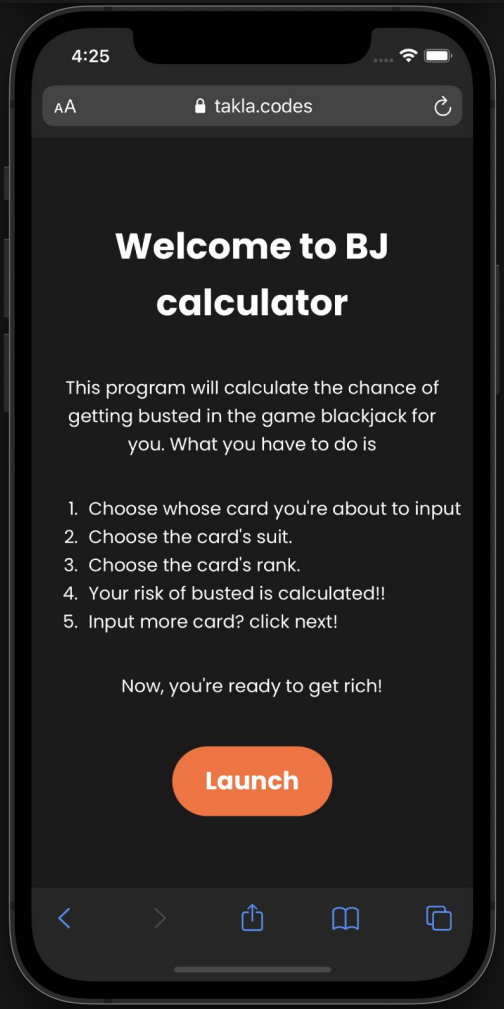
Blackjack calculator

Calculator that make you rich!

Chanon Janesiripanich (6205026) and Takla Nateeboon (6205029)



Blackjack calculator

You can play it here at

URL: bit.ly/BJcalculator

Work best on smartphone

Source code

URL: bit.ly/gitBJcal


$$$
Motivation
$$$

How to play Black Jack

Winning condition: Get total value of cards in hand 21 or closest to 21.
But not exceed 21, otherwise the player is busted!

How to play: Deal a card a time, after each dealt player can decide whether they want to deal more or stop. (House stands on all 17s)

Image courtesy: https://www.onlinecasinogokkennederland.nl/blackjack

Calculating the value of each card

1. 2 through 10 count at face value, i.e. a 2 counts as two, a 9 counts as nine.
2. Face cards (J,Q,K) count as 10.
3. Ace can count as a 1 or an 11 depending on which value helps the hand the most.

Card	Number cards (2,3,4,5,6,7,8,9,10)	J,Q and K	Ace
Value	The face value of the card	10	11 or 1

