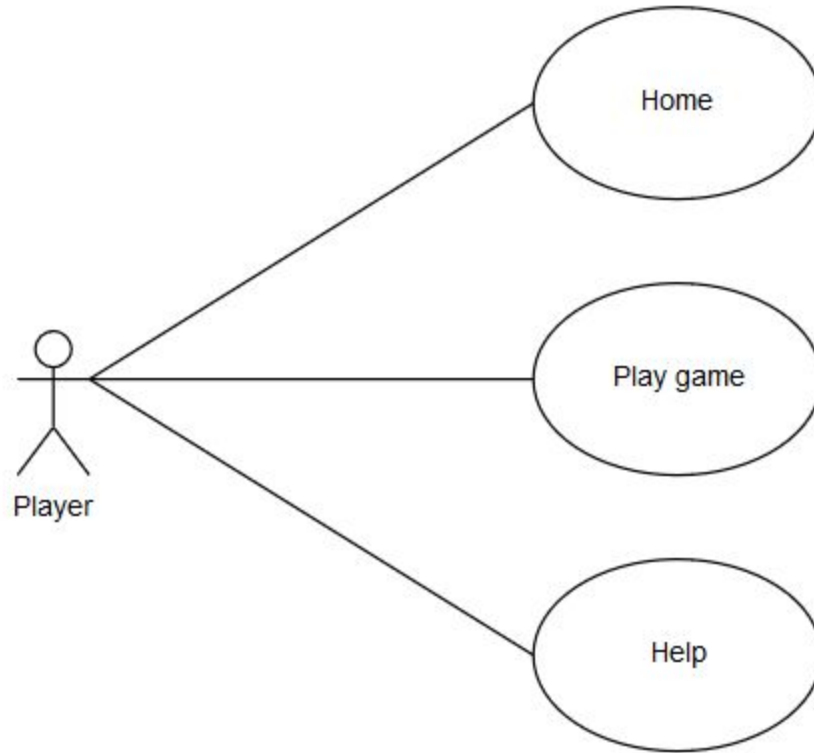


Use Case Document



Group E

Omar Perez

Chris Perez

Jesse Guardado

Table of Contents

- [1. Home button clicked](#)
- [2. Help button clicked](#)
- [3. Play game button clicked](#)
- [4. Add player](#)
- [5. Launch game](#)
- [6. Generate character](#)
- [7. Generate player order](#)
- [8. Shuffle deck](#)
- [9. Roll dice](#)
- [10. Accept dice](#)
- [11. Resolve dice](#)
- [12. Gain victory points from roll](#)
- [13. Gain heal points from roll](#)
- [14. Gain energy points from roll](#)
- [15. Gain smash points from roll](#)
- [16. Attack all players outside Tokyo](#)
- [17. Attack player inside Tokyo](#)
- [18. Monster dies](#)
- [19. Enter Tokyo City or Tokyo Bay](#)
- [20. Leave Tokyo City or Tokyo Bay](#)
- [21. Evacuating Tokyo Bay](#)
- [22. Buy power cards](#)
- [23. Replace power cards on table](#)
- [24. Get “keep” power card](#)
- [25. Get “Discard” power card](#)
- [26. Use power card](#)
- [27. Player wins](#)
- [28. Player loses](#)
- [29. Play again menu](#)
- [30. Roll green dice](#)
- [31. Use specialty mimic power card](#)
- [32. Use specialty poison power card](#)
- [33. Use specialty shrink power card](#)
- [34. Use specialty smoke cloud power card](#)
- [35. Receive token](#)
- [36. Cancel button pressed](#)
- [37. Pause menu button](#)
- [38. Resume game pressed](#)
- [39. Exit game pressed](#)
- [40. Game options pressed](#)

1 Use case: home button clicked

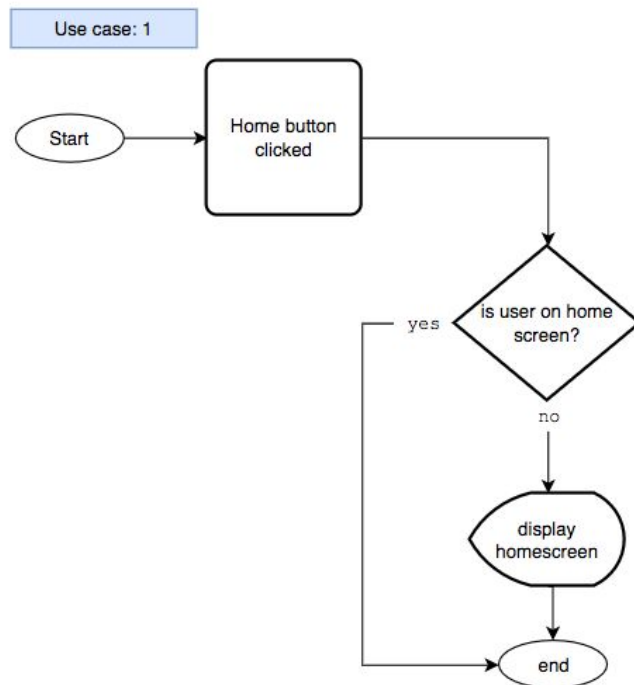
Actor: user and system

Pre-condition: launch application, be on application dashboard

Post-condition: system loads the home screen

Events:

1. Home button is pressed
 - a. Home button displays ripple effect
2. System displays the standard loading icon
3. Check to see if the user is already on home screen
4. System loads the HTML file for the home screen if user is not on home screen already



2 Use case: help button clicked

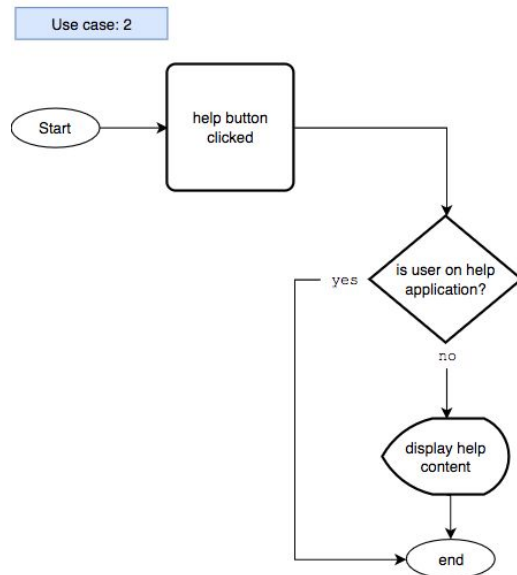
Actor: user and system

Pre-condition: launch application, be on the application dashboard

Post-condition: system loads the help section menu

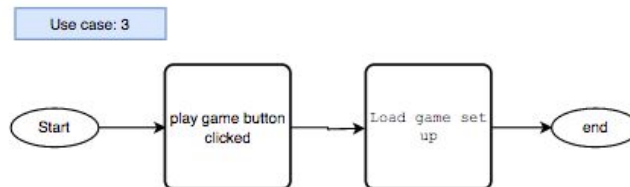
Events:

1. Help button is pressed
 - a. Help button displays a ripple effect
2. System displays the standard loading icon
3. Check to see if the user is already on the help screen
4. System loads the HTML file for the help screen if the user is not on the help screen already



3 Use case: game button clicked

- Actor: user and system
- Pre-condition: launch application, be on the application dashboard
- Post-condition: system loads the game set up
- Events:
 1. Game button pressed
 2. System loads the game screen set up
 - a. Check ()
- Extensions:
 3. Occurrence -- add player, game set up



4 Use case: add player

- Actor: user and system
- Pre-condition: in game setup menu, game setup has not been initialized
- Post-condition: Players name has been created and added
- Events:
 1. Enter first players name in the text box
 2. Enter second players name in the text box
 - a. Check (4)

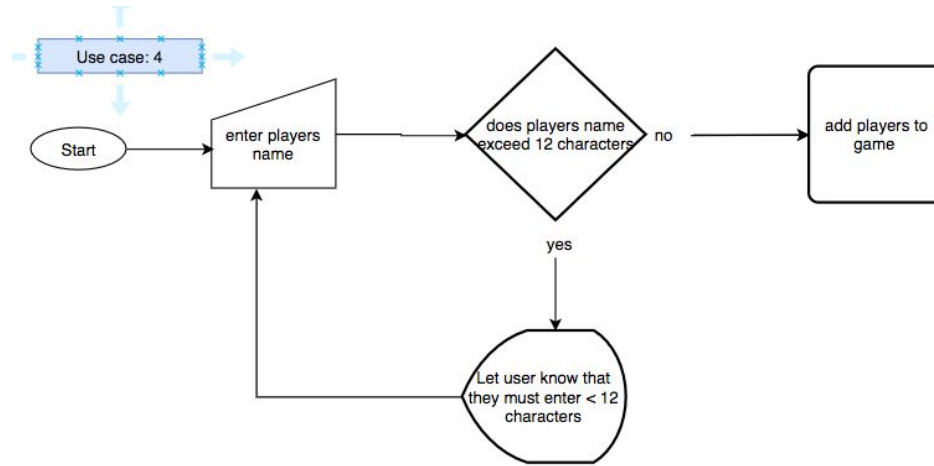
3. Press add players button

a. Check (5)

- Extensions:

