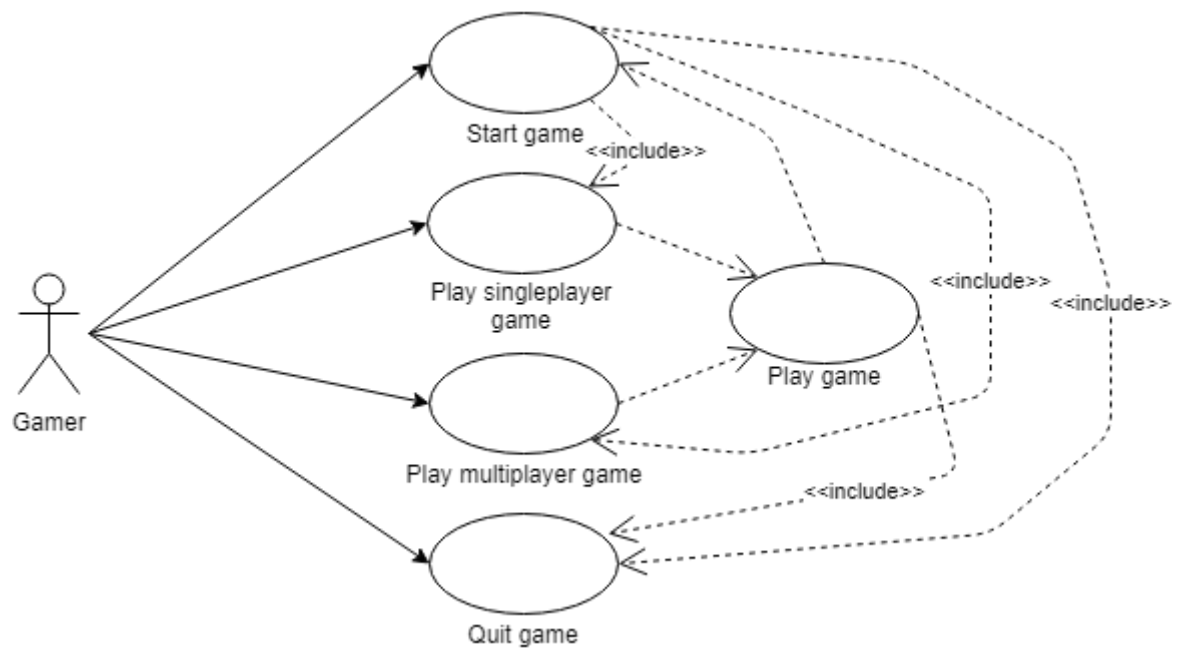
Basic:

- Write a fully dressed use case for a *basic* version of requirement UC 2 Play Game. This should take 4 hours.
- Create a state machine for playing the game. This should take 2 hours.
- Implement the models. This should take 3 hours.
- Create a class diagram. This should take an hour.

Additional:

- Create extended use case diagram, this should take 30min.
- Write fully dressed use case for single player game, this should take an hour.
- Write fully dressed use case for multiplayer version. This should take an hour.
- Create state machine for single player version.
- Create state machine for multiplayer version. This should take an hour.
- Implement as much as possible of the extended models. This should take 4 hours.
- Create a class diagram. This should take an hour.

2. Use Case Diagram



3. User Cases

UC 1 Start Game

Precondition: none.

Postcondition: the game 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.
3. The player makes the choice to play a single player game.
4. The system opens a single player game (see Use Case 2).

Repeat from step 2

Alternative scenarios

3.1 The player makes the choice to play a multiplayer game.

1. The system begin a multiplayer game (see Use Case 3)

3.2 The player makes the choice to quit the game.

1. The system quits the game (see Use Case 5)

3.3 Invalid menu choice

1. The system presents an error message.
2. Go to 2 in main scenario.

UC 2 Play Single Player Game

Precondition: System is running

Postcondition: A single player game has been played

Main scenario

1. Starts when player wants to play single player game.
2. The player chooses to play a new game.
3. System creates new game and play the game (See Use Case 4).
4. System ask if player want to play another game.
5. Player chooses to not play more games.
6. Return to menu (See Use Case 5)

Alternative scenarios

2.1 The player chooses to return to previous game.

1. Play previous game (See Use Case 4). If there is no previous game, play a new game.

5.1 Player chooses to play another game.

1. Go to 3 in main scenario.

UC 3 Play multiplayer game

Precondition: The system is running

Postcondition: A multiplayer game has been played.

