Crosswinds

Software Requirements Specification

Keith Co
Roger Han
Dan Harris
Charlene Jeune
Tim Jones
Brandon Kelch
Gary Mgerian
Noel Overkamp

Alex Francois (Project Manager)

Table of Contents

1. Introduction

The document is intended to establish the initial scope of the development effort.

Crosswinds is a multi-player turn-based strategy game.

2. External Interface Requirements

This section describes user and system requirements that affect the operation of the game within a given system or within a network context.

2.1 User interface

The user interface interacts with the game using the keyboard, and a USB gamepad (optional). (Mouse interaction might be considered in the future.)
The game's interface consists of 3D graphics and sound/music events.

Keyboard Input/Game Controller (Gary Mgerian)
The user will interact with the game through the keyboard. All of the desired moves and selections are made by inputting designated keys which perform commands such as commit move, select piece, quit game,etc.

COMMAND	***KEY(S)***
Commit	c, "Enter"
Unselect	u
Select Next Piece (iterate through the pieces)	'f' or '>', 'n'
Select Previous Piece (iterate though the pieces)	'd' or '<', 'm'
Translate	arrow keys (up,down,left,right)
Rotate clockwise	r
Rotate counterclockwise	e
Remove a piece	R
Quit	q

The Commands for the Game Controller are as follows:

COMMAND	***KEY(S)***
Commit	'O'
Unselect	'^' (triangle button)
Select Next Piece (iterate through the pieces)	'R2'
Select Previous Piece (iterate though the pieces)	'R1'
Translate	arrow keys (up,down,left,right)
Rotate clockwise	'X'
Remove a piece	|_| ("square button")

2.2 Hardware Interfaces

Regular PC with reasonable (contemporary) graphics and sound capabilities.

2.3 Software Interfaces

The system relies on various libraries such as the OpenGL library for graphics.

The system will be implemented using the MFSM architectural middleware; and existing relevant functional modules whenever available.

2.4 Communications interfaces

The system uses Windows Sockets to support communications between individual game components. The system implements an ad-hoc ring topology and matching protocol.

3. Functional Requirements

When you begin to word sentences for your requirements document, word them in the active voice, use the verb "shall," and try to make them pertain to only one feature at a time. Try to use only two subjects: player (or user) and software (or system).

3.1 Primary List

1. Software shall support multiplayer, and accommodate 2-4 players.

2. Software shall support text communications between the players.

3. Software shall allow player to select a piece type to place on the board.

4. Software shall allow player to select a piece already on the board.

5. Software shall show player which player is active.

6. Software shall display the scores of all the players.

7. Software shall allow successive rounds without changing the state of the game pieces.

8. *Current speed of token is displayed through highlighted path, this requirement is thus no longer relevant*

9. Software shall show player which pieces are available to place.

10. Software shall display current direction. (Is this similar #8 in that only the path will indicate the direction as well?)

11. Software shall display all pieces currently on the gameboard.

12. Software shall highlight current path of token (which is constantly updated).

13. Software shall support background music.

14. Software shall support sound effects.

15. Software shall show the timer counting down on a turn (If this is implemented).

16. Software shall allow the removal of a piece from the gameboard.

17. Software shall support 3D graphics.

18. Software shall display the entire gameboard.

19. Software shall have a limited start menu with options to create a game or join a game or quit.

20. Software shall support an option to quit the game at anytime.

21. Software shall support keyboard and/or gamepad input.

22. Software shall prevent unlawful moves (and inform player of their mistake).

23. Software shall accommodate teams of 2 per game. (If this is implemented)

24. Software shall support simple AI computer player. (time permitting)

25. *combined with 12*

26. Software shall allow player to end their turn (commit move or pass).

27. Software shall provide a way to access the rules of the game.

3.2 Components view

Networking

Requirements: 1, 2, 19

Game AI (game state)

Requirements: 5, 6, 7, 8, 9, 10, 15, 19, 23

Interaction (interaction state)

Requirements: 3, 4, 8, 12, 16, 20, 21, 22, 25, 26, 27

Graphics rendering

Requirements: 5, 6, 8, 9, 10, 12, 15, 17, 18, 25

Chat

Requirements: 2

Sound/music

Requirements: 13, 14

AI player

Requirements: 24

4. Performance Requirements

5. Design Constraints

6. Other Requirements

Appendix A: Use Cases

1. Context use case (high-level actions available to the player(s)) (Charlene Jeune)

Requirement(s) explored: 1 - 27

Player (actor) context (role): Player

Precondition(s): Game has been installed and started

Trigger(s): Player starts up the game. (Case #3)

Main course of action:
1) The player will choose from a sequence of options.
2) The player will connect to a session. (Case #2)
3) Main gameplay begins.
3) The game is turn-based, and therefore the player will be able to perform actions only when it is his turn. (Case #5)
4) The player can engage in chat with other players during the game if he wishes. (Case #9)
5) The game ends after a player has won three rounds of play. (Case #4)
6) After the player has won or lost a game, the player can decide to play again with the same group, to play again with a new group, or to disconnect.