4. Optional - add more players to the game -- up to 6

5. Exception - after each text input, determine if player name is too long -- max is 12 characters

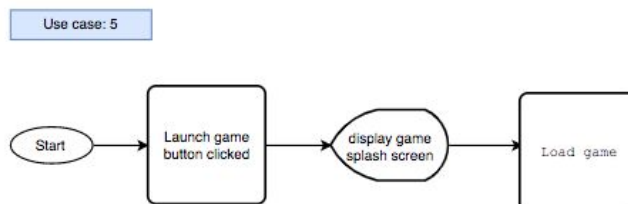


5 Use case: Launch Game button

- Actor: user and system
- Pre-condition: game setup has been completed
- Post-condition: game has launched and is ready for play
- Events:
 1. Press launch button
 2. Hide side panel
 3. System displays splash screen loader
 4. System loads the HTML file for the game

- Extensions:

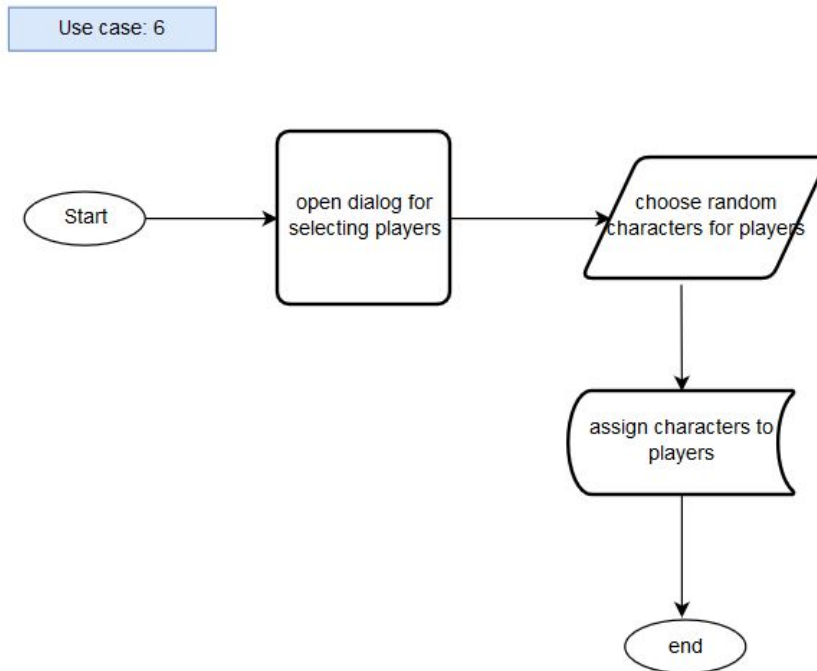
5. Exception -- game failed to launch due to hardware or system error.



6 Use case: generate character

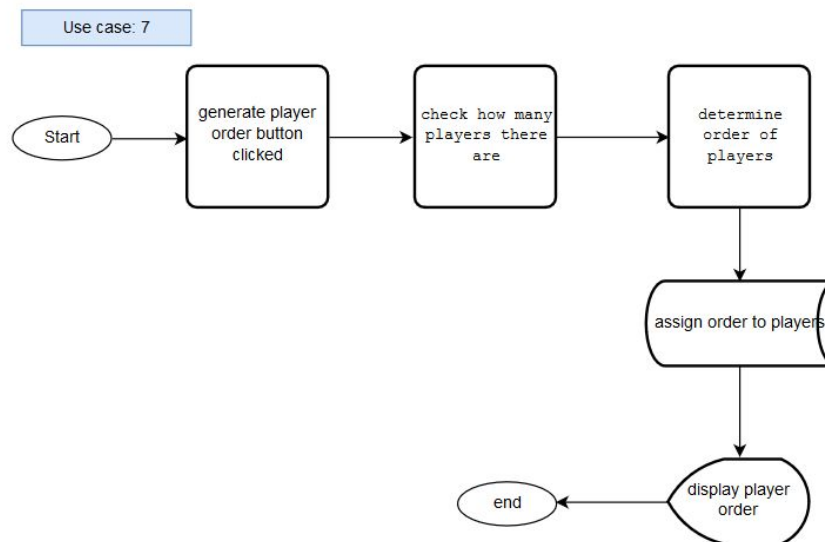
- Actor: user and system
- Pre-condition: game has successfully launched
- Post-condition: each player has a character chosen
- Events:
 1. Open dialog for selecting players
 2. Randomly choose characters for players
 - a. Check (4)
 3. Assign characters to players

- Extensions:
 4. Optional -- select choice of character



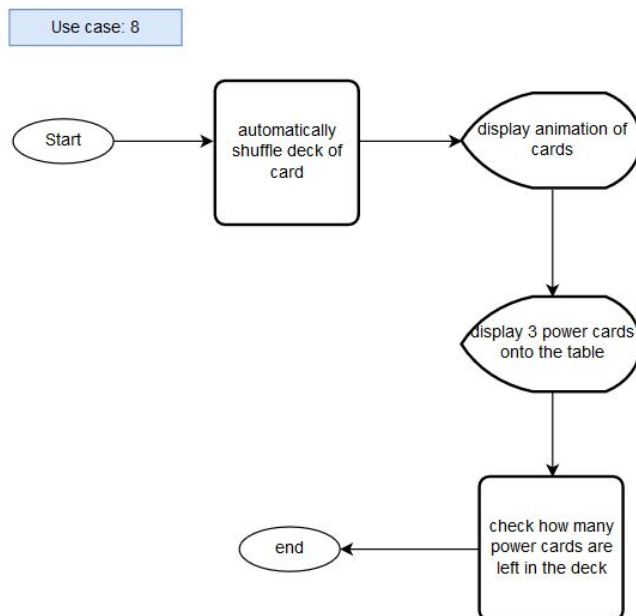
7 Use case: generate player order

- Actor: user and system
- Pre-condition: characters have been assigned to players
- Post-condition: the players order is determined
- Events:
 1. Press generate player order button
 2. Check the number of players
 3. Calculate order of players
 4. Display order of players



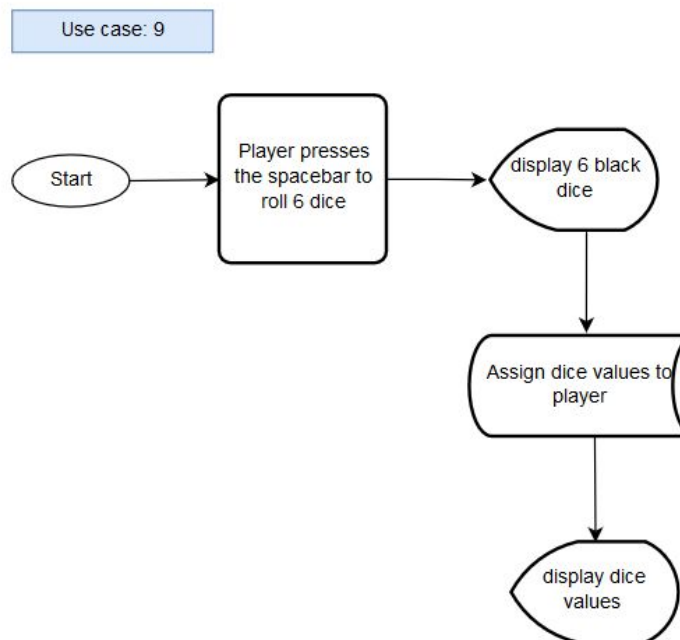
8 Use case: shuffle deck

- Actor: system
- Pre-condition: game loaded, order of players determined
- Post-condition: deck of cards have been randomly shuffled
- Events:
 1. Automatically shuffle the deck of cards
 2. Display the animated drop of shuffled cards
 3. Check to see how many power cards are left in the deck of cards
 4. Display 3 power cards on table



9 Use case: roll dice

- Actor: user and system
- Pre-condition: players turn to roll dice
- Post-condition: player rolls dice and determines dice values
- Events:
 1. Player presses the spacebar to roll the 6 black dice
 2. 6 black dice rolls onto the table
 3. Dice value is gained
 4. Display value of dice obtained on screen
 - a. Check (6)
 - b. Check (7)
 - c. Check (8)
 - d. Check (9)
 - e. Check (10)
 5. The value is assigned to the player
- Extensions:
 6. Choose which dice to keep
 7. Occurrence -- player gets n of a kind where $n = [3, 6]$
 8. Occurrence -- gain 1 energy point for each energy face shown
 9. Occurrence -- gain 1 health point for each heart face shown
 10. Occurrence -- gain 1 smash point for each claw shown

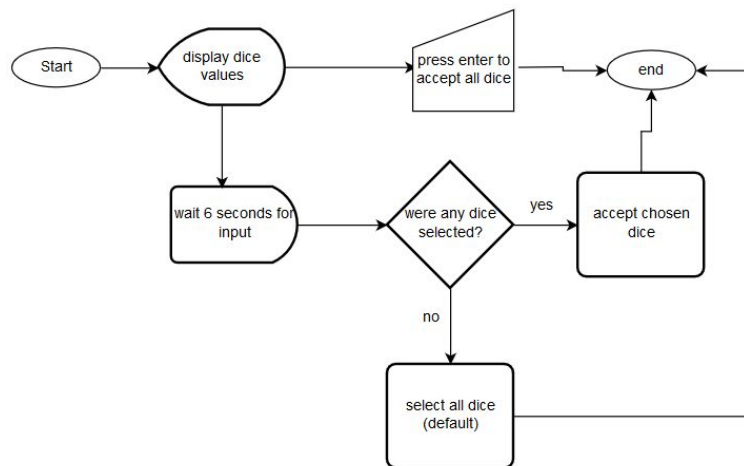


10 Use case: accept dice

- Actor: user and system
- Pre-condition: dice has been rolled
- Post-condition: selected dice is kept
- Events:
 1. Dialog pops up and displays which dice values were face up to select