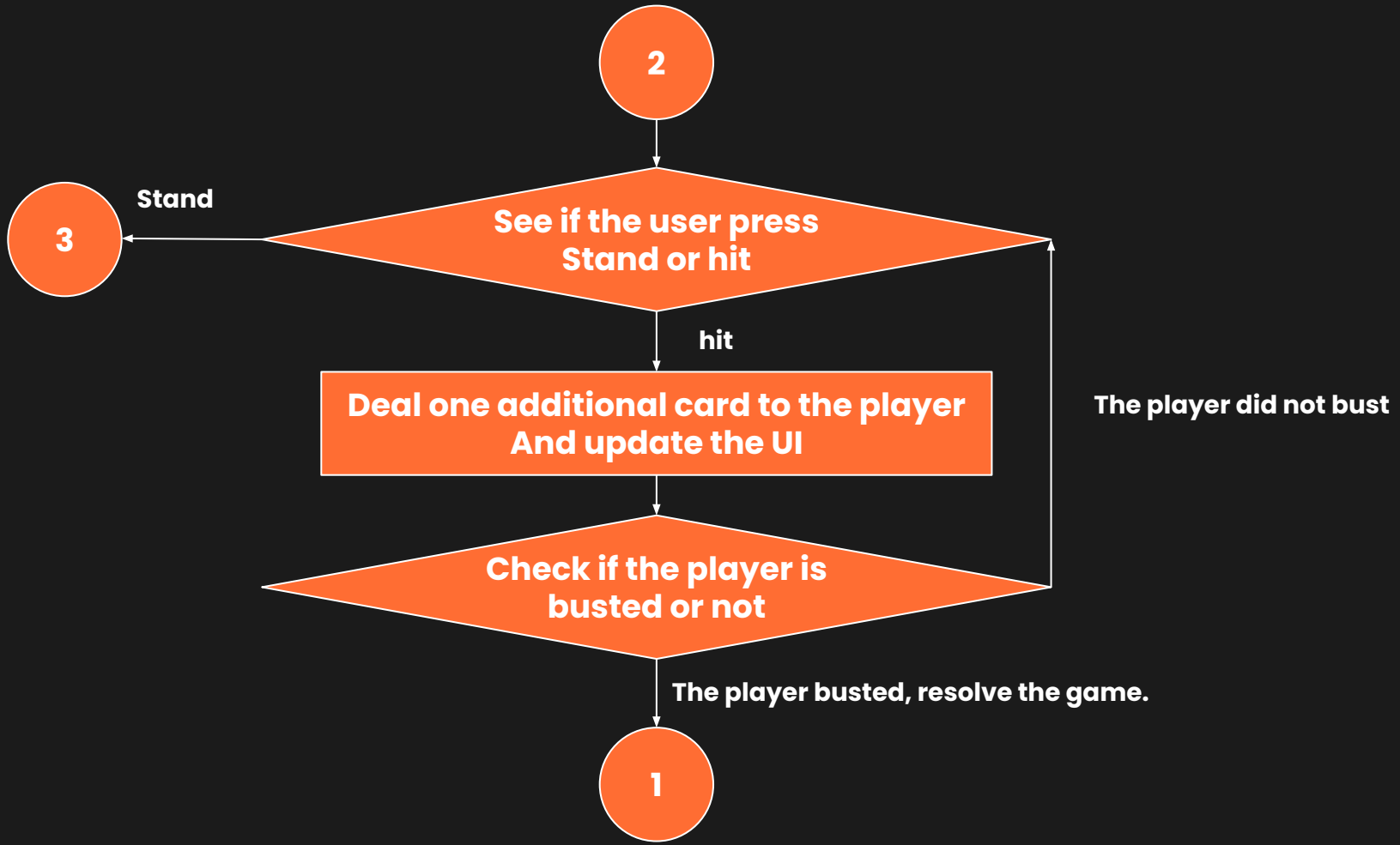
Problem statement

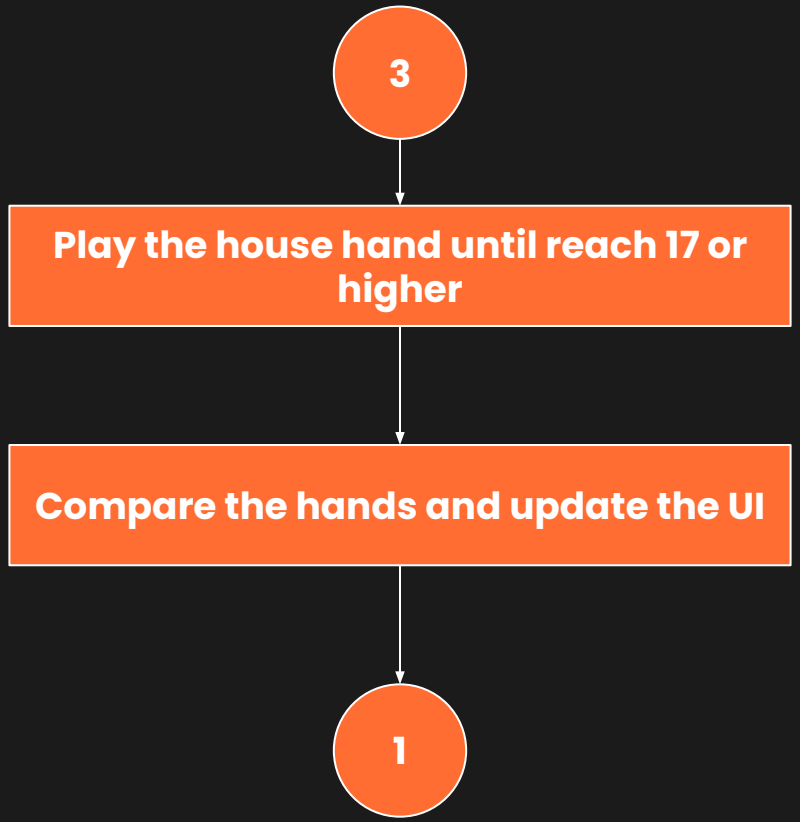
We want something that would help us decide what is the best decision to take, so that we can minimize the cash loss and maximize the profit from playing.

