

**Use Case:** startProgram

**Pre-Condition:** Program isn't running

**Post-Condition:** Program is running

**Events:**

1. Double-click the game icon.

**Use Case:** exitProgram

**Pre-Condition:** Program is running

**Post-Condition:** Program isn't running

**Events:**

1. Press the red 'exit' button in the top right corner. (This can be done at **any** point during the program's runtime).

**Use Case:** selectCharacter

**Actor:** Player

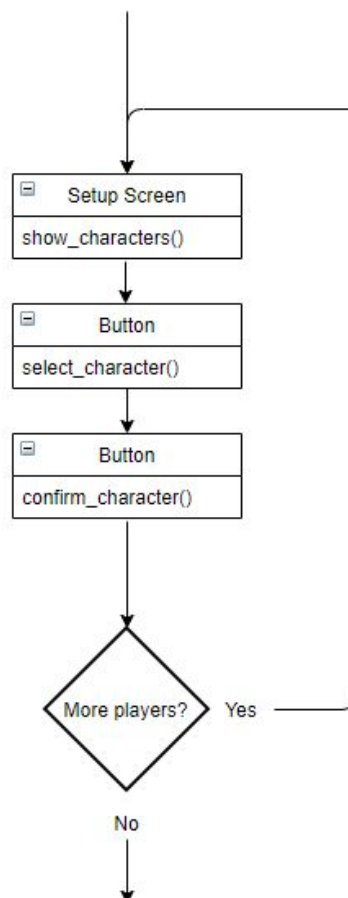
**Pre-Condition:** Game is running and in initial setup screen

**Post-Condition:** Player has selected their character

**Events:**

1. List of Characters show on screen.
2. User selects their character
3. User confirms their choice
4. Next Player selects their character (if last player to do so, continue on to next use case.)

**Uml:**



**Use Case:** rollDiceforOrder

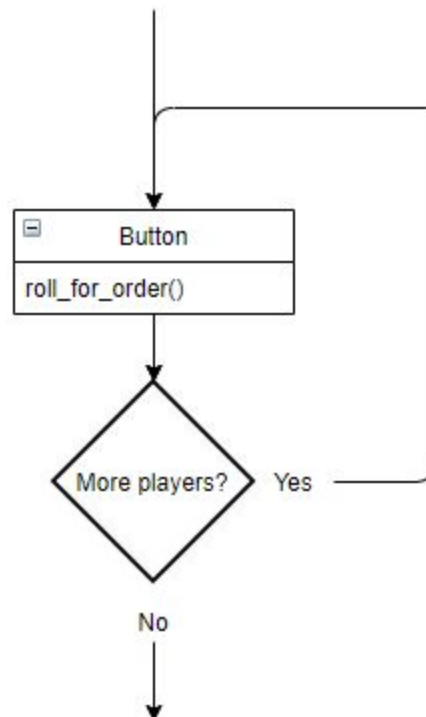
**Pre-Condition:** Players have selected their character

**Post-Condition:** Players have been given their roll order

**Events:**

1. User Presses Roll-Button
2. User receives their roll order. Order is saved in system memory.
3. Next User (if no next user, move to next use-case)

**Uml:**



**Use Case:** selectUsername

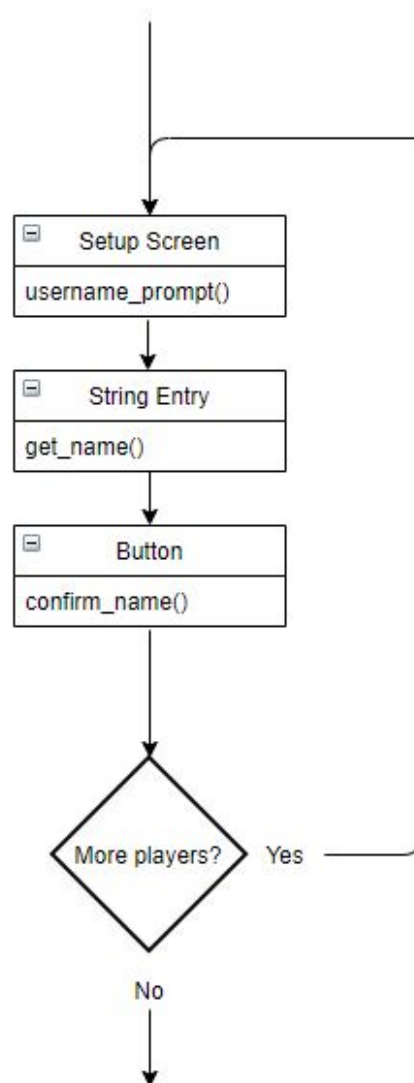
**Pre-Condition:** Game is running and in initial setup screen

**Post-Condition:** All players have their in-game username

**Events:**

1. Player is prompted to enter their name.
2. Player types in their name.
3. Player confirms name
4. Next player enters their name (if no next player, move to next use-case)