Main scenario

1. Starts when the player wants to play a multiplayer game.
2. Rules of multiplayer version is shown and player 1 is asked to enter a word.
3. Player enters a word.
4. System set the word and ask for confirmation that the player want to use this word.
5. Player confirms
6. System display that player 2 should now guess the word, player is asked to confirm to continue.
7. Player confirms.
8. The game is played (See use case 4).
9. Player 2 managed to guess the word.
10. Program display that player 2 won the game and is asked to confirm to continue.
11. Player confirms
12. Return to menu (See Use Case 5).

Alternative scenarios

3.1. The player enters invalid word

1. Error message is shown
2. Player is asked to enter a new word.
3. Player enters a new word.

5.1 Player does not confirm

1. Player is asked to enter a new word
2. Player enters a new word.
3. Go to 4 in main scenario.

7.1 Player does not confirm

1. System continues to wait, player must confirm.

9.1 Player 2 did not manage to guess the word

1. System display that player 1 won the game and is asked to confirm to continue.
2. Go to 11 in main scenario.

UC 4 Play game

Precondition: The system is running.

Postcondition: A hangman game has been played.

Main scenario

1. Starts when the player wants to play a game.
2. System show choices and clues and tell player to enter a letter or a choice.
3. Player enters a letter.
4. System tells where in the word that letter is placed.
5. System presents that game is won and ask player for confirmation to continue.
6. Player confirms.
7. Return to previous state.

Alternative scenarios

3.1 The player makes invalid input.

1. Error message is shown.
2. Player is asked to make a new input
3. Player makes new input

3.2 The player makes the choice to quit the game.

1. The system quits the game (see Use Case 6)

3.2 The player makes the choice to return to menu.

1. Player is asked to confirm.
2. Player confirms.
 - 2.1 If player do not confirm: Go to 2 in main scenario.
3. The game returns to the menu. (See UC 1)

4.1 Word does not contain letter.

1. Add part of hangman.
2. Letter is added to guessed letters.
3. Go to 2 in main scenario.

5.1 All the letters in the word is not guessed yet

1. Go to 2 in main scenario.

5.2 Max number of wrong guesses is reached.

1. Game presents game over and ask player to confirm to continue.
2. Player confirms
3. Return to previous state.

UC 5 Quit Game

Precondition: The system is running.

Postcondition: The system 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.