2. Timer countdown runs for 6 seconds to choose desired dice
 - a. Check (5)
 3. Choose which dice to keep
 - a. Check (6)
 - b. Check (7)
 4. Press enter to conclude acceptance of dice or wait for timer to rundown
- Extensions:
 5. Occurrence -- If no dice were selected, all dice are kept after timer finishes (default acceptance)
 6. Occurrence -- press keybind (A) to accept all dice (default acceptance)
 7. Optional -- press 1, 2, 3, 4, 5, or 6, or click desired dice to select which dice to keep

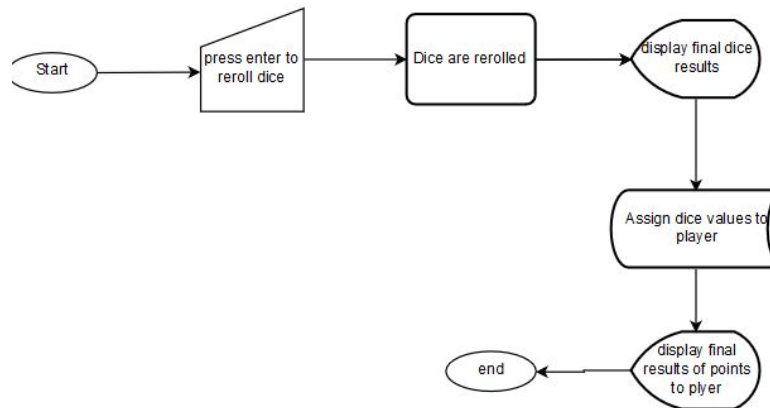
Use case: 10



11 Use case: resolve dice

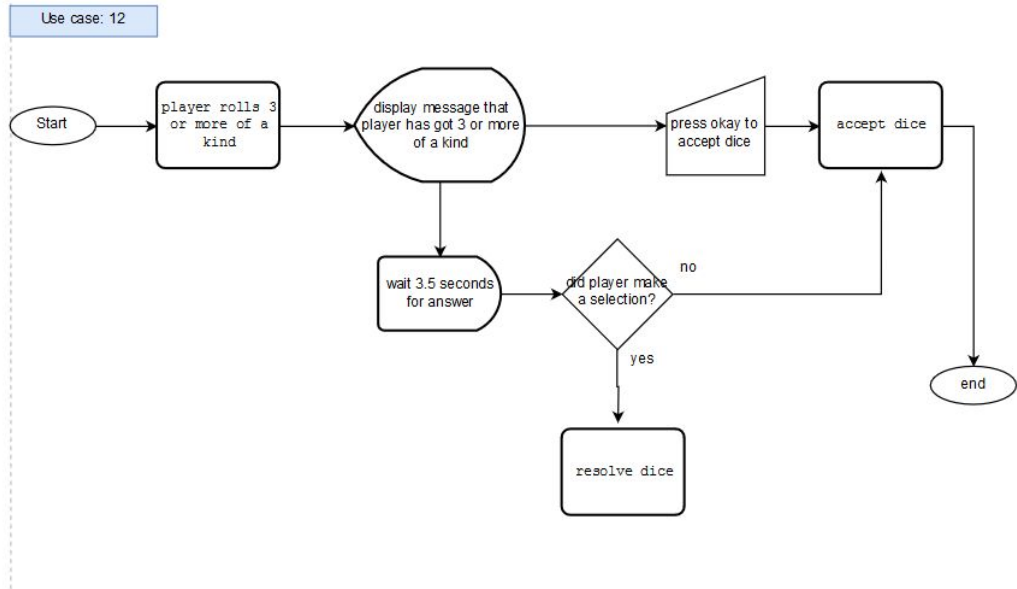
- Actor: user and system
- Pre-condition: not all dice were accepted in a roll
- Post-condition: all dice have been resolved
- Events:
 1. Click reroll button or the keybind enter
 2. Dice are rerolled
 3. Dialog pops up and displays final dice results
 4. Desired dice have been selected and given to player
 5. Display the players victory, heal, smash, or energy points-- or all that are gained
 6. Players dice are resolved

Use case: 11



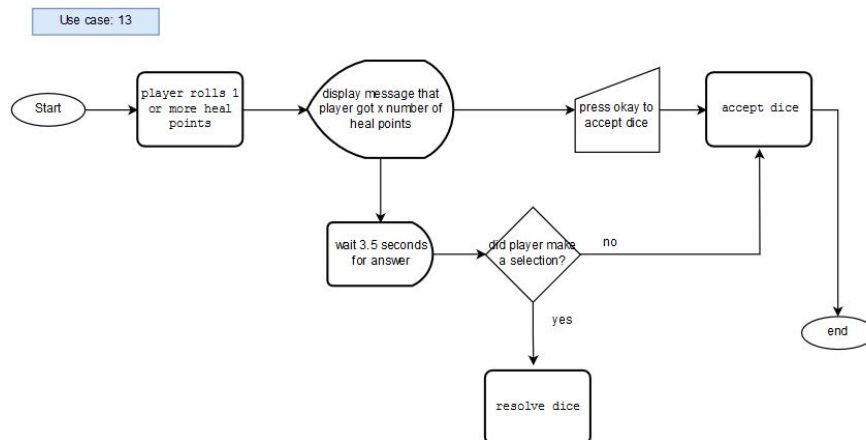
12 Use case: gain victory points from roll

- Actor: user and system
- Pre-condition: player rolls 3 or more of a kind
- Post-condition: player accumulates x amount of victory points
- Events:
 1. Check to see how many victory points players has
 2. Dialog displays that the player has gotten 3 or more of a kind
 3. Player presses the button okay in response or waits for the dialog to disappear in 3.5 seconds
 4. Player can choose to accept the 3 or more of a kind or choose to resolve the dice
 - a. Check (7)
 - b. Check (8)
 5. Victory points are gained
 6. Victory points are displayed and added to the victory point score
- Extensions:
 7. Optional -- accept dice
 8. Optional -- resolve dice



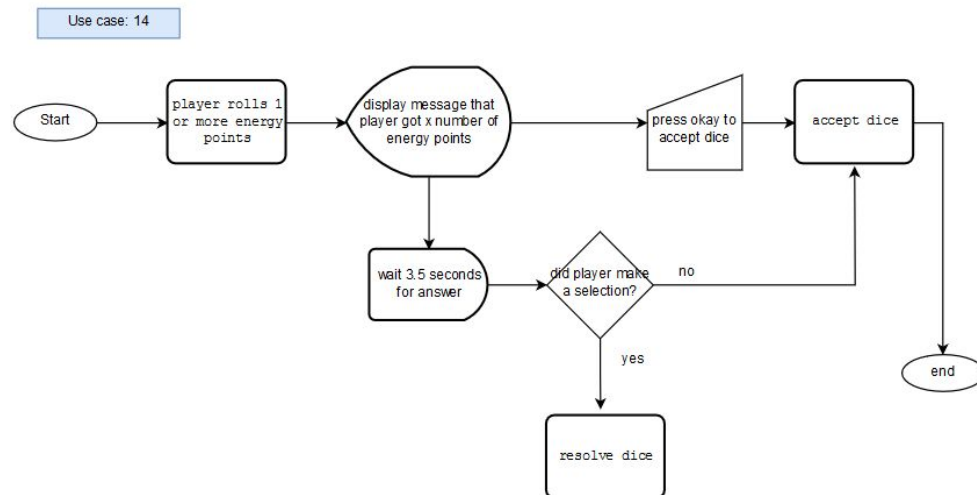
13 Use case: gain heal points from roll

- Actor: user and system
- Pre-condition: player rolls dice with heal value, player cannot have more than 10 heal points
- Post-condition: player accumulates x amount of heal
- Events:
 1. Calculate the number of heal points in a roll
 2. Check to see how many heal points player has
 3. Dialog displays how many heal points player has gained
 4. Player presses the button okay in response or waits for the dialog to disappear in 3.5 seconds
 5. Player can choose to accept heal points
 - a. Check (8)
 - b. Check (9)
 6. Heal points are gained
 7. Heal points are displayed and added to the heal point score
- Extensions:
 8. Optional -- accept dice
 9. Optional -- resolve dice



14 Use case: gain energy points from roll

- Actor: user and system
- Pre-condition: player rolls dice with energy value
- Post-condition: player accumulates x amount of energy cubes
- Events:
 1. Calculate the number of energy points in a roll
 2. Dialog displays how many energy points player has gained
 3. Player presses the button okay in response or waits for the dialog to disappear in 3.5 seconds
 4. Player can choose to accept energy points
 - a. Check(7)
 - b. Check (8)
 5. Energy points are gained
 6. Energy points are displayed and added to the energy point score
- Extensions:
 7. Optional -- accept dice
 8. Optional -- resolve dice