**Uml:**



**Use Case:** leaveTokyo

**Pre-Condition:** Player monster is in Tokyo and has just taken damage from an attack

**Post-Condition:** Player monster yields Tokyo to the player monster that attacked him/her.

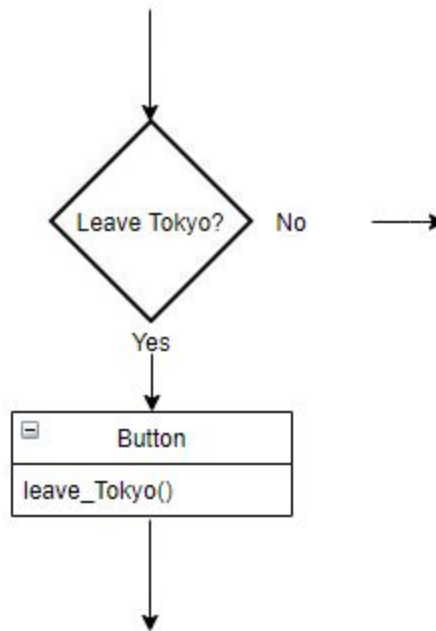
**Events:**

1. Player is attacked.
2. Player chooses to yield to attacker.
3. Attacker is now in Tokyo. Player is outside Tokyo.

**Extensions:**

2. Player can choose to remain in Tokyo and suffer further damage.
3. Player remains in Tokyo. Attacker remains outside Tokyo.

**Uml:**



**Use Case:** spendEnergyToken

**Pre-Condition:** A player has energy tokens in their possession and wants to purchase a power card/ use card effects.

**Post-Condition:** Energy Tokens have been taken from the player and the accompanying power card has been added to their inventory/ card effect takes effect.

**Events:**

1. Player presses button to bring up powercard menu.
2. Player selects the card they want to purchase/use card effect with.
3. The card is added to their inventory/ card effect takes effect. The tokens are subtracted.

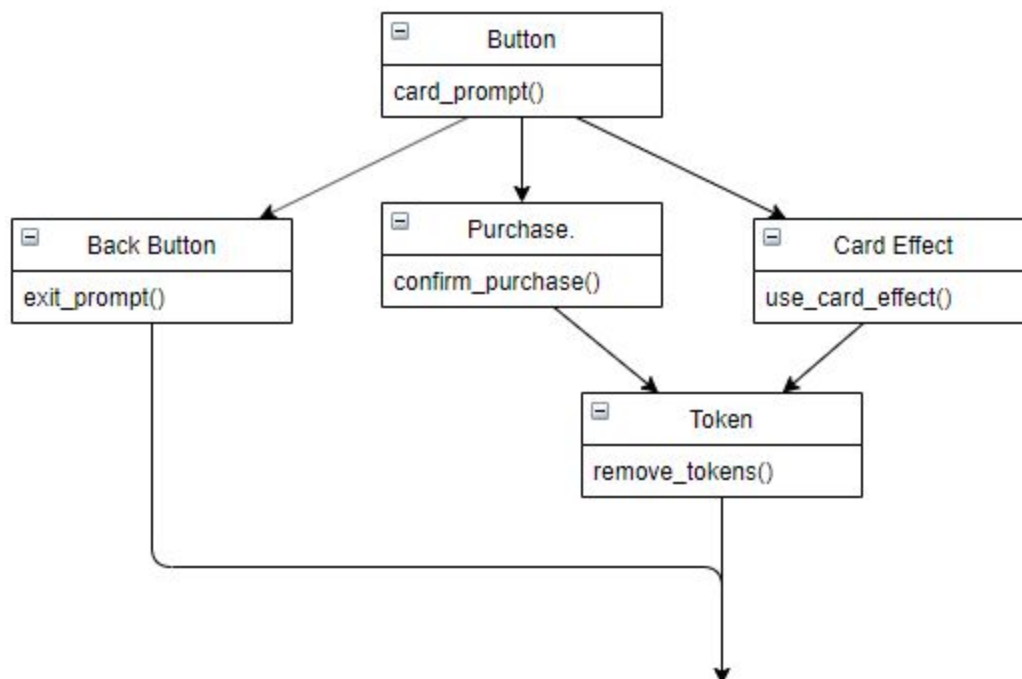
**Extensions:**

1a-3a: User presses 'back' in the purchase prompt

2a.1: Prompt is removed

2a.2: Rejoin step 1

**Uml:**



**Use Case:** Role dice

**Pre-Condition:** It is a player's turn.

**Post-Condition:** The player has rolled their dice 3 times. (Unless they decide to stop earlier than 3.)

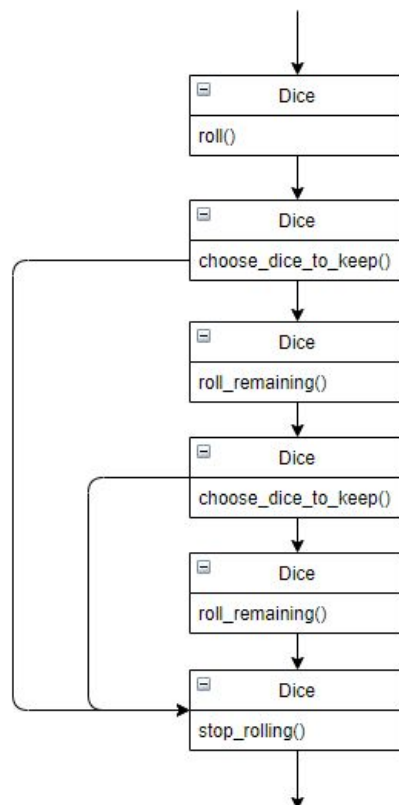
**Events:**

1. Player rolls dice.
2. They select which dice to keep.
3. The player rolls the remaining dice.
4. Again, they select which dice to keep.
5. The player rolls the remaining dice.

