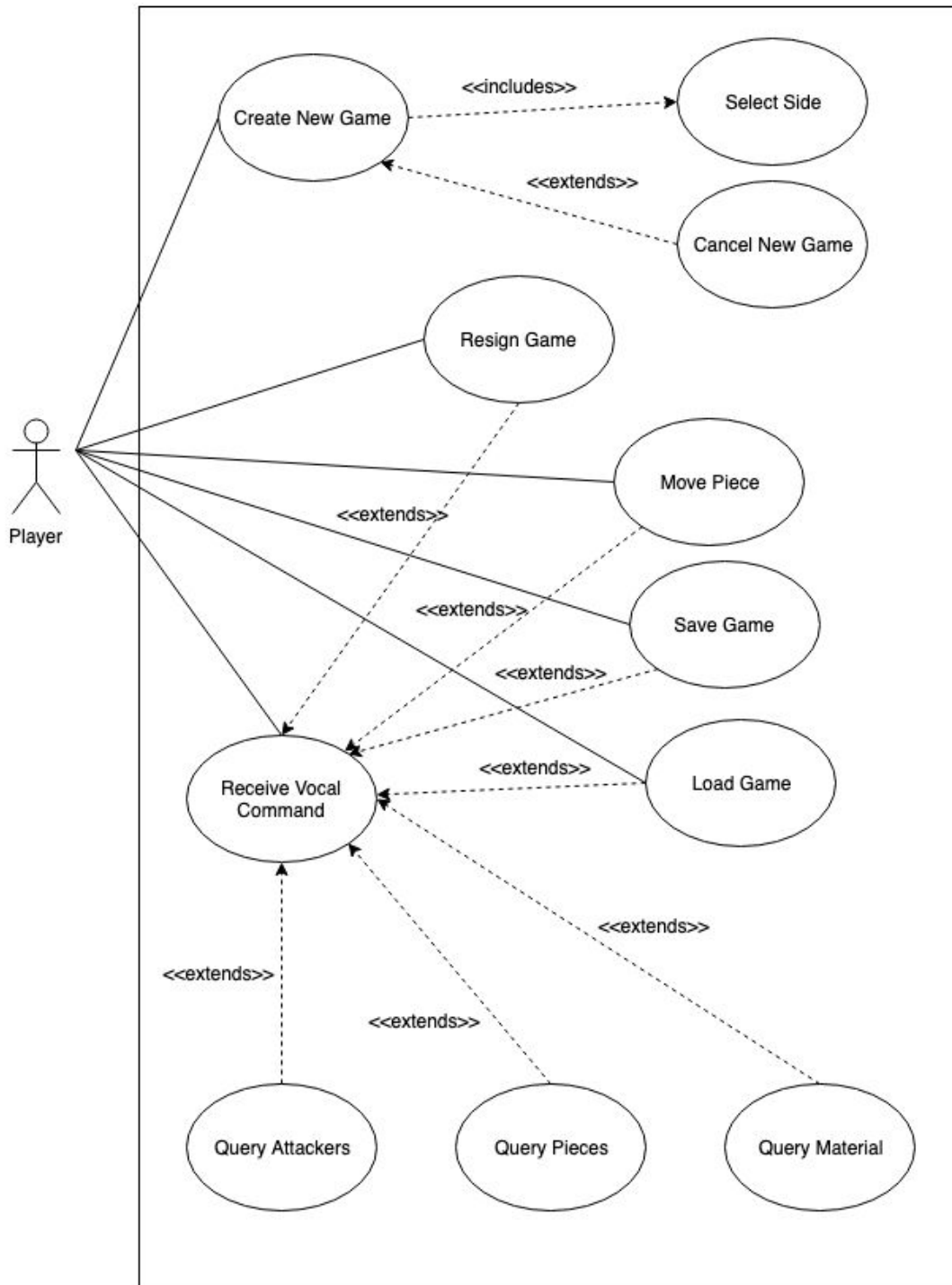


Use Cases

Team Checkmate

Kyle Davey
Ojasvi DSilva
David Thornton
Timothy VanSlyke



| Use Case Identification and History | | | |
|-------------------------------------|--------------------------------------------------------------------------|-----------------|--|
| Use Case ID | ReceiveVocalCommand | | |
| Use Case Name | Receive Vocal Command | Version Number | |
| End Objective | Accept a vocal command from the user and go to the correct next use case | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | Player issues a vocal command | | |
| Frequency of Use | This use case can be invoked an arbitrary number of times per game. | | |