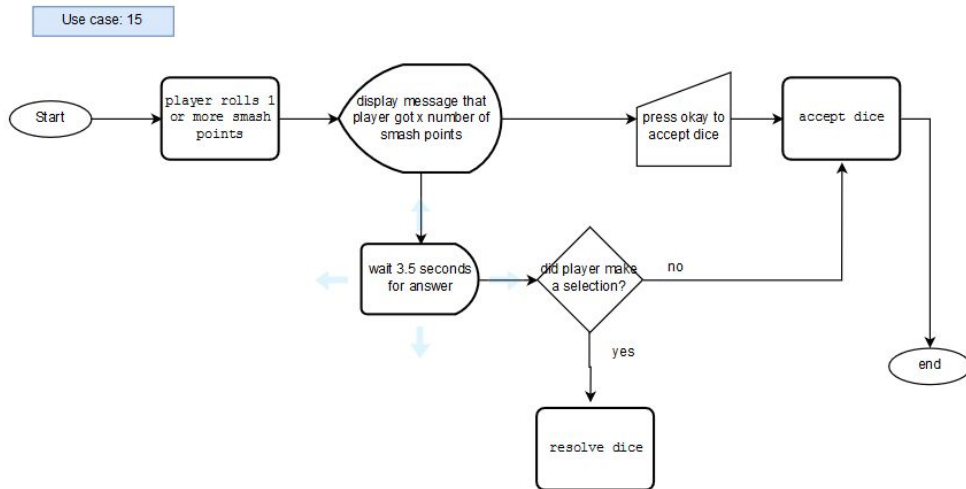
15 Use case: gain smash points from roll

- Actor: user and system
- Pre-condition: players have all passed round 1, player rolls dice with smash value
- Post-condition: player accumulates x amount of smash points, player can attack
- Events:
 1. Calculate the number of smash points in a given roll
 2. Dialog displays how many smash points player has gained
 3. Player presses the button okay in response or waits for the dialog to disappear in 3.5 seconds
 4. Check how many points player has
 - a. Cannot be more than 10 points
 5. Player can choose to accept smash points
 - a. check(8)
 - b. check(9)

6. Smash points are gained
7. Smash points are displayed
 - a. check(10)

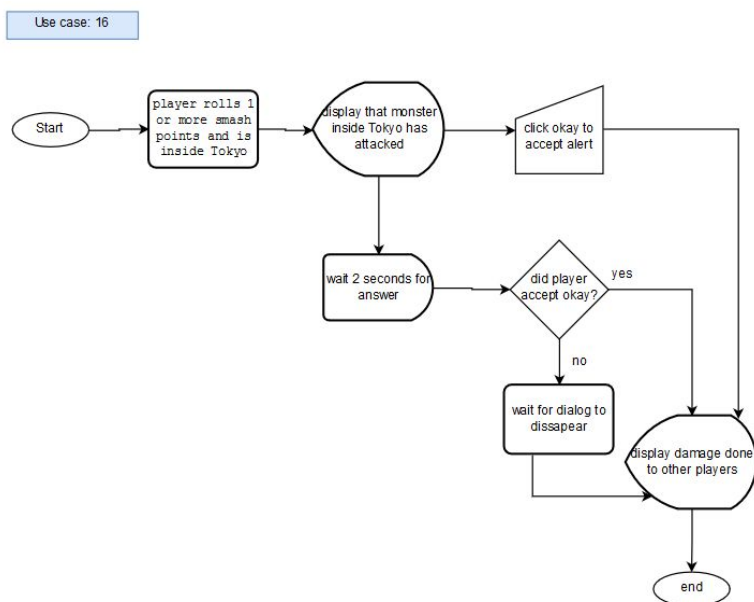
- Extensions:

8. Optional -- accept dice
9. Optional -- resolve dice
10. Occurrence -- attack opponents



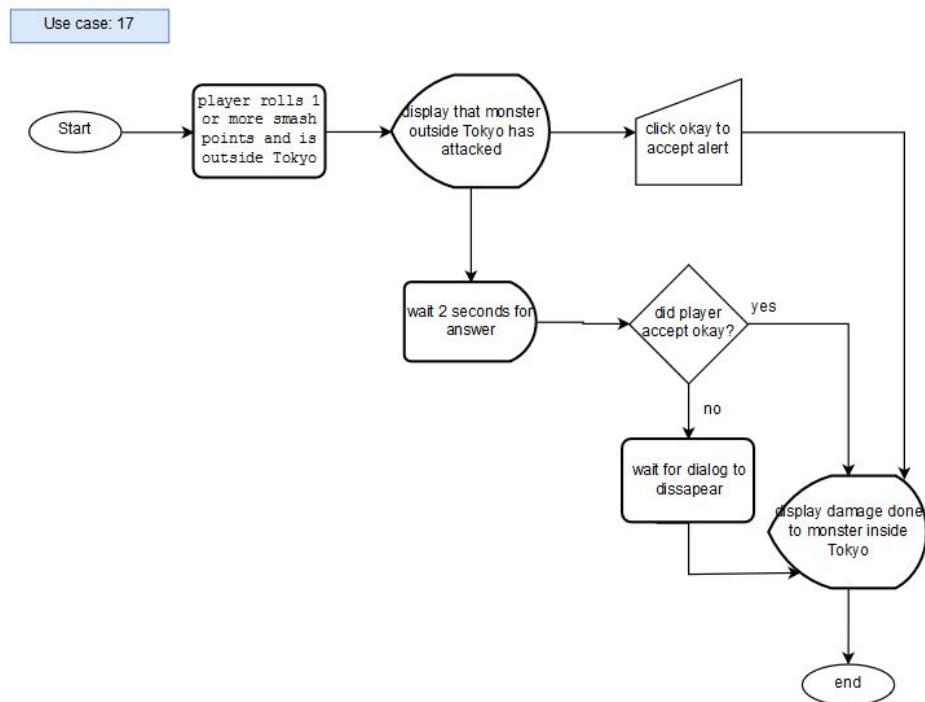
16 Use case: attack all players outside Tokyo

- Actor: user and system
- Pre-condition: player rolled dice and gained x amount of smash points, player is inside tokyo
- Post-condition: player attacked opponents outside of tokyo
- Events:
 1. Dialog displays that player has attacked players outside of tokyo x amount of times
 2. Player presses the button okay in response or waits for the dialog to disappear in 2 seconds
 3. Another dialog displays the damage it did to the other players outside of tokyo



17 Use case: attack players inside Tokyo

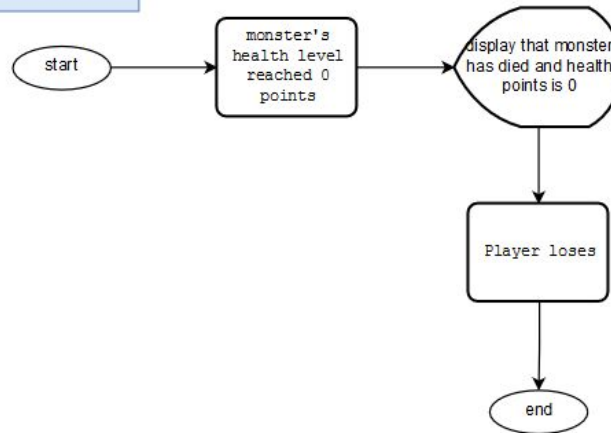
- Actor: user and system
- Pre-condition: player rolled dice and gained x amount of smash points, player is outside of tokyo
- Post-condition: player attacks opponents inside of tokyo
- Events:
 1. Dialog displays that player has attacked players inside of tokyo x amount of times
 - a. Check (4)
 2. Player presses the button okay in response or waits for dialog to disappear in 2 seconds
 3. Another dialog displays the damage it did to the players inside tokyo
- Extensions:
 4. Optional -- Player inside tokyo can choose to leave if damaged



18 Use case: monster dies

- Actor: system
- Pre-condition: player has dropped to 0 victory points
- Post-condition: Player loses the game, tokyo is now available to other players
- Events:
 1. Player has taken damage to 0 health points
 2. Dialog displays that players healthp points is 0
 - a. Check (3)
- Extensions:
 3. Occurrence -- player loses

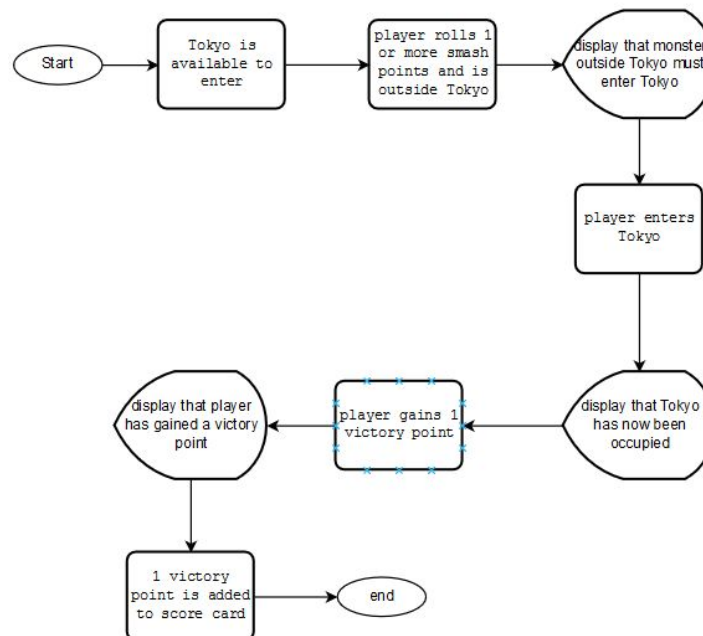
Use case: 18



19 Use case: enter Tokyo City or Tokyo Bay

- Actor: user
- Pre-condition: tokyo must be available to enter, all dice have been resolved and you have smash points
- Post-condition: player is now inside tokyo
- Events:
 1. Player rolls a smash
 2. Dialog displays that the player must enter tokyo (does it automatically)
 3. Player enters tokyo
 4. Pop up displays that player has entered tokyo
 5. Player gains 1 victory point for entering tokyo
 6. Dialog displays that player has gained a victory point
 7. 1 victory point is added to score card for player