**Extensions:**

2. If the player wishes to keep all their dice, they do not have to reroll.
4. If the player wishes to keep all their dice, they do not have to reroll.

**Uml:**



**Use Case:** rollGreen

**Pre-Condition:** Player has a card active that allows the roll of the green dice.

**Post-Condition:** The green dice have been rolled.

**Events:**

1. Player plays card the lets player roll green dice

**Use Case:** changeDice



**Use Case:** mimic card**Pre-Condition:**

1. Player has the card
2. Is on player's turn and is played on "end of turn"
3. There are other cards that other players has to mimic
4. Has to have 1 energy in order to change the power if it is already mimicked.

**Post-Condition:**

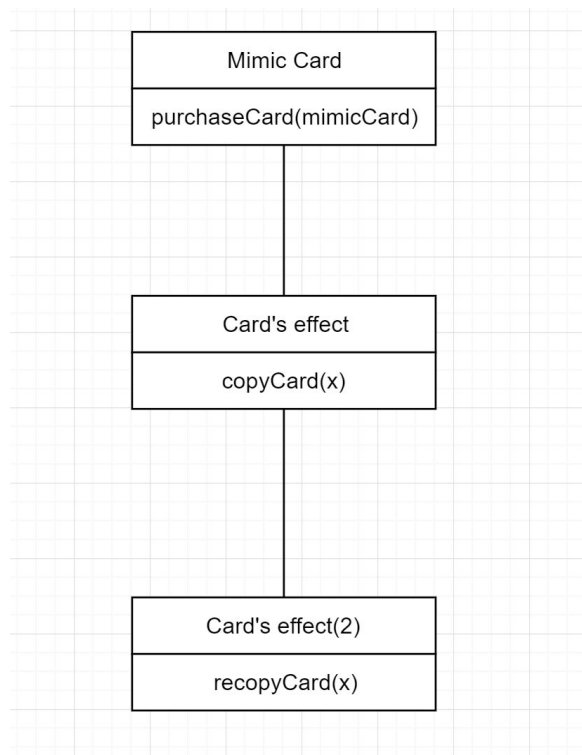
1. Mimics a card in the field
2. Mimics a new card if already mimicked originally.

**Events:**

1. User buys micic card with 8 energy.
  - a. System checks if there are enough energy for the user to buy the card
2. User chooses which card to mimic
  - a. System checks if there are any cards in the first place
3. User chooses a new card to mimic
  - a. System checks if user has enough energy to change the card
4. User ends turn or chooses or another card to use

**Extensions:**

1. (3) User decides not to choose a new card to mimic.
  - a. Go to step 4

**Uml:**

## Use Case: monsterBatteries

### Pre-Condition:

1. Player has the card
2. Player has enough energy to put on reverse

**Post-Condition:**

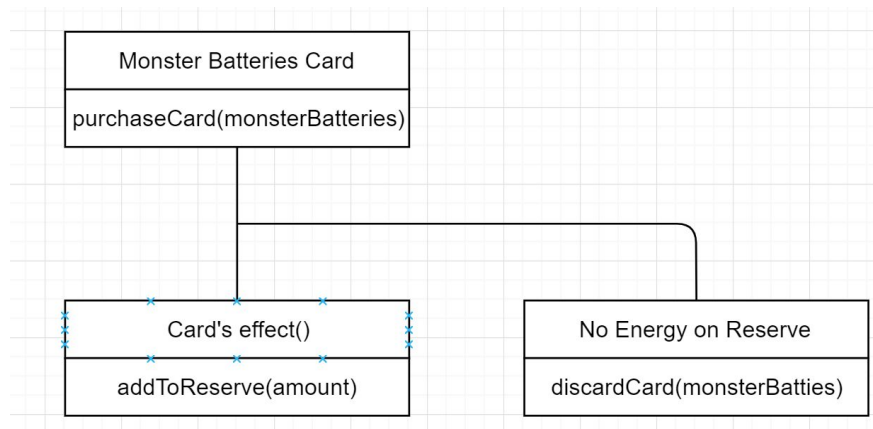
1. Take two energy off and add to reverse at start of turn
2. If no energy on the card, discard card

### Events:

1. Player buys the card
  - a. System checks if the player has enough energy to buy the card
2. Player places 0 - X energy on the card
  - a. System keeps track of the amount of energy on the card
  - b. System matches the amount of energy given
3. When player turn, take two energy off and add to reserve