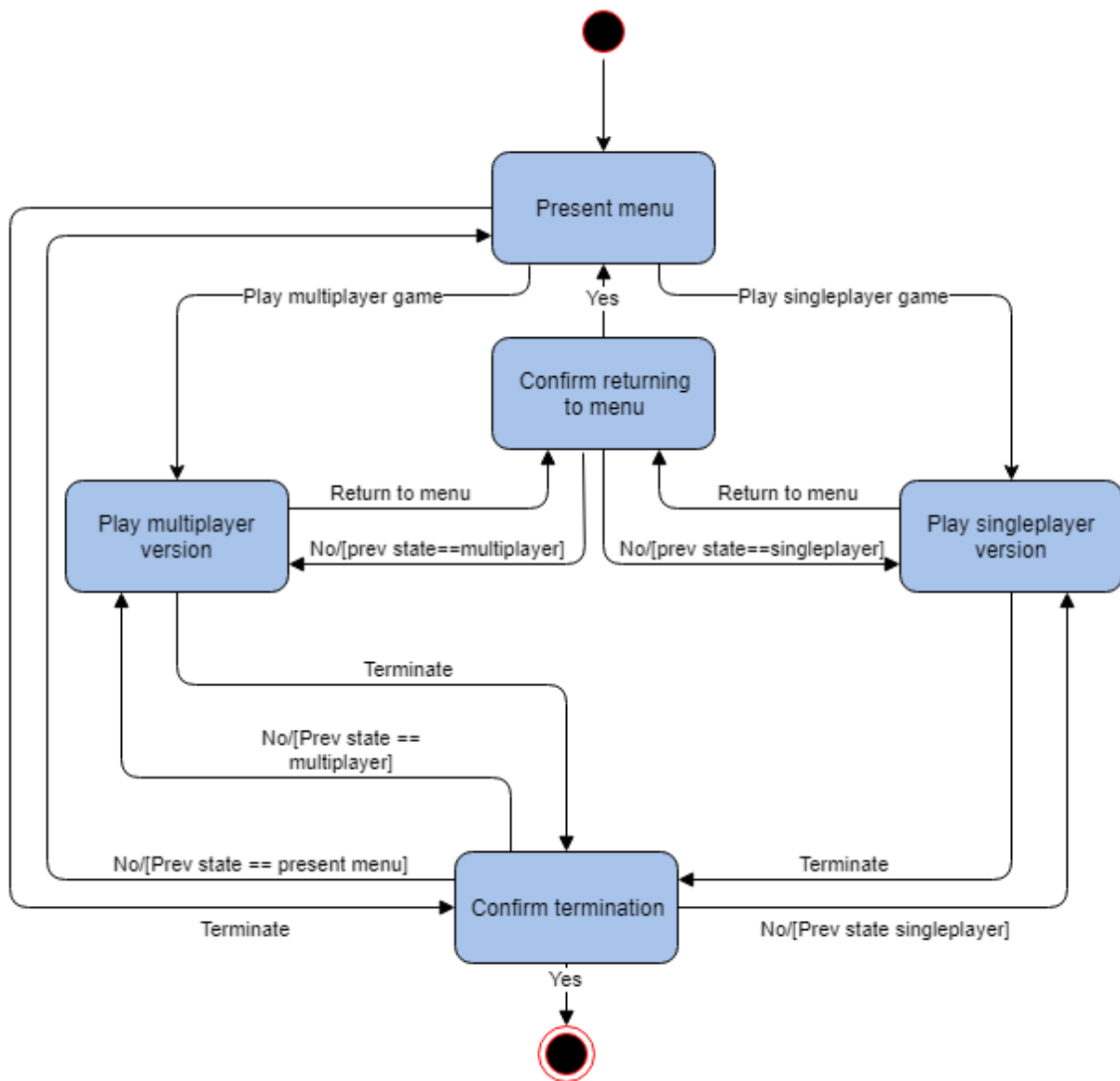
Alternative scenarios

3.1. The player does not confirm