| Preconditions |
|---------------|
| |

| Basic Flow | | |
|------------|-----------------------------|-------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues a voice command | System determines what command was issued |
| 2 | | System performs action |

| Exception Flow | | |
|----------------|---------------------------|-----------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues voice command | System notifies user that command was invalid |

| Post Conditions |
|-----------------|
| |

| Includes and Extension Points |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Extended by:</p> <ul style="list-style-type: none"> • QueryAttackers • QueryPiece • QueryMaterial • UndoMove • CreateNewGame • ResignGame • MovePiece • ExitGame • LoadGame |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|---------------------------------------------------------------------------------|
| This use case describes an accessibility feature for visually impaired players. |

| Use Case Identification and History | | | |
|-------------------------------------|------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | QueryAttackers | | |
| Use Case Name | Query Attackers | Version Number | |
| End Objective | Inform the player of which pieces on the board are currently attacking a position. | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | The player issues a vocal command to query which pieces are attacking a position. | | |
| Frequency of Use | The command may be issued an arbitrary number of times during a game. | | |

| Preconditions |
|-------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> There is a game currently active. |

| Basic Flow | | |
|------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query which pieces are attacking a position. | The system audibly lists each piece, along with its position and color, that is currently attacking the position. |

| Exception Flow | | |
|----------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query which pieces are attacking a position while there is no game active. | The system informs the user that there is no game active. |

| Post Conditions |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • User is then asked to make a move(Normal Flow) • The system preserves its previous state (exceptional flow). |

| Includes and Extension Points |
|-------------------------------|
| Extends: ReceiveVocalCommand |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|---------------------------------------------------------------------------------|
| This use case describes an accessibility feature for visually impaired players. |

| Use Case Identification and History | | | |
|-------------------------------------|-----------------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | QueryPiece | | |
| Use Case Name | Query Piece | Version Number | |
| End Objective | Inform the player what if there is a piece on a square on the board and what said piece is. . | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | The player issues a vocal command to query a position on the board. | | |
| Frequency of Use | The command may be issued an arbitrary number of times during a game. | | |

| Preconditions |
|-------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> There is a game currently active. |

| Basic Flow | | |
|------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query a position on the board. | The system verifies that a game is currently active and informs the user of what piece is currently located at the specified position, if there is one there. |

| Exception Flow | | |
|----------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query a position on the board while there is no active game. | The system audibly informs the player that there is no active game. |

| Post Conditions |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • User is then asked to make a move(Normal Flow) • The system preserves its previous state (exceptional flow). |

| Includes and Extension Points |
|-------------------------------|
| Extends: ReceiveVocalCommand |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|---------------------------------------------------------------------------------|
| This use case describes an accessibility feature for visually impaired players. |

| Use Case Identification and History | | | |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | QueryMaterial | | |
| Use Case Name | Query Material | Version Number | |
| End Objective | Inform the user of what pieces have been taken and state which player has a material advantage and by how much. | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | The player issues a vocal command to query what pieces they have taken and what pieces they have lost. | | |
| Frequency of Use | The command may be issued an arbitrary number of times during a game. | | |

| Preconditions |
|--------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> There is a game currently active. At least one piece in the game has been taken |

| Basic Flow | | |
|------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query the board on what pieces have been taken. | The system verifies that a game is currently active and informs the player which pieces they has taken and which pieces he has lost. |

| Exception Flow | | |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to query the board on what pieces have been taken when there is no active game. | The system audibly informs the player that there is no active game. |
| 2 | The player issues a vocal command to query the board on what pieces have been taken, when all the pieces are still on the board. | The system verifies that a game is currently active and informs the player that all the pieces are still on the board. |

| Post Conditions |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • User is then asked to make a move(Normal Flow) • The system preserves its previous state (exceptional flow). |