## Extensions:

## Uml:



**Use Case: opportunistCard****Pre-Condition:**

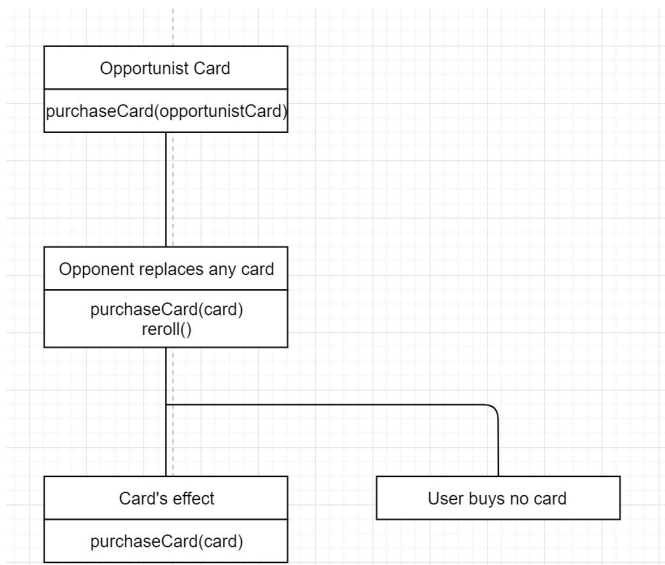
1. Player has the card
2. On other players turn
  - a. Player buys a card and a new card is revealed
  - b. Player rerolls all three cards

**Post-Condition:**

1. Player has the choice to buy the card as soon as it's revealed

**Events:**

1. Player buys the card
  - a. System checks if the user has enough energy to buy the card with
2. Other player buys a card / rerolls and cards are replaced
  - a. System allows the player with opportunist card to buy first

**Extensions:****Uml:**

**Use Case:** purchaseCard

**Pre-Condition:**

1. Player's turn to buy card

**Post-Condition:**

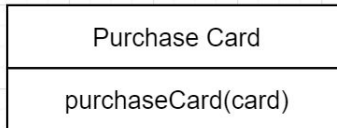
1. Purchases card

**Events:**

1. Player purchases card

**Extensions:**

**Uml:**



**Use Case:** plotTwistCard**Pre-Condition:**

1. User has the card
2. User rerolls

**Post-Condition:**

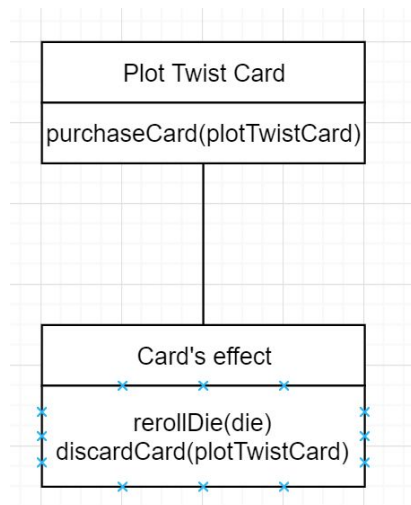
1. User chooses a die that has been rolled and changes the result

**Events:**

1. User buys the plot twist card
  - a. System checks if the user has enough energy to buy the card
2. User rolls on the start of the turn
  - a. System rolls the dice and reveals what is rolled
3. User chooses a dice to reroll
  - a. System takes the user input and chances the die rolled
  - b. System discards the user plot twist card
4. User goes to next step
  - a. System moves the user's state to resolve

**Extensions:**

1. (3) User decides to not reroll any card and keeps the card
  - a. Go to step 4.

**Uml:**

**Use Case:** psychicCard**Pre-Condition:**

1. User has the card
2. On another user's turn and they rolled their dice

**Post-Condition:**

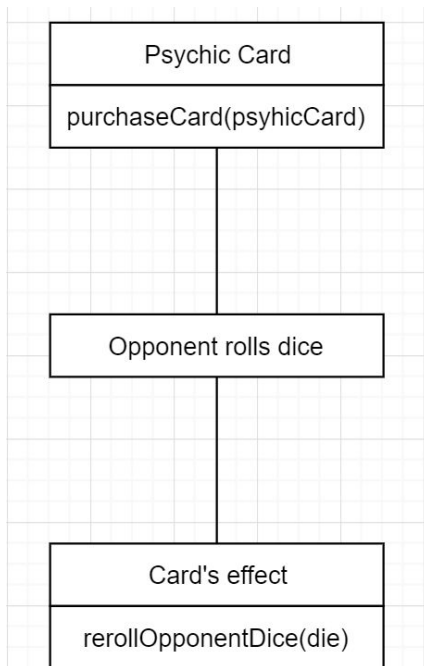
1. User that has the psychic card changes the roll of a die

**Events:**

1. User buys the card
  - a. System checks if the user has enough energy
2. Opponent rolls their dice
  - a. System gives the option to user to they would like to change the die of opponent
3. Opponent goes to their resolve phase

**Extensions:**

1. (2) Opponent rolls a heart
  - a. System discards the card of the user
  - b. Go to step 3

**Uml:**

**Use Case:** stretchyCard**Pre-Condition:**

1. User has the card
2. Player's turn to roll the dice
3. Enough energy to use the card's effect

**Post-Condition:**

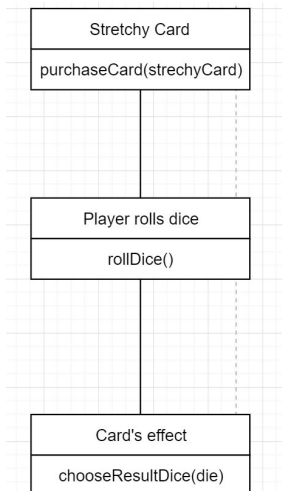
1. Change one of the die to any result

**Events:**

1. User has enough energy to buy the card
  - a. System checks if the user has enough energy to buy the card
2. Player's start of the turn to roll the dice
  - a. System rolls for the user and reveals the result
3. User decides to reroll one of their results
  - a. System checks if user has enough energy then changes accordingly to user's request
4. User go to next phase