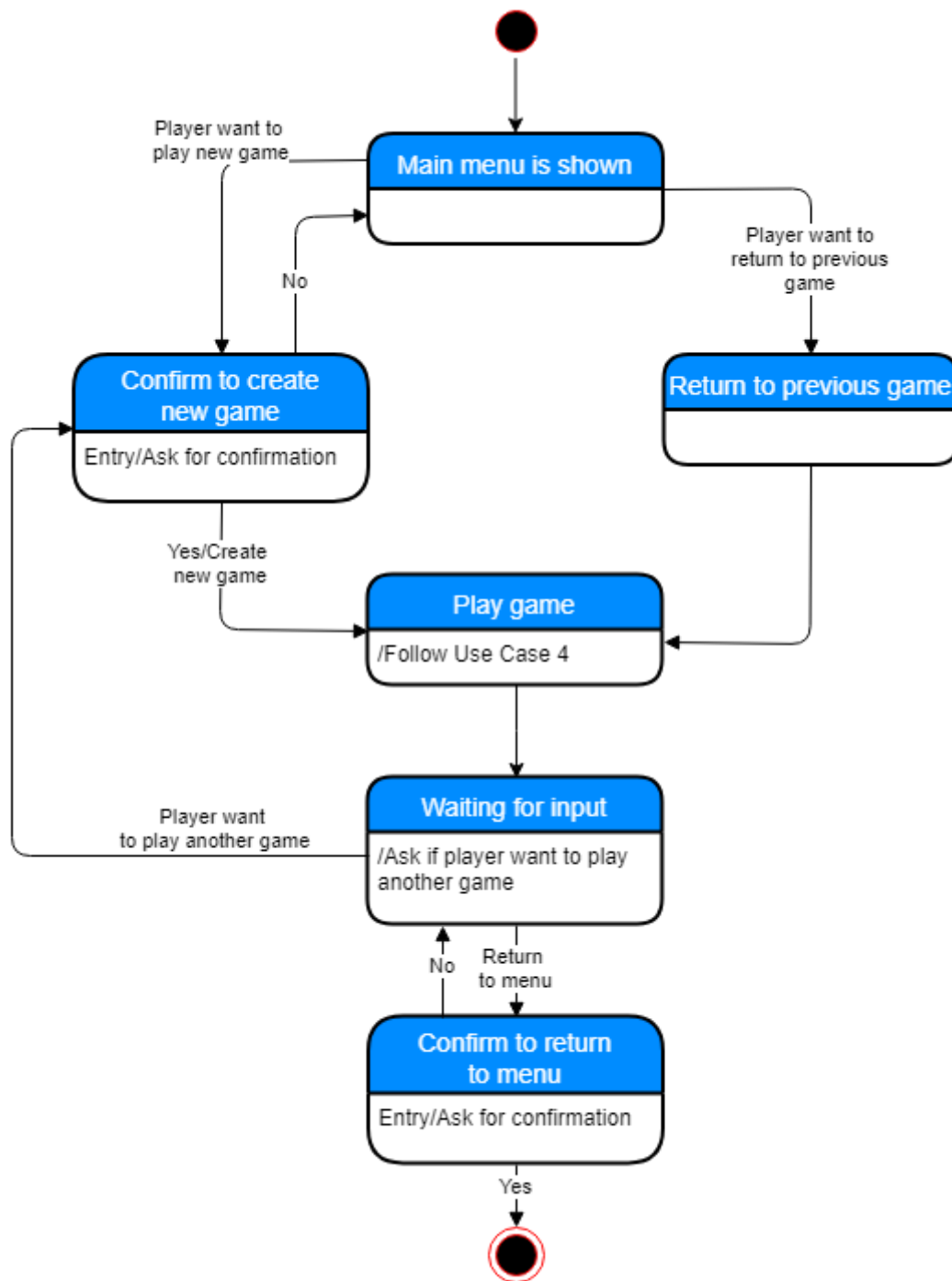
1. The system returns to its previous state