Alternate course(s) of action:
1) The player decides not to play the game and closes it immediately.

Exceptional course(s) of action:
1) Game does not run properly.
2) Game crashes the system.

2. Player connects to session (Dan Harris)

Requirement(s) explored:

<u>Player (actor) context (role)</u>: Player

<u>Precondition(s)</u>: The player wants to initiate a session or join an existing session.

<u>Trigger(s)</u>: The player opens the game and chooses to start a new session, or knows an IP address of a computer in an existing session.

<u>Main course of action</u>:
1) The player will be presented with two choices upon startup - start a new session or join an existing session.
2) The player commits to one of those choices.

<u>Alternate course(s) of action</u>:
The player cancels his choice and returns to the main menu.

<u>Exceptional course(s) of action</u>:
The player enters an IP address that does not correspond to an existing session.

3. Player plays game (Brandon Kelch)

<u>Requirement(s) explored</u>: 3 - 12, 14 - 16, 18, 21, 22, 24, 26

<u>Player (actor) context (role)</u>: User

<u>Precondition(s)</u>:
Program is up and running.

<u>Trigger(s)</u>:
User clicks icon/types command to start a game.

<u>Main course of action</u>:
Game starts up. Rounds are played and points are scored. One of the players reaches 3 points and the match is finished.
Game ends and user has the option to start a new session.

<u>Alternate course(s) of action</u>:
User decides not to play another session and the player is returned to the start menu.

<u>Exceptional course(s) of action</u>:
User quits the program before the game session is finished. There should be some kind of warning to let them know they are quitting the current session and will not be able to rejoin. Players pieces will remain on the board and a simple AI computer player will take over till the game is finished.

4. Player plays round (Noel Overkamp)

<u>Requirement(s) explored</u>: 2, 3, 4, 7, 16, 20, 21, 26

<u>Player (actor) context (role)</u>: Player

<u>Precondition(s)</u>:
Players have joined a session and started a game.

<u>Trigger(s)</u>:
Previous player commits move to end his/her turn.

A previous round has been completed.

The player is informed they are the first to play in a round.

Main course of action:
Player places/moves/removes piece during his/her turn. (cases #6, 8, 10)
Player chats with other players. (case #9)

Player plays turn. (case #5)

Player evaluates game state. (case #7)

Alternate course(s) of action:

Exceptional course(s) of action:
Player quits the game.

5. Player plays turn (Gary Mgerian)

Requirement(s) explored:16, 20, 21, 23, 26, 27

Player (actor) context (role): Player

Precondition(s): A player has just finished their turn, and the system has processed and completed their move and it is now the next players turn.

Trigger(s):The system notifies the player that it is now their turn.

Main course of action: The player begins their turn by selecting a game piece and placing it onto the board. (See "Player selects and positions piece (selects and moves piece)" for details). The system will display the token as the user begins to move it across the board. It will also display where the token will end up (if the player decides to "commit" the move) based on the location of the game pieces which are on the board. (See "System Displays Path of Token"). If the move they are trying to make breaks any rules, the system will notify the player that their move is illegal and it will not let them "commit" (See "Player Breaks a Rule"). Once this is done, the system will adjust the token to the appropriate location on the board and then notify the next person that it is their turn.

Alternate course(s) of action: In the case that it is one of the players turn to play, and they decide that they do not want to move the token in any direction, or they are simply away from the computer their turn will simply be skipped. Because, the system enforces a time limit for each turn, once the time limit has been reached, the system will automatically move on to the next player.

Exceptional course(s) of action: The player whose turn it is decides to quit the game. In this case the system will take over by skipping their turn each time.

6. Player selects a piece (Charlene Jeune)

Requirement(s) explored: 3, 4, 9, 11, 14, 16, 18, 20, 21, 25, 26

Player (actor) context (role): Player

Precondition(s): It is the player's turn

Trigger(s): The player desires to position (Case #8) or remove (Case #10) a piece.