**Extensions:**

1. (3) User decides to not use the card's effect
  - a. Go to step 4

**Uml:**

**Use Case: smokeCard****Pre-Condition:**

1. User has the card
2. User's turn to roll their dice
3. User still has charges left

**Post-Condition:**

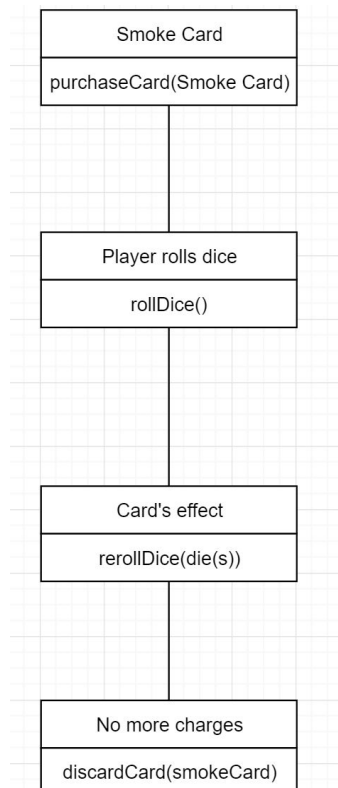
1. User rerolls their dice depending on how much charges they use

**Events:**

1. User buys the card
  - a. System checks if user has enough energy
2. User's turn to roll the dice
  - a. System rolls the dice and displays the result
3. User decides to reroll
  - a. System rerolls the dice
4. User ends phase

**Extensions:**

1. (3) User decides to reroll and runs out of charge. Discard the card
  - a. Go to step 4
2. (3) User decides to not reroll
  - a. Go to step 4

**Uml:**



**Use Case:** superJumpCard**Pre-Condition:**

1. User has the card
2. User is going to take damage

**Post-Condition:**

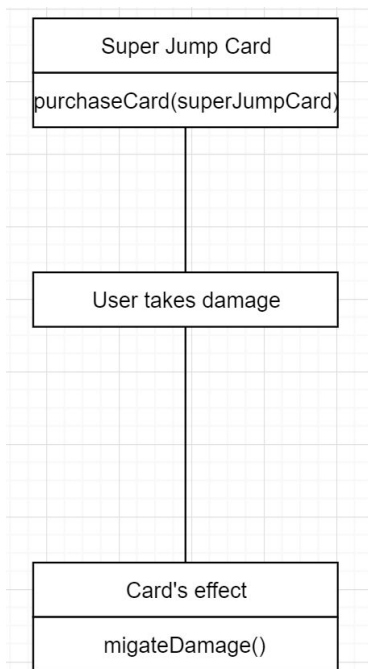
1. User negates 1 damage

**Events:**

1. User buy the card
  - a. System checks if user has enough energy to buy the card
2. User takes damage and negates damage
  - a. System checks if user has enough energy to use the card's effect
3. User negates one damage
4. Opponent resumes turn

**Extensions:**

1. (2) User decides not to negate damage
  - a. Go to step 4

**Uml:**

**Use Case: Frenzy**

**Pre-Condition:** User's has the energy to purchase Frenzy and can buy cards

**Post-Condition:** User gets an extra turn once it is their turn and they finish their current turn

**Events:**

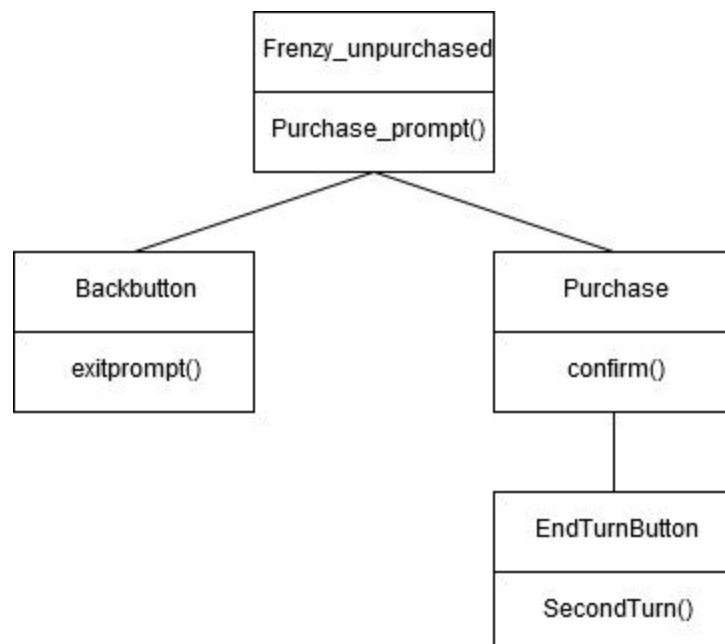
1. The user attempts to purchase the Frenzy card
  - a. The system confirms if the user wants to purchase the card and checks if they have enough energy. If they don't, the "yes" button is greyed out
2. The user selects "yes"
  - b. The system discards the Frenzy card into the discard pile and deducts the energy from their total
  - c. When it's the User's turn their the system transforms the "end turn" button into a "second turn"
3. When the user presses the button they have a second turn after their turn