4. Extended state machine

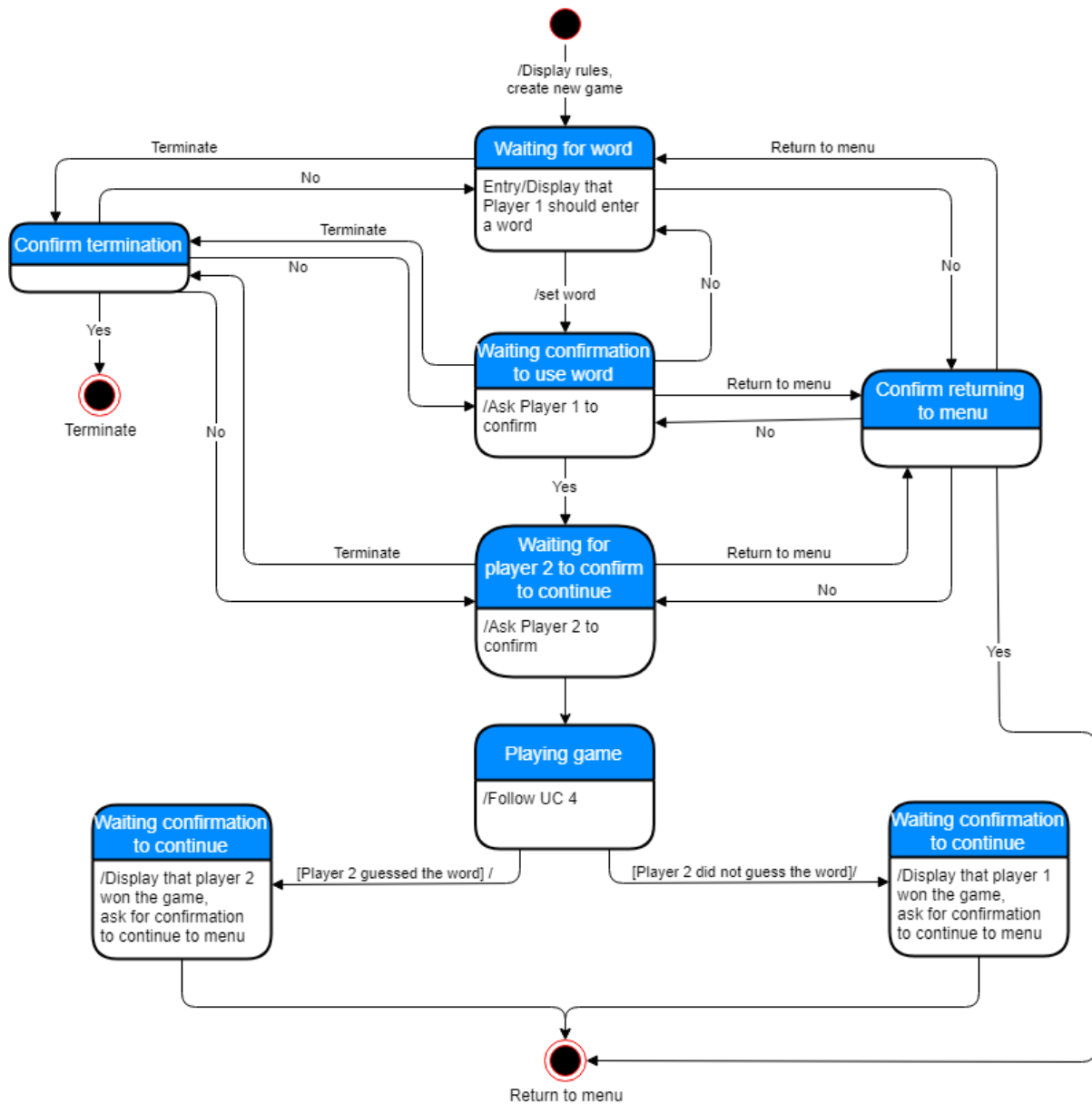
Extended hangman state chart



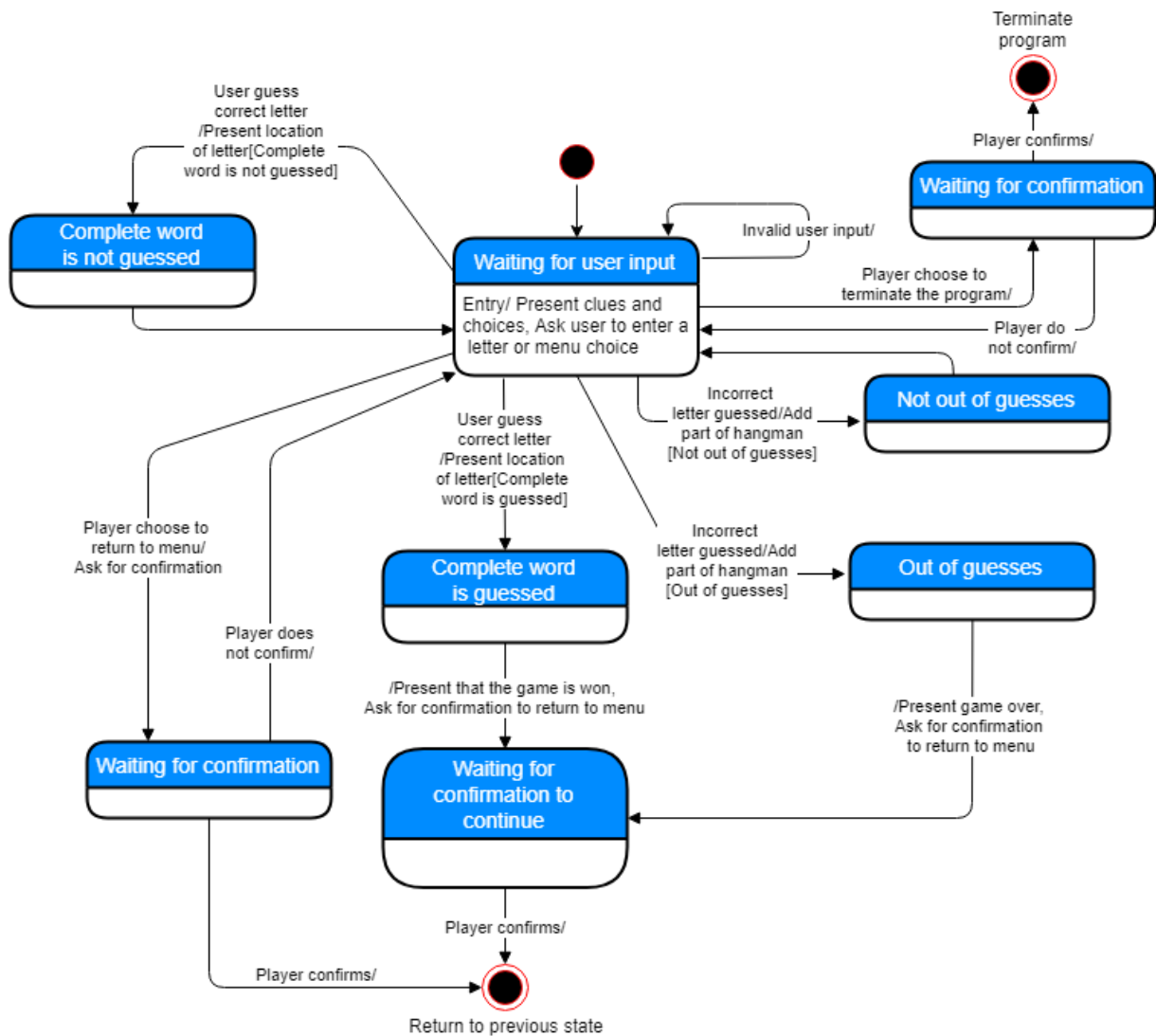
Play single player version state chart (User Case 2)



Play multiplayer version state chart (User Case 3)