| Includes and Extension Points |
|-------------------------------|
| Extends: ReceiveVocalCommand |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|---------------------------------------------------------------------------------|
| This use case describes an accessibility feature for visually impaired players. |

| Use Case Identification and History | | | |
|-------------------------------------|-------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | CreateNewGame | | |
| Use Case Name | Create New Game | Version Number | |
| End Objective | Begin a new game either against the engine, or against another player | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | Vocal command to start a new game, or new game button | | |
| Frequency of Use | Once per game. Can occur an arbitrary number of times while the software is running | | |

| Preconditions |
|------------------------------------------|
| A new game can be initiated at any time. |

| Basic Flow | | |
|------------|-----------------------------------------------|--------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues command to start new game. | The system halts the currently-active game, if any, and asks user to specify game options. |
| 2 | User issues commands to specify game options. | The system starts new game, discarding the state of the previous game. |

| Exception Flow | | |
|----------------|-----------------------------------------------------|---------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User requests that the system abandon the new game. | The system discards the new game request and resumes the previous game, if any. |

| Post Conditions |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> User is then asked to select side (normal flow). The system preserves its previous state (exceptional flow). |

| Includes and Extension Points |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Includes: SelectSide, EndGame Extends: ReceiveVocalCommand Extended by: CancelNewGame |

| Special Requirements |
|------------------------------------------------------------------------------------------------------------------------|
| The system should only discard the currently-active game once the new game creation process has finished successfully. |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |

| Use Case Identification and History | | | |
|-------------------------------------|-----------------------------------------------------------------------|-----------------|--|
| Use Case ID | SelectSide | | |
| Use Case Name | Select Side | Version Number | |
| End Objective | Select side of the board to play on | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | After issuing CreateNewGame command, user will select side to play on | | |
| Frequency of Use | Occurs after each CreateNewGame | | |

| Preconditions |
|---------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> The user has confirmed they want to create a new game |

| Basic Flow | | |
|------------|------------------------------------------|-----------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User selects one of the sides to play on | The system then creates the game with the users pieces being on the selected side |

| Post Conditions |
|--------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Game is started with user playing from their selected side |

| Includes and Extension Points |
|-------------------------------|
| Included by: CreateNewGame |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |

| Use Case Identification and History | | | |
|-------------------------------------|--------------------------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | CancelNewGame | | |
| Use Case Name | Cancel New Game | Version Number | |
| End Objective | Cancel the process of creating a new game, and resume the current game if one is already active | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | User issues the command to cancel the game creation following the initial command to create a new game | | |
| Frequency of Use | Can occur any time during the CreateNewGame process | | |

| Preconditions |
|---------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> User is in the CreateNewGameProcess |

| Basic Flow | | |
|------------|----------------------------------------|---------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues command to cancel the game | System will abort the CreateNewGame process |

| Exception Flow | | |
|----------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The player issues a vocal command to cancel the game, when user is not in the CreateNewGame Process | The system audibly informs the player that they are not in the CreateNewGame process. |

| Post Conditions |
|-------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> User is given the option to Create a New Game(Normal Fow) |

| Includes and Extension Points |
|-------------------------------|
| Extends CreateNewGame |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |

| Use Case Identification and History | | | |
|-------------------------------------|------------------------------------------------------------|-----------------|--|
| Use Case ID | ResignGame | | |
| Use Case Name | Resign Game | Version Number | |
| End Objective | Concede defeat in the current active game and end the game | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | User issues command to resign the game | | |
| Frequency of Use | This use case can be invoked once per game. | | |

| Preconditions |
|---------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> There is a game currently active. It is the user's turn. |

| Basic Flow | | |
|------------|----------------------------------------------|------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The user issues a command to resign the game | The system moves to the EndGame use case |

| Exception Flow | | |
|----------------|----------------------------------------------------------|-------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | It is not the user's turn or there is not an active game | The system notifies the user of the error and no changes are made |

| Post Conditions |
|--------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> The system moves to the EndGame use case |

| Includes and Extension Points |
|---------------------------------------------------|
| Includes: EndGame Extends: ReceiveVocalCommand |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |