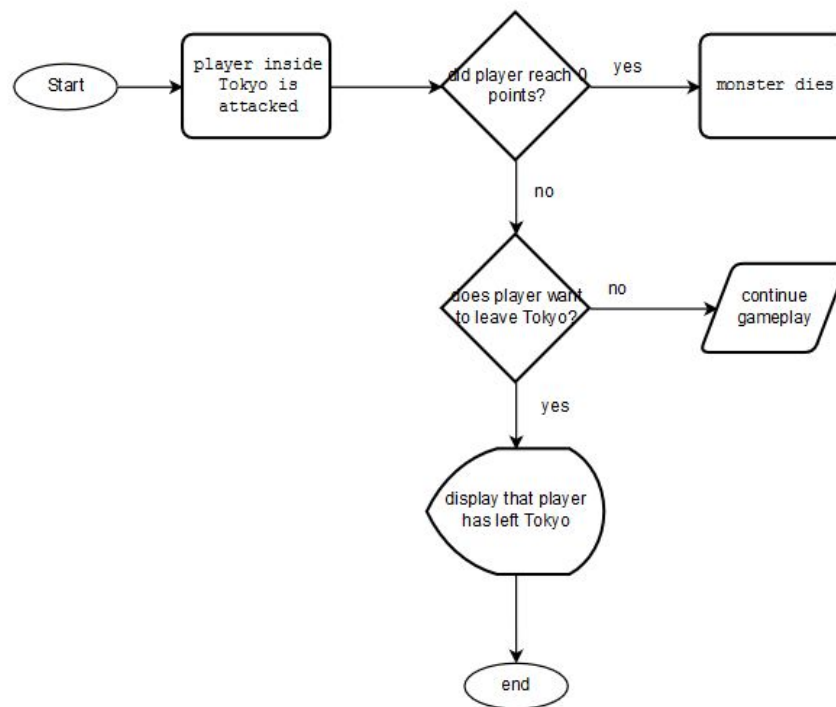
Use case: 19



20 Use case: leave Tokyo City or Tokyo Bay

- Actor: user and system
- Pre-condition: it is not players turn, player is attacked by the opponent, player is killed
- Post-condition: player is now not occupying tokyo
- Events:
 1. Player is attacked
 - a. Check (4)
 2. Popup displays that player has left tokyo
 - a. Check (5)
 3. Player is moved to outside tokyo
- Extensions:
 4. Occurrence -- monster dies
 5. Occurrence -- tokyo available to enter

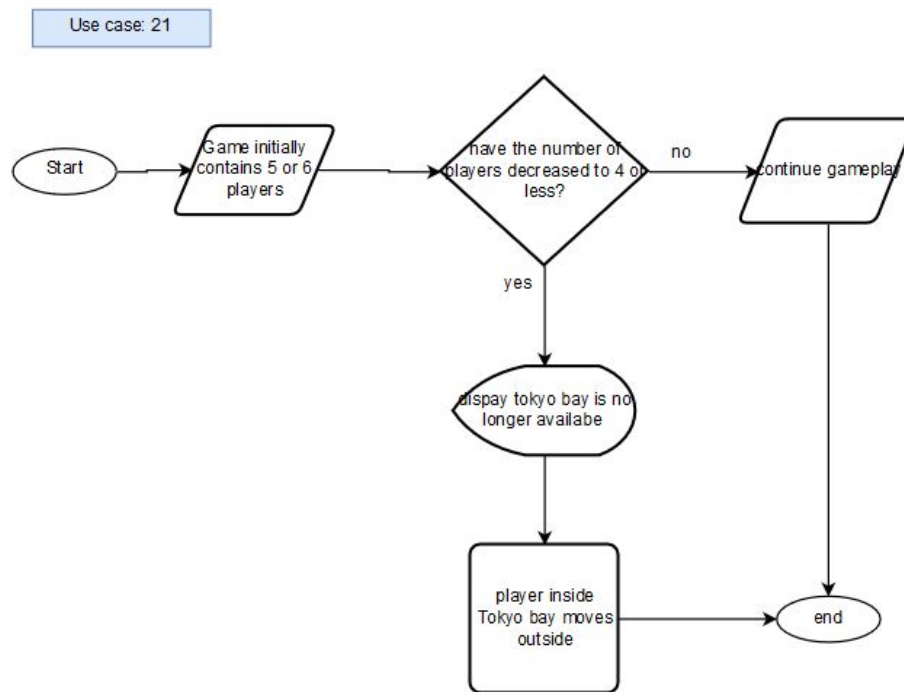
Use case: 20



21 Use case: evacuating tokyo bay

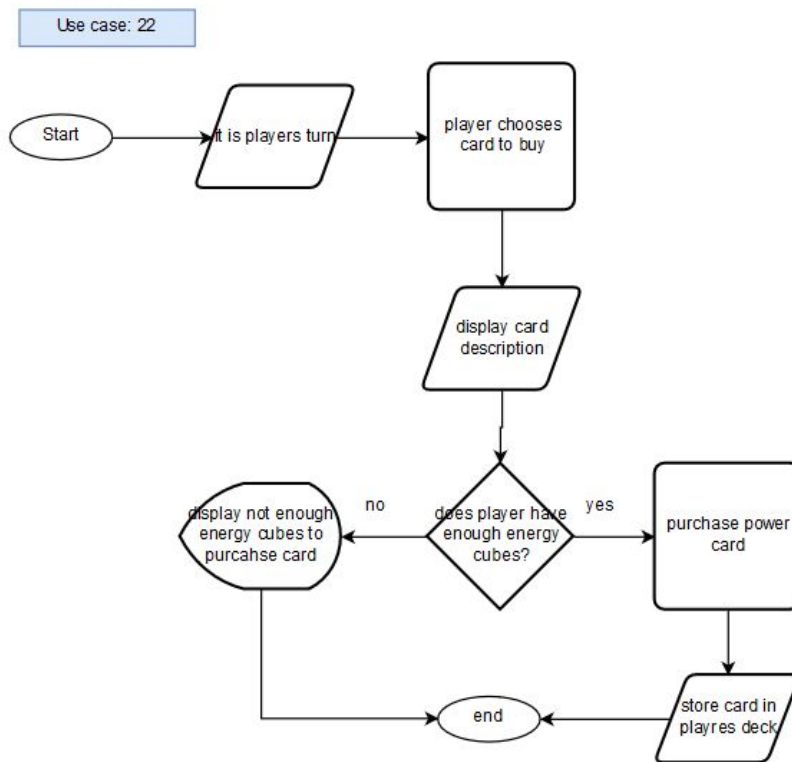
- Actor: system
- Pre-condition: the number of players initially contained more than 4 players, the number of players alive has decreased to 4 players
- Post-condition: tokyo bay is not available to enter anymore
- Events:
 1. 1 or 2 players have died
 2. Player inside tokyo bay must evacuate
 3. Display pop up of "evacuating tokyo bay"
 - a. Only if player is inside tokyo bay
 - b. If no player is occupying tokyo bay, display that tokyo bay is now unavailable

4. Player is moved outside of tokyo
5. Tokyo bay is disregarded from the map



22 Use case: buy power cards

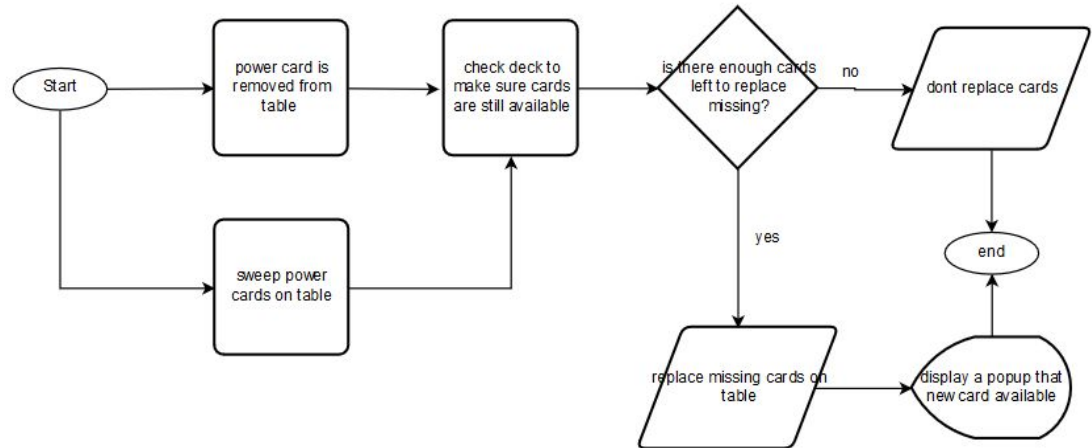
- Actor: user
- Pre-condition: must have sufficient funds (energy cubes) to purchase power cards, players turn, player rolled dice and resolved, power cards must be available to purchase
- Post-condition: 1 or more power cards have been purchased per turn
- Events:
 1. Check how many energy points the player has
 2. Player checks to see which power cards are available to purchase
 - a. Check (4)
 - b. Check (5)
 3. Player purchases 1 or more cards
 - a. Check (6)
- Extensions:
 4. Optional -- sweep power cards
 5. Occurrence -- show description of power card
 6. Occurrence -- replace power cards (immediately) if available



