Computer Science 2005

Group 1 – Fully Dressed Use Case Description

Set Up a Game

Primary Actor: User

Stake Holder and Interest:

- User: would like to load the game, ask for help or a start game by selecting the computer's difficulty level and board design.
- Computer AI: To be able to know its difficulty level

Pre-Conditions:

- There is at least one human player.

Success guarantee (Post Conditions):

- The user is able to start the game.

Main success scenario:

1. The user chooses the board design, the number of computer bots playing and their difficulty level. [Alt 2: user loads the game]
2. The user would then select to start the game. [Alt 1: use case ends]
3. The system then retrieves all the match details and settings of the game and designs the board accordingly.
4. The system will then get all the robot pieces (4 different colours) and set them on board.
5. The system would then set up a scoreboard, match timer and hints needed to assist the users.
6. The system then confirms to the user that the game has set up.

Alternative Flows:

Alt 1: Use case ends

1. Use case ends.

Alt 2: user loads the game

1. The system will then find the save game and load it.
2. Flow resumes at flow 3.

Exceptions:

- If at any time the system is unable to retrieve the save game, the system would then inform the user of the problem and the nature of failure.

Special Requirements:

- Confirmation of the game has loaded within 15 seconds.

Open Issues:

- How will we provide features for user with colour vision deficiency?
- Do we have a save game before we load anything?

Take a turn

Primary Actor: User

Stake Holder and Interest:

- User: wants to make bid for his/her route selection, set the position of different robots on the board.
- Computer AI: wants to make bid for its route selection and set the position of different robots on the board.

Pre-Conditions:

- The game is set up properly.

Success guarantee (Post Conditions):

The user is able to make bid and move legally on the board.

Main success scenario:

1. The system would randomly select and display a target chip to all players.
2. The system would then provide an opportunity for all players to make their bid for route selection
3. The players would then go on and make a bid (Users will make bid in the bid box).
4. The system would then receive details of the bid made by the player and would then display the timer for all players to place a bid higher, lower or equal to the bid placed by first user.
5. The system then checks if all bid made were legal.
6. The system then goes on to get the next lowest bid and provide an opportunity for that player with lowest bid to make the move.
7. User would then make their move.
8. When the move is made, the system will retrieve the details of moves made by the user/computer and checks if the moves made were legal and done under required number of moves.

9. If the moves were legal the system would then add a score and target chip to that user details. [Alt 1: Player didn't meet the requirement]
10. The system would then check if all target chips are used [Alt 2: All target chips not used].
11. System would then display the scores and declare the winner.

Alternative Flows:

Alt 1: Player didn't meet the requirement by either making illegal move or more moves than required

1. Flow resumes at step 6.

Alt 2: All target chips not used

1. Flow resumes at step 1.

Exceptions:

- if at any time the system is unable to make/process player selection the use case would then end.

Special Requirements:

- Colours of game display and size of text fonts used must cater the need of users with colour vision deficiency.
- Confirmation of saving game (or reason to failure) would be provided to user within 5 seconds of clicking the button.

Open Issues:

- How will we implement hints that will actually help the user to make their move?