**Extensions:**

1a-3a: User presses 'back' in the purchase prompt

2a.1: Prompt is removed

2a.2: Rejoin step 1

**UML:**

**Extensions:****Use Case:** Background Dweller**Actor:** User**Pre-Condition:** User's turn and is currently selecting what to do with their dice card  
Background Dweller**Post-Condition:** User selected 3's dice are rerolled**Events:**

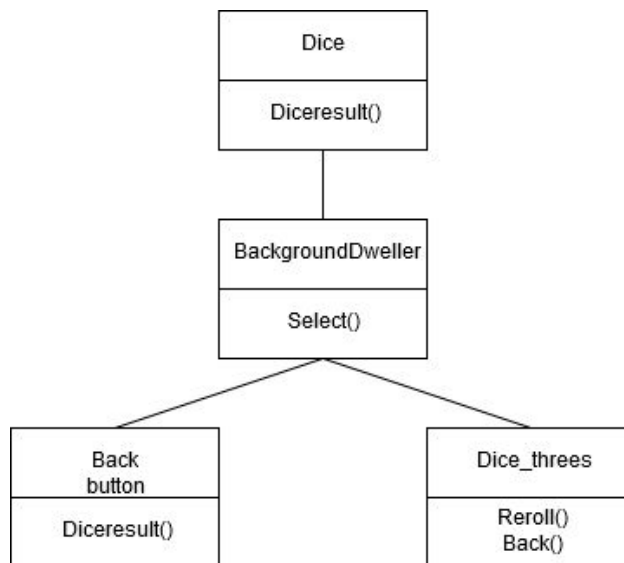
1. User just finished rolling their die and selects the Background Dweller Card
  - a. The system first checks if there are any 3's if there is none the card does nothing
  - b. Otherwise it prompts the user which 3's to reroll
2. The user selects which 3's to reroll
  - c. The dice are rerolled and the card is greyed out to signify use

**Extensions:**

1a-2a: User presses 'back'

2a.1: Prompt is removed

2a.2: Rejoin step 1

**UML:**

**Use Case:** Counter Remove**Actor:** User**Pre-Condition:** User monster has a negative counter (spit or shrink) on them and is rolling their dice and is not in Tokyo**Post-Condition:** The designated counter is removed from their monster**Events:**

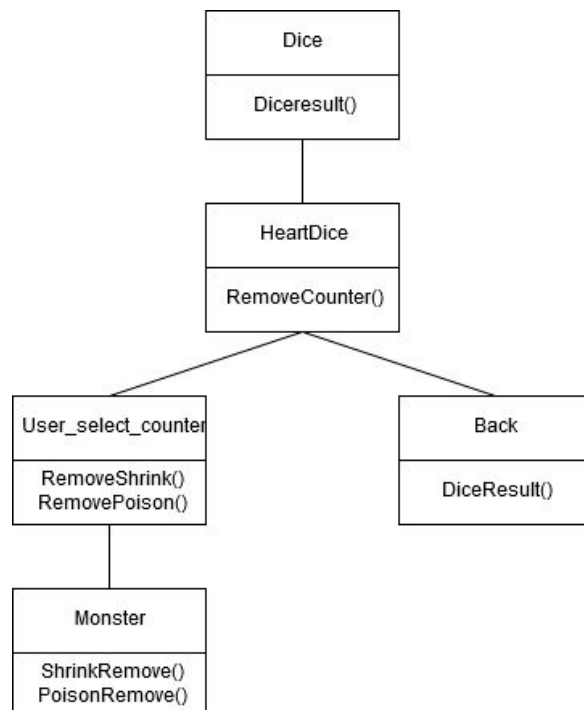
1. User is not in Tokyo and is in selecting what to do with their dice
  - a. System prompts the user on what they want to do with their dice
2. User selects remove tokens on one of their heart dices
  - b. System prompts the user on which counter to remove (when applicable)
3. User selects the counter
  - c. System removes the counter off their monster
4. User selects what to do with the rest of their dice

**Extensions:**

2a-6a: User presses 'back' during the dice select screen

2a.1: Prompt is removed

2a.2: Rejoin step 2

**UML:**

**Use Case:** Start Again

**Actor:** User

**Pre-Condition:** User won the game

**Post-Condition:** Game is replayed

**Events:**

1. User wins the game
  - a. System prompts the user with a win screen that contains a replay button and an exit button
2. User presses replay button on the win screen
  - b. System takes user to setup screen for a new game.

**Use Case:** Exit after victory

**Actor:** User

**Pre-Condition:** User won the game

**Post-Condition:** Game is exited

**Events**

1. User wins the game
  - a. System prompts the user with a win screen that contains a replay button and an exit button
2. User presses the exit button
  - b. System terminates the program and takes them to the desktop