23 Use case: replace power cards on table

- Actor: system
- Pre-condition: power cards must be available (in stock)
- Post-condition: replaced 1 or more missing cards with new cards from the deck
- Events:
 1. Power card is removed from the table
 - a. Check (5)
 - b. Check (6)
 2. Check to see if there are enough power cards to replace the missing ones shown on the table
 3. Replace missing cards on table
 4. Display a popup that says “new power cards available to be purchased!” and disappears in 1.5 seconds
- Extensions
 5. Occurrence -- power card purchased
 6. Optional -- sweep power cards

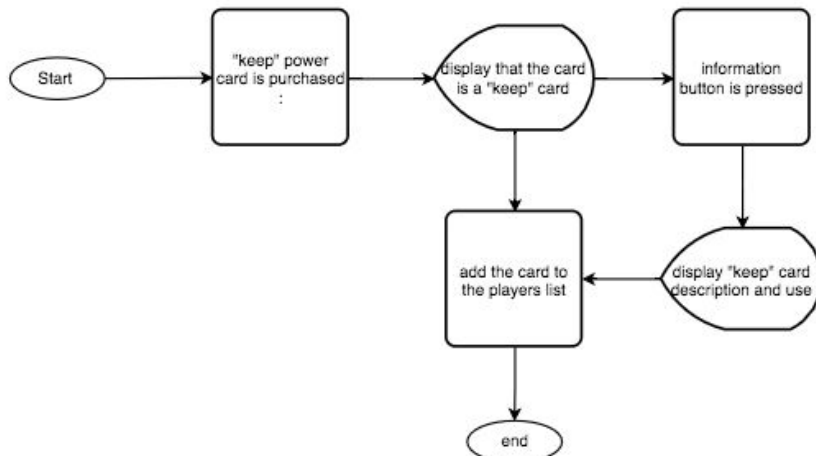
Use case: 23



24 Use case: get "keep" power card

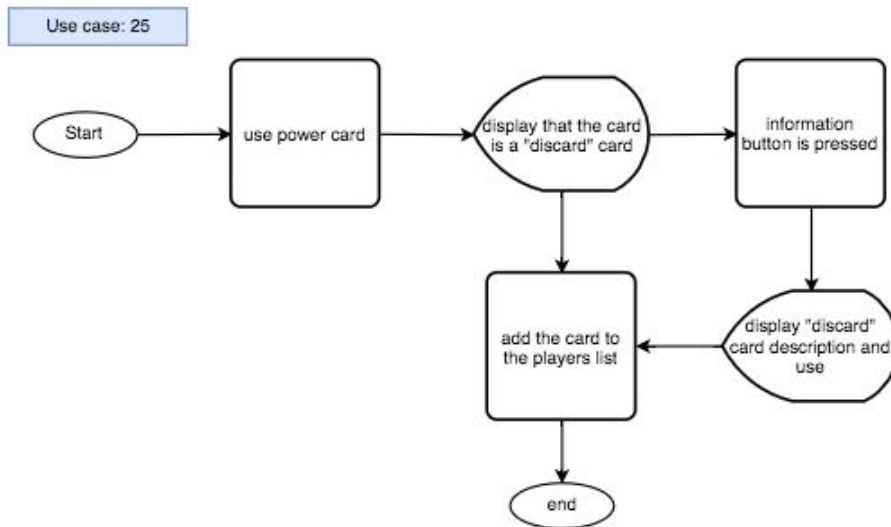
- Actor: user and system
- Pre-condition: user must have purchased a power card that has a "keep" tag
- Post-condition: player keeps the power card the entire game
- Events:
 1. Display that the card purchased is a "keep" card
 2. Display an information symbol button and let the user know that they can keep the card throughout the whole game
 3. Add power card to the list of cards the player has obtained
 - a. check(4)
- Extensions:
 4. Optional -- use power card

Use case: 24



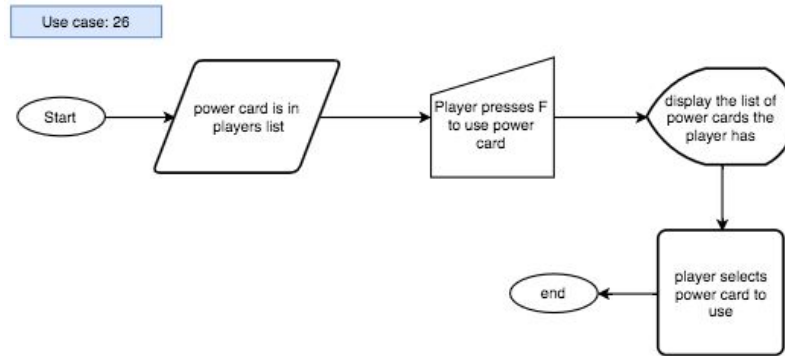
25 Use case: get "discard" power card

- Actor: user and system
- Pre-condition: user must have purchased a power card that has a "discard" tag
- Post-condition: players power card is discarded from the game immediately after use
- Events:
 1. Display that the card purchase is a "discard" card
 2. Display an information symbol button and let the user know that they must discard the card immediately after use
 - a. Check (3)
- Extensions:
 3. Optional -- use powercard



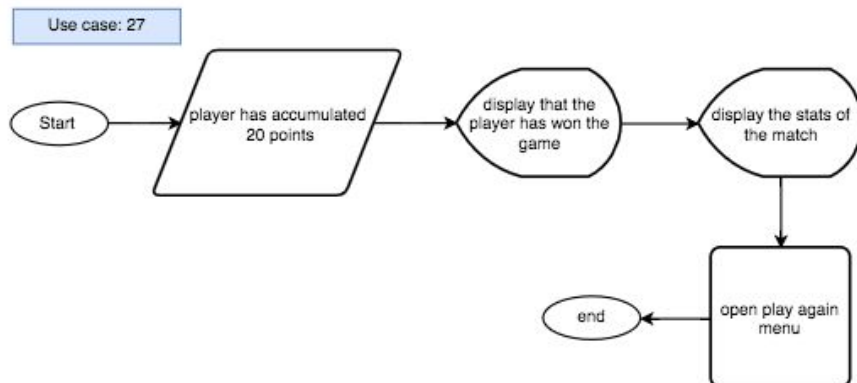
26 Use case: use power card

- Actor: user and system
- Pre-condition: player has purchased the power card, it is in players list of cards
- Post-condition: effect of power card has been used
- Events:
 1. Player clicks button that says "Use power card" or uses the keybind F to use powercard
 2. Dialog pops up and asks which power card the player will like to use from the list of purchased cards
 - a. Check (6)
 - b. Only if they have more than power card display this
 3. Power card is selected
 4. Popup closes
 5. Power card is used
 - a. Check (7)
- Extensions:
 6. Optional -- cancel button clicked
 7. Occurrence -- power card effect and properties is initiated



27 Use case: player wins

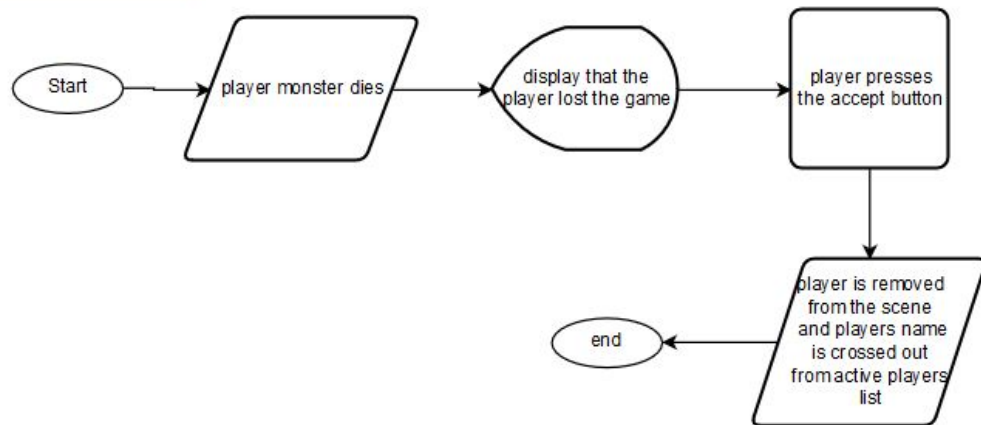
- Actor: user
- Pre-condition: player has accumulated 20 points and contained at least one health point or all other players have died
- Post-condition: player wins the game
- Events:
 1. Display a pop up for player stating "Winner! Player has won the game. The game is over."
 2. The game is over
 3. Display a dialog of the players stats in order (most to least victory points) and which player was killed off
 - a. Check (4)
- Extensions:
 4. Occurrence -- play again menu



28 Use case: player loses

- Actor: user
- Pre-condition: players monster dies
- Post-condition: player loses the game
- Events:
 1. Display a pop up for player stating "Your monster has died, you lose"
 2. Player clicks the accept button (no other option)
 3. Player is removed from the scene of the game
 4. Players name is crossed out from the active players list