| Use Case Identification and History | | | |
|-------------------------------------|---------------------------------------------------------------------|-----------------|--|
| Use Case ID | MovePiece | | |
| Use Case Name | Move Piece | Version Number | |
| End Objective | Move a piece in the currently active game | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | User issues command specifying a move sequence. | | |
| Frequency of Use | This use case can be invoked an arbitrary number of times per game. | | |

| Preconditions |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • There is a game currently active. • The specified move sequence is valid for current half-turn (i.e. the specified piece belongs to the player whose turn it is). • The specified move sequence is not otherwise invalid (moving into check, failing to move out of check, or an invalid castle). • The current half-turn does not belong to the chess engine. |

| Basic Flow | | |
|------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The user issues a command to move a piece. | The system validates the command against the appropriate preconditions and updates the current game board accordingly. |

| Exception Flow | | |
|----------------|-----------------------------------------------------|------------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | The user issues an invalid command to move a piece. | The system notifies the user of the error and makes no changes to the board state. |

| Post Conditions |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> The requested piece is moved appropriately (normal flow). The game state is as it was previously (exceptional flow). Will invoke EndGame if the move results in checkmate |

| Includes and Extension Points |
|------------------------------------------------------|
| Extends: ReceiveVocalCommand Extended by: EndGame |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |

| Use Case Identification and History | |
|-------------------------------------|----------|
| Use Case ID | SaveGame |

| | | | |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|--|
| Use Case Name | Save Game | Version Number | |
| End Objective | The program will the save the current game state so that the user can return to it at a later time | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | User is exiting a game and wants to save the game state. | | |
| Frequency of Use | This use case can be invoked an arbitrary number of times per game, but will only be invoked if the user wants to exit the game and save the game state. | | |

| Preconditions |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • There is currently an active game to save. • Player is exiting the game and wants to save the game |

| Basic Flow | | |
|-------------------|------------------------|------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | From ExitGame use case | System asks the user if they would like to save |
| 2 | User selects "Yes" | System saves the game state, overwriting the previous save |

| Alternate Flow | | |
|-----------------------|---------------------|-----------------------|
| Step | User Actions | System Actions |

| | | |
|---|------------------------|-------------------------------------------------|
| 1 | From ExitGame use case | System asks the user if they would like to save |
| 2 | User selects “No” | System does not save the game state |

Post Conditions

The game can be loaded on command

Includes and Extension Points

Extends: ExitGame

Special Requirements

Business Rules

Other Notes

| Use Case Identification and History | | | |
|-------------------------------------|-----------------------------------------------------------------------------------------|-----------------|--|
| Use Case ID | LoadGame | | |
| Use Case Name | Load Game | Version Number | |
| End Objective | Load the saved game state | | |
| Created By | | On (date): | |
| Last Update By | | On (date): | |
| Approved By | | On (date): | |
| User/Actor | Player | | |
| Business Owner Name | | Contact Details | |
| Trigger | The user would like to load the saved game state | | |
| Frequency of Use | This use case can be invoked any number of times as long as there is a saved game state | | |

| Preconditions |
|-----------------------------|
| There is a saved game state |

| Basic Flow | | |
|------------|--------------------------------------------|-----------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues command loading the saved game | System notifies the user that the current game will be ended and asks to continue |
| 2 | User selects "Yes" | System loads the saved game state and ends the current game |

| Alternate Flow | | |
|----------------|--------------------------------------------|-----------------------------------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User issues command loading the saved game | System notifies the user that the current game will be ended and asks to continue |
| 2 | User selects “No” | System does not load game |

| Exception Flow | | |
|----------------|---------------------------------------------------|------------------------------------------------------|
| Step | User Actions | System Actions |
| 1 | User asks to load game but there is no saved game | System notifies the user that there is no saved game |

| Post Conditions |
|----------------------------|
| Saved game state is loaded |

| Includes and Extension Points |
|---------------------------------------------------|
| Extends: ReceiveVocalCommand Includes: EndGame |

| Special Requirements |
|----------------------|
| |

| Business Rules |
|----------------|
| |

| Other Notes |
|-------------|
| |