Main course of action:

1) The player evaluates the game state. (Case #7)
2) The player moves cursor onto or in some other way highlights a piece.
3) A button is pushed to select it.
4) A sound and a change in the look of the piece indicates that it has been selected.
5) To deselect piece before it is committed to the board a different button can be pushed.
6) The player positions piece (Case #8) or removes it (Case #10).

Alternate course(s) of action:
1) The players decides to pass their turn.

Exceptional course(s) of action:
1) The player quits the game.

7. Player evaluates game state (Keith Co)

Requirement(s) explored: 5, 6, 8, 9, 10, 11, 15, 18, 22, 25

Player (actor) context (role): Player

Precondition(s): Game has been started and screen has been rendered. Information has been displayed to screen.

Trigger(s): When game starts, player can begin to evaluate game state.
Player usually evaluates game state after someone's turn.
Player can evaluate game state at any time.

Main course of action: A player will probably care about these changes in game state first and foremost (1st Priority):
1) The token's location.
2) The addition/removal/movement of any game pieces on the board.
3) The direction of the token.
4) The speed of the token.

A player will look at these things after the more immediate concerns above have been evaluated. These are 2nd Priority:
5) Who's turn it is.
6) How much time is left. (Assuming we implement a time limit on a turn).
7) How many pieces the player has left.
8) The score.

Alternate course(s) of action: Some players may look at the game state in a different order. Here is one example:
1) Who's turn it is.
2) How much time is left. (Assuming we implement a time limit on a turn).
3) The token's location.
4) The direction of the token.
5) The speed of the token.
6) The addition/removal/movement of any game pieces on the board.
7) How many pieces the player has left.
8) The score.

Exceptional course(s) of action: If the game freezes or crashes, players may lose focus and their analysis of the game state becomes unpredictable as they try to fix/exit the game.

8. Player positions game piece (Tim Jones)

Requirement(s) explored: 3, 4, 5, 9, 11, 15, 16, 18, 22, 26

Player (actor) context (role): Player

Precondition(s): The player has evaluated the current game state

Trigger(s): The player wishes to alter the game state by:
1) Placing a new game piece on the board that belongs to the active player.
2) Remove a game piece from the board that belongs to any player.

Main course of action:
1) The player wishes to place a new game piece on the board that belongs to the active player.
a) The player can choose from any of the pieces that they haven't placed on the board that belong to the active player
b) The piece that the player chooses cannot intersect other pieces that are already on the board
c) The piece that the player chooses cannot overlap the area of the token on the board
d) The piece that the player chooses cannot intersect with any players goal area or the center four squares
e) The piece that the player chooses cannot have it's arrow adjacent to either the center four squares on the board or any players goal
f) Once these above rules have been met then the current player may commit that piece to the board.

2) The player wishes to remove a game piece that is currently on the board
a) The player can choose from any of the pieces that are currently commited to the game board
b) Once the piece that is currently on the board has been chosen the active player can choose to remove that piece
c) The removed piece is taken off of the board and given back to the original players possession

Alternate course(s) of action:
The player may choose to not change the current game state and allow the token to move without placing, altering or removing any game pieces from the board.

Exceptional course(s) of action:
The player runs out of time before they are able to make a decision, thus forfeiting their turn, causing the token to move, and allowing the next player to begin their turn.

9. Player sends a text message (Roger)

Requirement(s) explored: 2

Player (actor) context (role): Player

Precondition(s): Other players are connected to the game. The game may or may not have started but other players should be present on the server/network.

Trigger(s): When the player wishes to transmit a message, he/she can hit a key on the keyboard to indicate the beginning of a text string.

Main course of action: The main course of action is to transmit a string of text to the other players currently in the game. A string of text is entered using the keyboard and then transmitted by hitting the 'enter' key.

Alternate course(s) of action: An alternate course of action is to transmit a string of text privately to another player of the game. Other players will not be able to view this text. This string of private text will only appear on the sender and recipient's chat area.

Exceptional course(s) of action: The player cancels the message that may have been typed already. This deletes all the text that may or may not have been previously written. If the player wishes to send another text message, he/she must start over.

10. Player removes a piece (Charlene Jeune)

Requirement(s) explored: 3, 4, 9, 11, 12, 14, 16, 18, 21, 25, 26

Player (actor) context (role): Player

Precondition(s): It is the player's turn. It is more likely to happen if the player has no pieces left in his "bank".

Trigger(s): Player decides it is is the best course of action after evaluating game state (Case #7).

Main course of action:
1) The player selects a piece from the board. (Case #6)
2) The player uses directional buttons or joystick to move piece off of board.
3) The piece is returned back to its owner's piece bank.
4) A sound is played that signals the action has been completed correctly.
5) The player's turn is over and the token moves at the current speed according to its current highlighted path (Case #11).