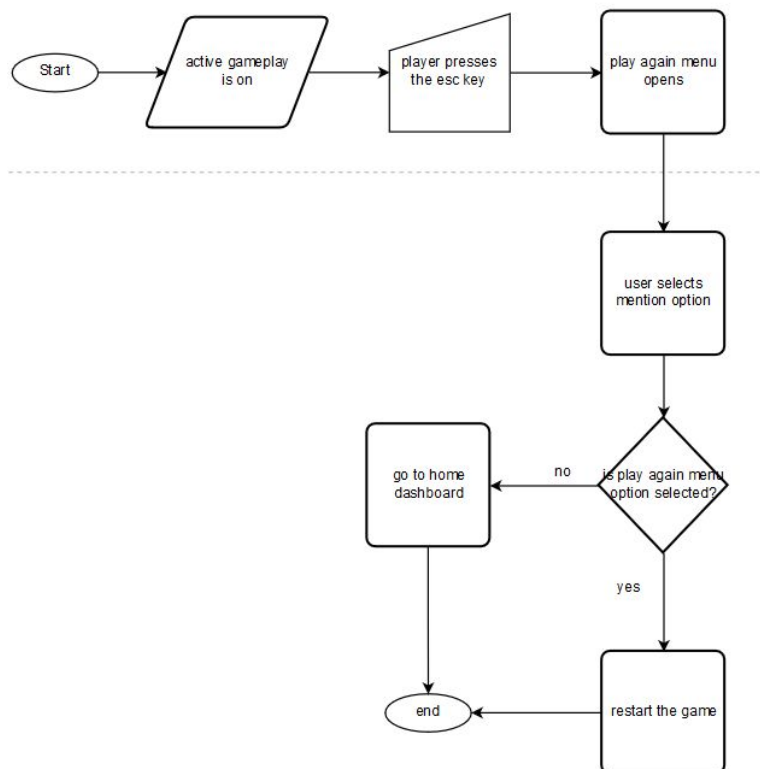
Use case: 28



29 Use case: play again menu

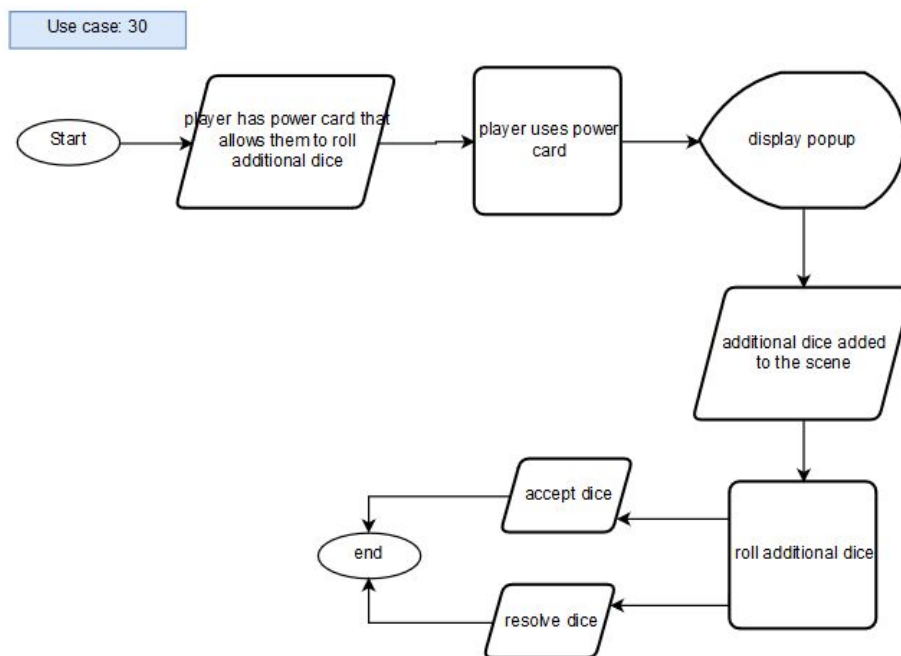
- Actor: user and system
- Pre-condition: the game is finished
- Post-condition: new game begins or ends
- Events:
 1. Display play again menu with a “yes, with same players” or “Yes, with different players” or “No, go to dashboard”
 2. User selects option
 3. Pop up closes and option is initiated
 4. Load loading icon for option initiated

Use case: 29



30 Use case: roll green die

- Actor: user
- Pre-condition: must have purchased power cards that allow player to gain additional dice to roll, must be players turn
- Post-condition: roll additional dice
- Events:
 1. Player uses powercard
 - a. Check (5)
 2. Add additional green dice to the scene
 3. Display popup letting the user know that additional dice have been added to the players roll
 4. Roll additional green dice
 - a. Check (6)
 - b. Check (7)
- Extensions:
 5. Occurrence -- use power card
 6. Optional -- accept dice
 7. Optional -- resolve dice



31 Use case: use specialty mimic power card

- Actor: user and system
- Pre-condition: player must have purchased a power card with the mimic specialty
- Post-condition: mimic token assigned to player, card copies effect of their power card
- Events:
 1. Player gets mimic specialty power card
 - a. Check (6)
 2. Display popup that the mimic specialty power card is now in use
 3. Display information button to describe mimic card and the specialty effect of it to the player
 4. Distribute mimic token to other player

a. Check (7)

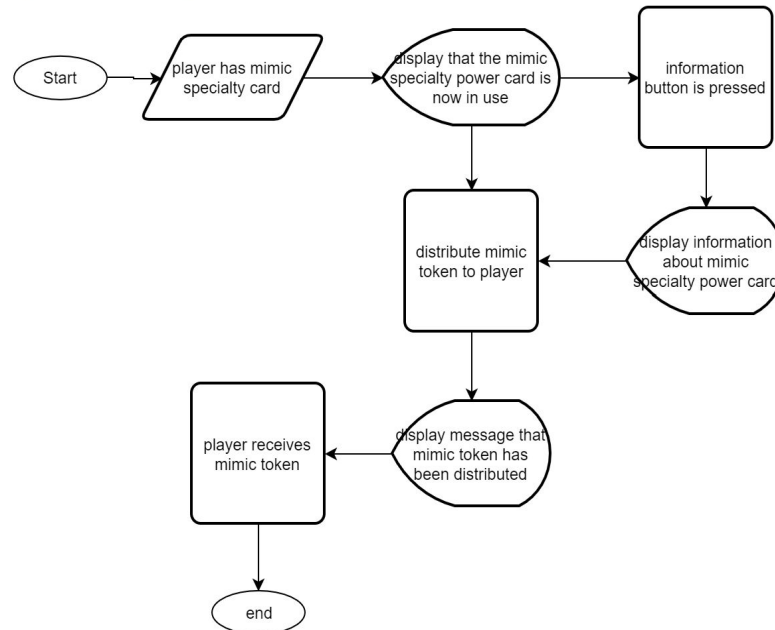
5. Display message to user that mimic token has been distributed

- Extensions:

6. Occurrence -- use power card

7. Occurrence -- player receives token

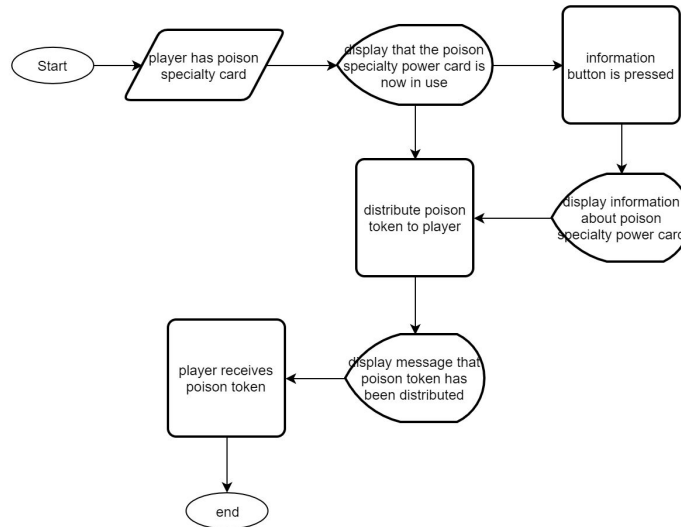
Use case: 31



32 Use case: use specialty poison power card

- Actor: user and system
- Pre-condition: player must have purchased a power card with the poison specialty
- Post-condition: card assigns a poison token to opponent, do damage to players with token
- Events:
 - Player gets poison specialty power card
 - Check (6)
 - Display popup that a poison specialty card is now in use
 - Display information button to describe poison specialty and the effects of it to the player
 - Distribute poison tokens to other players
 - Check (7)
 - Display message to user that poison token has been distributed
- Extensions:
 - Occurrence -- use power card
 - Occurrence -- player receives token