**User case:** ResolveOrder

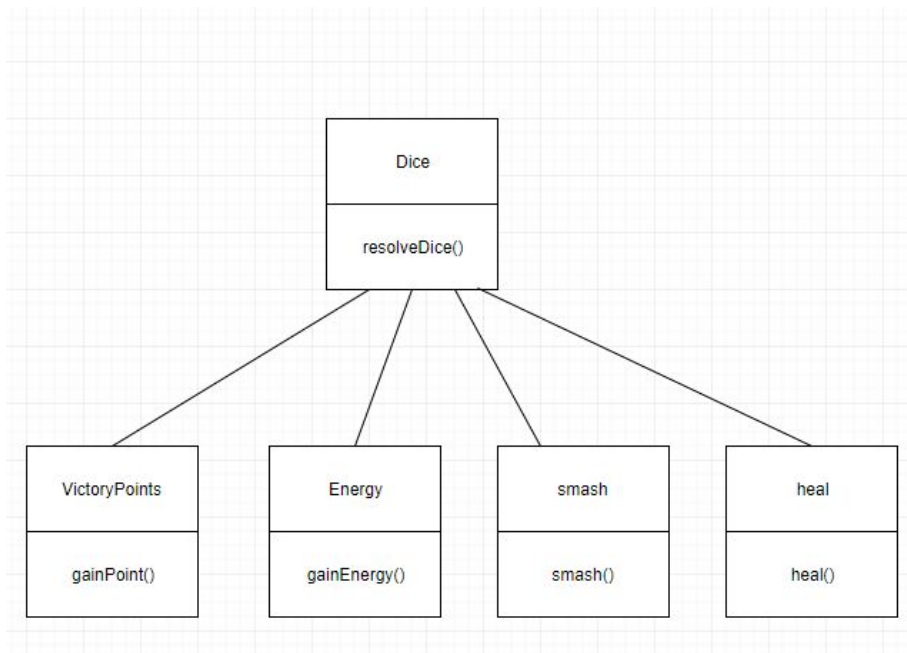
**Actor:** User

**Pre condition:** All dice are finished rolling

**Post condition:** User has exhausted all dice abilities

**Events:**

1. User finishes rolling dice
  - a. System asks the user what action they would like to perform first with the dice
2. User decides what order they want to use their dice abilities in.
  - a. User must keep using abilities until they run out of dice.



**User case:** SmashOutside

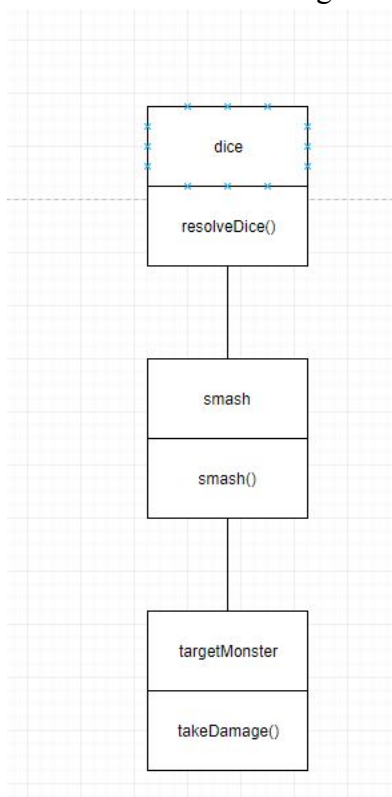
**Actor:** User

**Pre Condition:** User rolls a smash card

**Post Condition:** Target Monster takes damage

**Event:**

1. User rolls a smash action on their dice
  - a. They decide to use it and monsters take damage
2. Monsters that take damage inside tokyo can yield and leave tokyo





**User Case:** buyCard

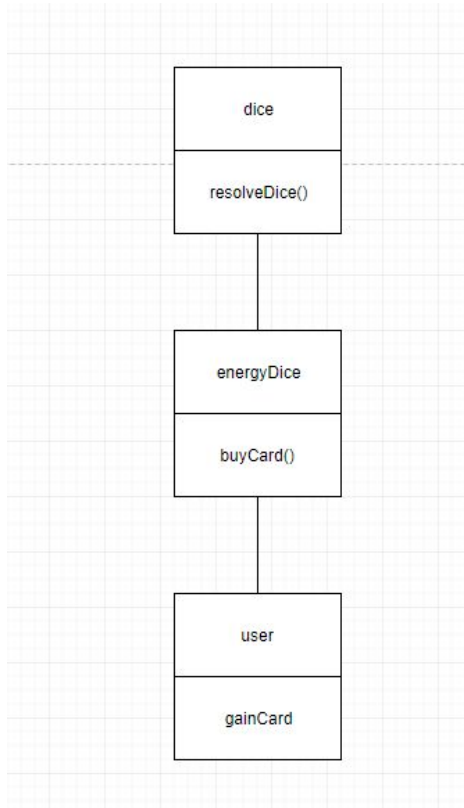
**Actor:** User

**Pre condition:** User has enough energy to buy a card

**Post condition:** User gains a card

**Event:**

1. User resolves dice after rolling and has enough energy to buy a card
  - a. User spends and buys an energy card
2. User gains a card