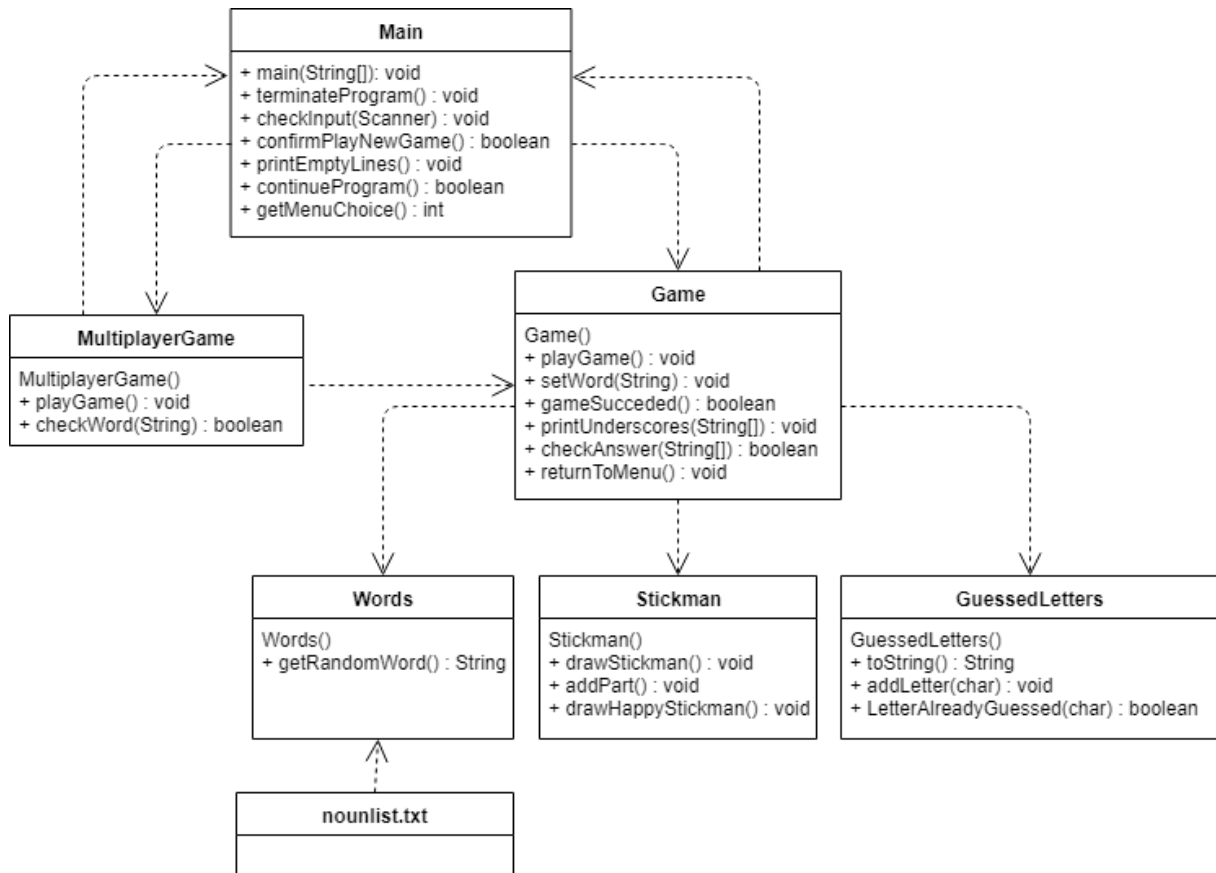


Play game state machine (User Case 4)



5. Class Diagram of implementation

Class diagram of the implementation so far.



Both private and public methods are included in this diagram.

6. Time Log for iteration 2

Iteration 2

Task	Date	Estimated time	Actual time	Reflection
Reading literature	5-18/2	7-8 hours	8 hours	
Attend lectures	6-20/2	14 hours	14 hours	
Write a fully dressed use case for a <i>basic</i> version of requirement UC 2 Play Game	13-20/2	4 hours	4 hours	Had to go back after starting the extended version so this actual time may not be exact.
Create a state chart for playing the game.	13-20/2	2 hours	6 hours	Took a lot of time to understand. Find and learn a program to draw state chart. Some time was wasted doing unnecessary work.
Implement the models.	16-19/2	3 hours	4 hours	Time went faster than expected
Create extended use case diagram	13-18/2	30min	45 min	Fairly ok estimation. Changed this diagram several times.
Write fully dressed use case for single player	18-20/2	1 hour	40min	
Write fully dressed use case for multiplayer version	18-20/2	1 hour	1 hour	
Create state machine for single player version	18-19/2	1hour	50min	
Create state machine for multiplayer version	18-19/2	1hour	1hour	
Implement the extended models	18-22/2	4 hours	5-6 hours	Not all extended model functions are not implemented. Some functions was harder to implement than expected.
Updating models and user cases	16-22/2		3 hours	A lot of time was consumed changing things since I changed my mind on how to do them.
Create a class diagram.	19-20/2	1 hour	1 hour	

It was very hard to measure time even though toggle was used. Especially when I was stressed and went back and forth between different tasks. The actual time may have been slightly underestimated but is as accurate as possible.

7. Implementation comments

The game has been implemented using java. I have used the IDE Eclipse and have only played the game here. Not sure if these commands work in the terminal.