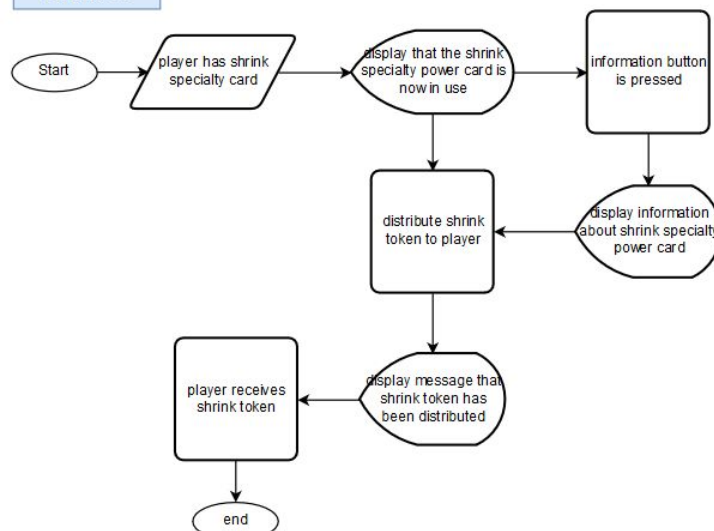
Use case: 32



33 Use case: use specialty shrink power card

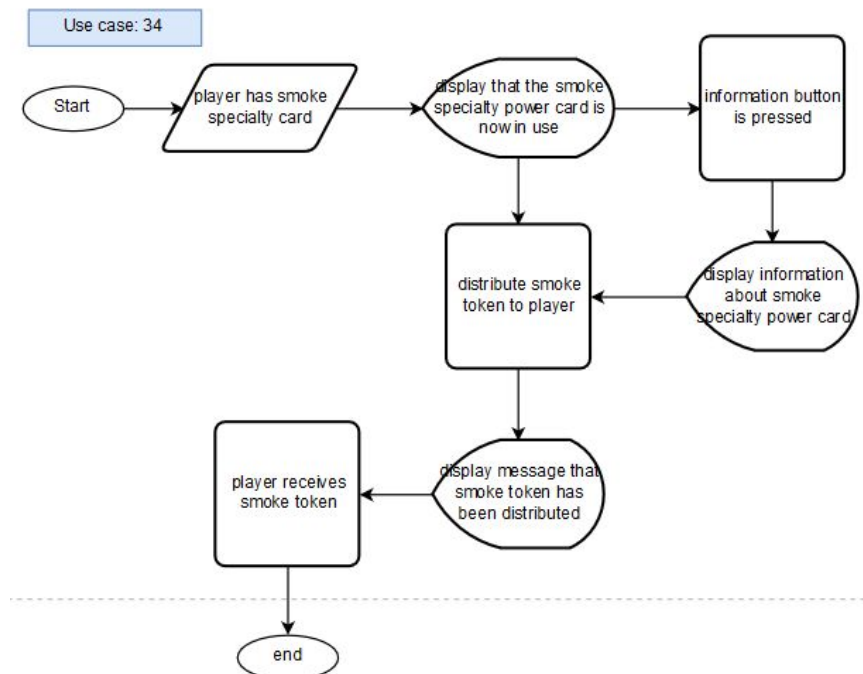
- Actor: user and system
- Pre-condition: player must have purchased a power card with the shrink specialty
- Post-condition: card assigns a shrink token to opponent, opponent rolls with one less dice
- Events:
 1. Player gets shrink specialty power card
 - a. Check(6)
 2. Display popup that shrink specialty card is now in use
 3. Display information button to describe shrink specialty and the effects of it to the player
 4. Distribute shrink tokens to other players
 - a. Check(7)
 5. Display message to user that shrink tokens have been distributed
- Extensions:
 6. Occurrence -- use power card
 7. Occurrence -- player receives token

Use case: 33



34 Use case: use specialty smoke cloud power card

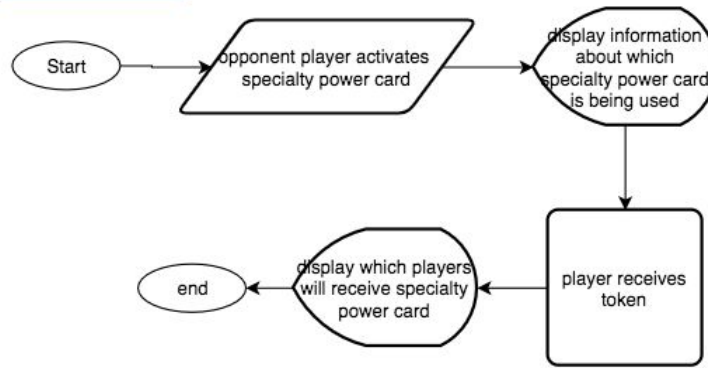
- Actor: user and system
- Pre-condition: player must have purchased a power card with the smoke specialty
- Post-condition: smoke cloud card assigned to player
- Events:
 1. Player gets smoke cloud power card
 - a. Check (6)
 2. Display popup that a smoke cloud specialty card is now in use
 3. Display information button to describe poison specialty and the effects of it to the player
 4. Distribute poison tokens to other players
 - a. Check (7)
 5. Display message to user that smoke tokens have been distributed
- Extensions:
 6. Occurrence -- use power card
 7. Occurrence -- player receives token



35 Use case: receive token

- Actor: user and system
- Pre-condition: specialty power card is used
- Post-condition: player is affected by specialty power card and receives a token
- Events:
 1. Opponent player activates specialty power card
 2. System displays which specialty power card is being used
 - a. Can be one of 4 specialty power cards
 3. Display pop up stating which type of token and how many tokens player will receive
 4. Display player names on who will receive the tokens

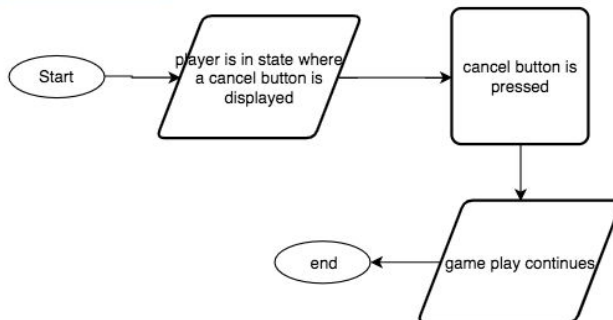
Use case: 35



36 Use case: cancel button pressed

- Actor: user and system
- Pre-condition: for any pop up or dialog containing a cancel button that will allow a player to cancel an action
- Post-condition: user action is closed
- Events:
 1. Popup or dialog appears displaying information that involves user action
 2. User has option to press the cancel button
 3. User presses the cancel button
 4. Dialog containing the information closes
 5. Player is back to previous screen
 6. Game play is continued

Use case: 36



37 Use case: pause menu shown

- Actor: user and system
- Pre-condition: any moment of time that the game is being played, no one has won the game
- Post-condition: game menu is opened
- Events:
 1. Player presses the key bind "esc" to open the pause menu
 2. Menus appears on screen
 - a. Check (5)
 - b. Check (6)

c. Check (7)

3. Player selects option

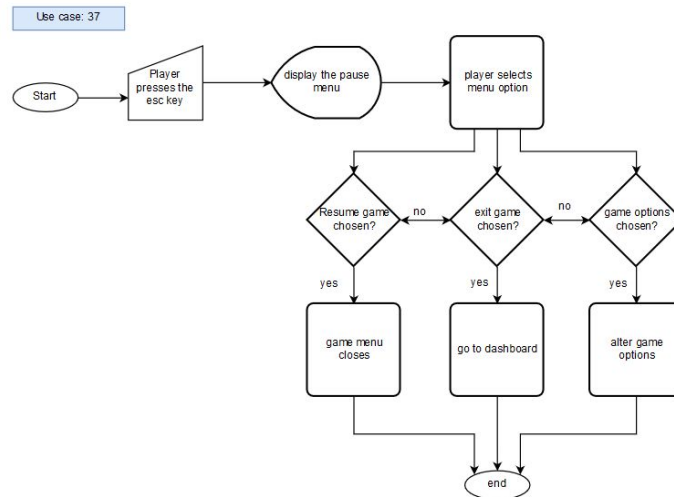
4. Menu closes

- Extensions:

5. Optional -- resume game

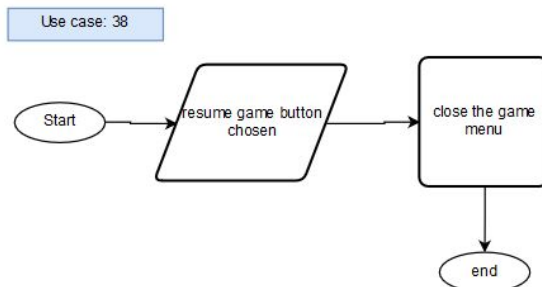
6. Optional -- exit game

7. Optional -- game options



38 Use case: resume game pressed

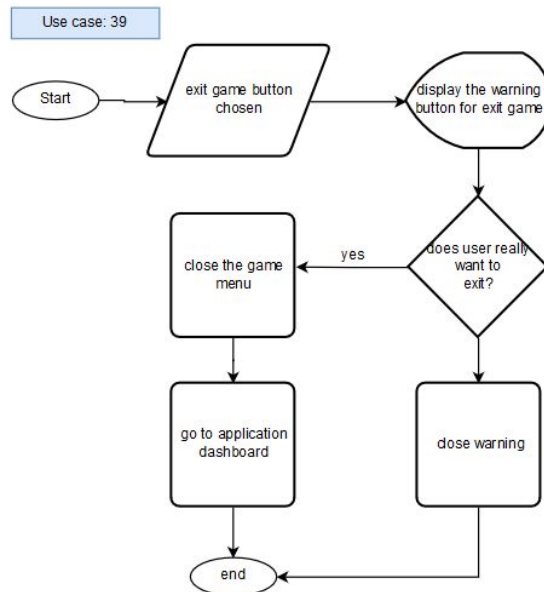
- Actor: user and system
- Pre-condition: pause menu is open
- Post-condition: game is resumed
- Events:
 1. Resume game button is pressed
 2. Pause menu is closed
 3. Game resumes to state
- Extensions:
 4. Occurrence -- pause menu shown



39 Use case: exit game pressed

- Actor: user and system
- Pre-condition: pause menu is open
- Post-condition: the game is closed/exited

- Events:
 1. Exit game button is pressed
 2. The menu is closed
 - a. Check (5)
 3. System exits the game
 4. System is back on dashboard
- Extensions:
 5. Occurrence -- player draw (no one wins or loses)



40 Use case: game options pressed

- Actor: user and system
- Pre-condition: pause menu is opened
- Post-condition: game options have or have not been altered
- Events:
 1. Game options button is pressed
 2. Display a new menu to show which player is changing their game options
 3. Game option is chosen
 4. Game options chosen are initiated
 5. Game menu is closed
- Extensions:
 6. Optional -- change keybinds

Use case: 40