Alternate course(s) of action:
1) The player decides to add a piece to the board (Case # 8) or move one of their pieces on the board (Case #8).
2) The player decides to pass their turn.

Exceptional course(s) of action:
1) The player quits the game.

11. System shows path of the token (Noel)

Requirement(s) explored: 12, 25

Player (actor) context (role): System

Precondition(s): A game has been started.

Trigger(s):

The player's turn begins.

The player manipulates a piece.

Main course of action:

Determine the path of token, taking into account pieces on the board and the number of spaces it will move. Everytime a piece is placed on the board or moved or removed the path should be updated to reflect this move. For inactive players (those waiting their turn) the path is only updated at the beginning/end of turns.

Alternate course(s) of action:

The highlight path feature could be made an option decided during the creation of a game.

Exceptional course(s) of action:

The program crashes.

12. Player Breaks a Rule (Gary Mgerian)

Requirement(s) explored:22

Player (actor) context (role):Player

Precondition(s): Player must make a move which breaks an established game rule.

Trigger(s): The player attempts to position a game piece in a manner that is considered "invalid" (example: having the arrow on a game piece be less than one block away from any goal).

Main course of action: When the player does something that breaks one or more rules of the game, the system will notify the player that their move will be considered invalid, meaning they can not commit to it. The system will notify the user by either displaying an image, text, not displaying the path of the token, or outputting some sound which will allow notify the player that this move will break a game rule.

Alternate course(s) of action: It might be possible for system AI to suggest an alternative move.

Exceptional course(s) of action: The user decides that they are not going to try to make a move which will be allowed. In that case, the system will simply skip over their turn.

13. System Displays Path of Token (Gary Mgerian) (How is this different from Use Case 11?)

Requirement(s) explored:

Player (actor) context (role):

Precondition(s): A player is playing their turn.

Trigger(s): The game has begun. Once the game starts, the system will, at all times, show the path of the token if the player whose ever turn it is was to press the commit button.

Main course of action: As the game begins, and the system established who will go first as well as the initial starting position of the token and its direction, it will display the path of the token if the player was to commit to that move. The system will recognize the game pieces which are on the board, and it will show the appropriate path of the token if it where to go over any of the game pieces.

Alternate course(s) of action:

Exceptional course(s) of action: A game piece is positioned in such a way that it breaks a rule (see "Player Breaks a Rule"). When this happens, the system will not show the path of the token.

14. System Evaluates Game State(Gary Mgerian)

Requirement(s) explored:

Player (actor) context (role):

Precondition(s): There has been some sort of a change made in the game (i.e. Player makes the token into one of the goals).

Trigger(s): The system evaluates the game state each time a change has been made to the game. So any changes made to the game state would trigger the system to evaluate the game state.

Main course of action: The system evaluates the state of the game and either produces some sort of output (such as when a round is over and someone has won or a player is attempting to break a rule) or it simply updates the game as necessary.

Alternate course(s) of action:

Exceptional course(s) of action: There is somesort of a bug in the system which causes the game to frieze. In this case, the system might not be able to function properly.

Appendix B: Requirements Matrix

Requirement (#)	Status (% complete)	Title	Use case ref. (#)	Test case ref. (#)	Dependencies	Priority (low-med-high)
1	0%	Networking	#2			high
2	0%	text chat	#9			low
3	0%	Graphics				high
4	0%	multiplayer				high
5	0%	Show/indicate active player				high
6	0%	Display player scores				high
7	0%	transition consecutive rounds				high
8	0%	highlight token path	#11/#13			high
9	0%	display current pieces on board				high
10	0%	Addition of a piece	#6,8			high
11	0%	Movement of a piece	#6,8			high
12	0%	removal of a piece	#10			high
13	0%	display entire				high

		board		
14	0%	End turn	#5	high
15	99%	keyboard		high
16	0%	option to quit game		high
17	0%	Turn Timer	#5	med
18	0%	Start menu with options	#1	med
19	0%	3D graphics		med
20	0%	Show available pieces to play		high
21	0%	Sound effects		med
22	0%	Prevent unlawful move		med
23	0%	Option to access rules		low
24	0%	Background music		low
25	0%	Artificial Intelligence	#11/13, 12	low
26	0%	Theme explored		low
27	0%	Accommodate teams of 2		low
28	99%	game pad input		med
	0%	Animating the movement of the token after the player commits to a move.		low
	0%			low