**User case:** SweepFaceupCard

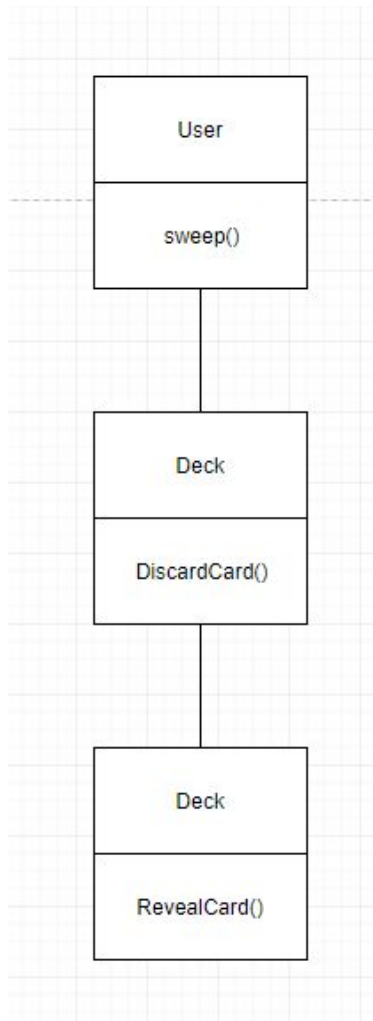
**Actor:** User

**Pre Condition:** User must have enough energy to use this action

**Post Condition:** Power cards currently available are discarded and new ones are available

**Event:**

1. User has enough energy to use sweep
  - a. Top 3 cards are discarded
2. 3 new cards are available and are ready to be purchased.



**User case:** CamoCard

**Actor:** User

**Pre Condition:** User takes damage and has this card in order to use it

**Post Condition:** User has a new dice roll to resolve and takes damage

**Event:**

1. User takes damage
  - a. Must roll a dice for each damage point
2. User does not take damage if they roll a heart
  - a. User has a new dice set to resolve

**User Case:** healMonster

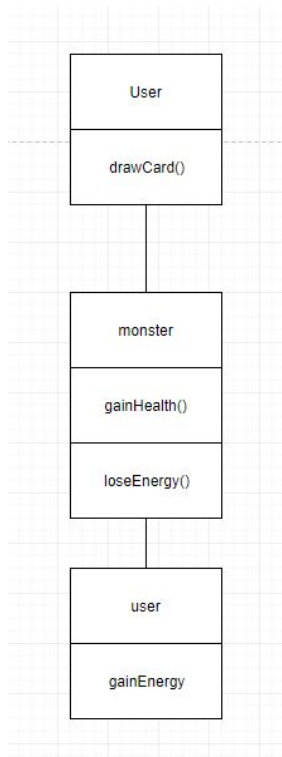
**Actor:** User

**Pre Condition:** User draws a card that allows them to heal a monster of their choice

**Post Condition:** Monster that is healed must pay two energy points

**Events:**

1. User draws a card that allows them to heal a monster
  - a. User chooses a monster
2. Monster pays two energy after healed
  - a. User gains energy



**User Case:** ChangeDice

**Actor:** User

**Pre Condition:** User can change their dice if they don't like the outcome

**Post Condition:** Dice value is changed

**Events:**

1. User rolls dice and they don't like the result
  - a. Can choose to reroll for a better outcome

**User Case:** peek

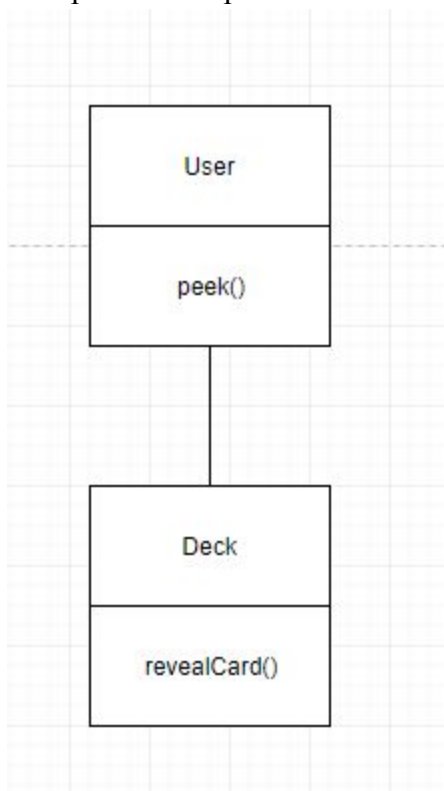
**Actor:** user

**Pre condition:** User has a card that allows them to use peek

**Post condition:** User knows what card is at the top of the deck

**Events:**

1. User peeks the top of the deck and they know what is on top



**User case:** Metamorph

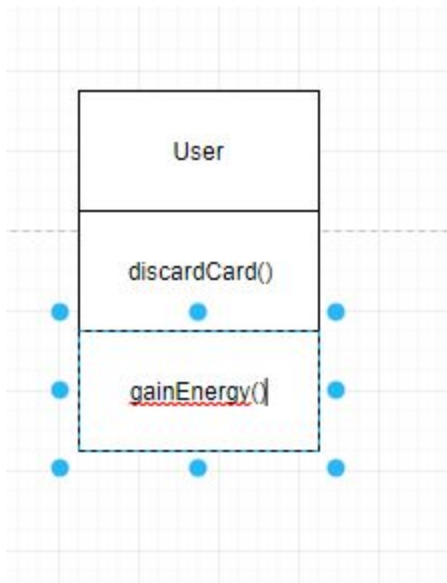
**Actor:** User

**Pre condition:** User must have metamorph card in order to use

**Post Condition:** User loses a card, but gains energy

**Events:**

1. User has metamorph
  - a. Can choose a card in order to discard
2. User gains energy, but loses a card



**User Case:** mimicCard**Actor:** user**Pre condition:** User must have the card**Post condition:** The mimic card changes its ability to the target card, but the user loses one energy point**Events:**

1. User uses mimic and picks an enemy
  - a. Enemy's card ability is copied onto the mimic card
2. User loses an energy point for using the mimic card