Image courtesy:
http://www.kilwinningarchers.com/how-challenging-is-video-poker/

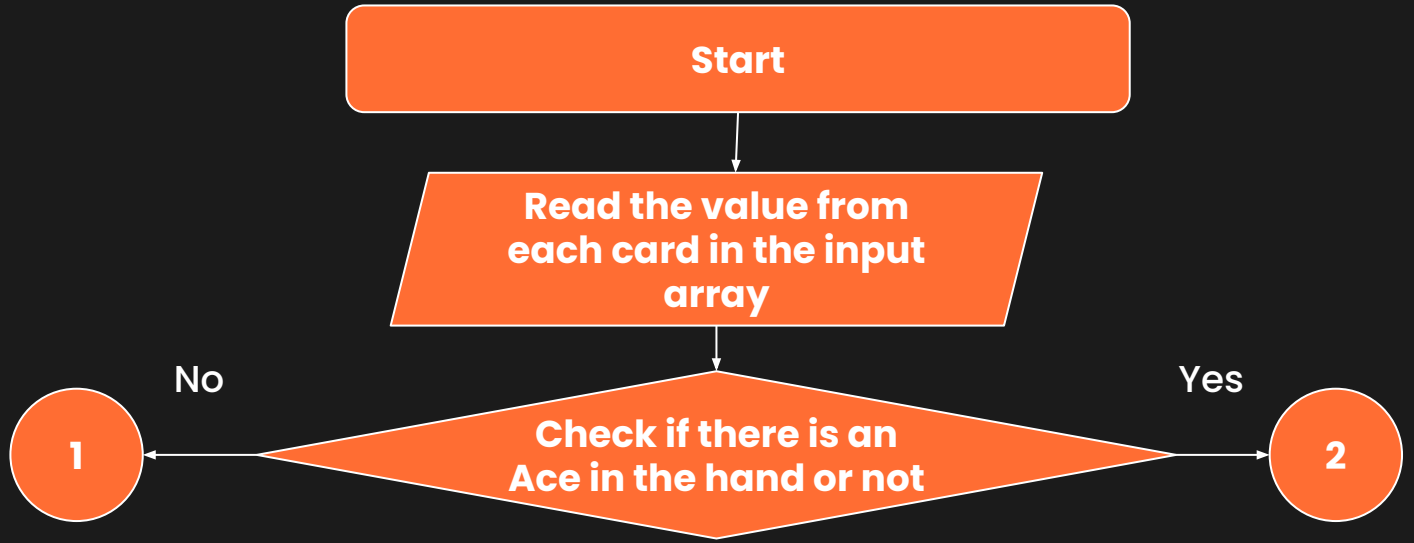
Input-Output Ideas

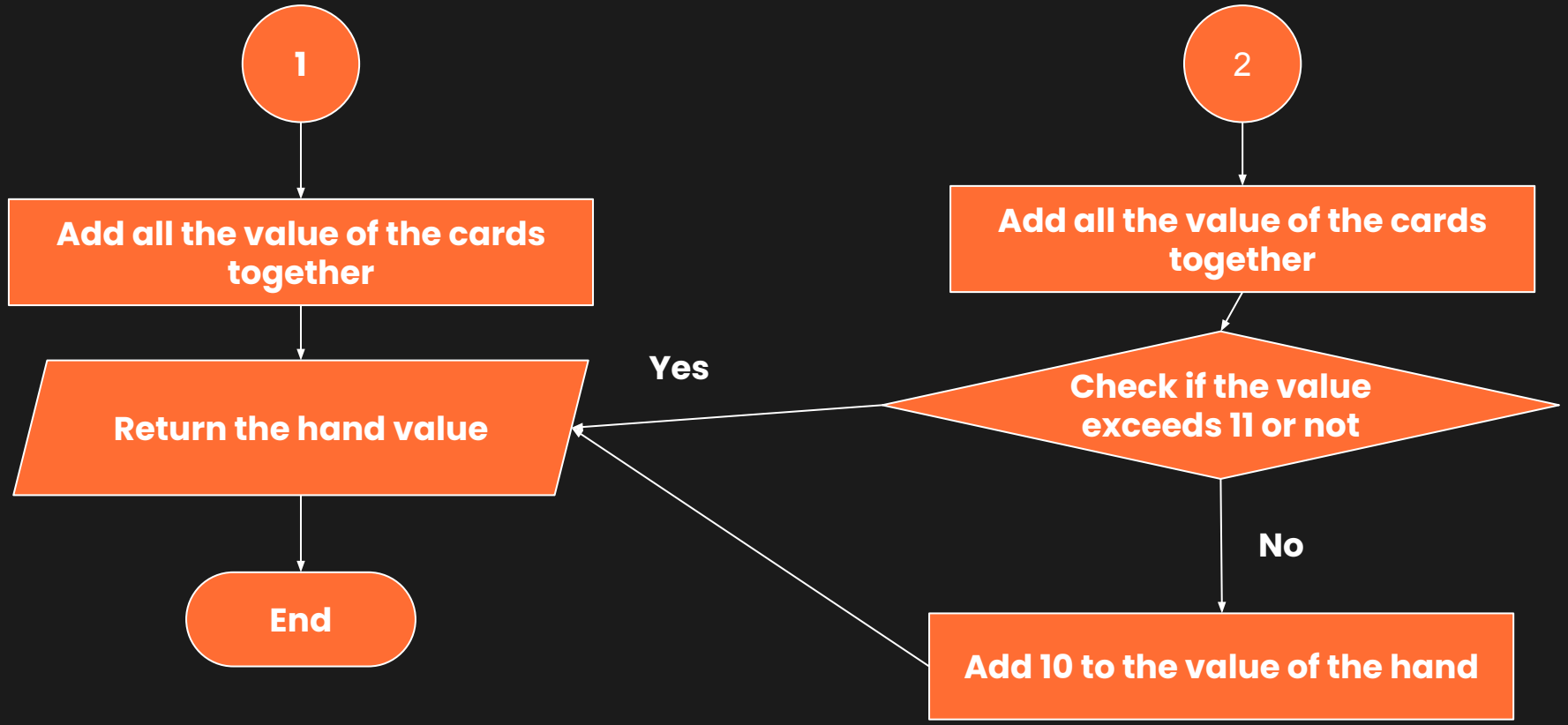
Problem exploration: Basic gameplay


2
See if the user press
Stand or hit
Stand
3
hit
Deal one additional card to the player
And update the UI
The player did not bust
Check if the player is
busted or not
The player busted, resolve the game.
1


3
Play the house hand until reach 17 or higher
Compare the hands and update the UI
1

Problem exploration: getValue();



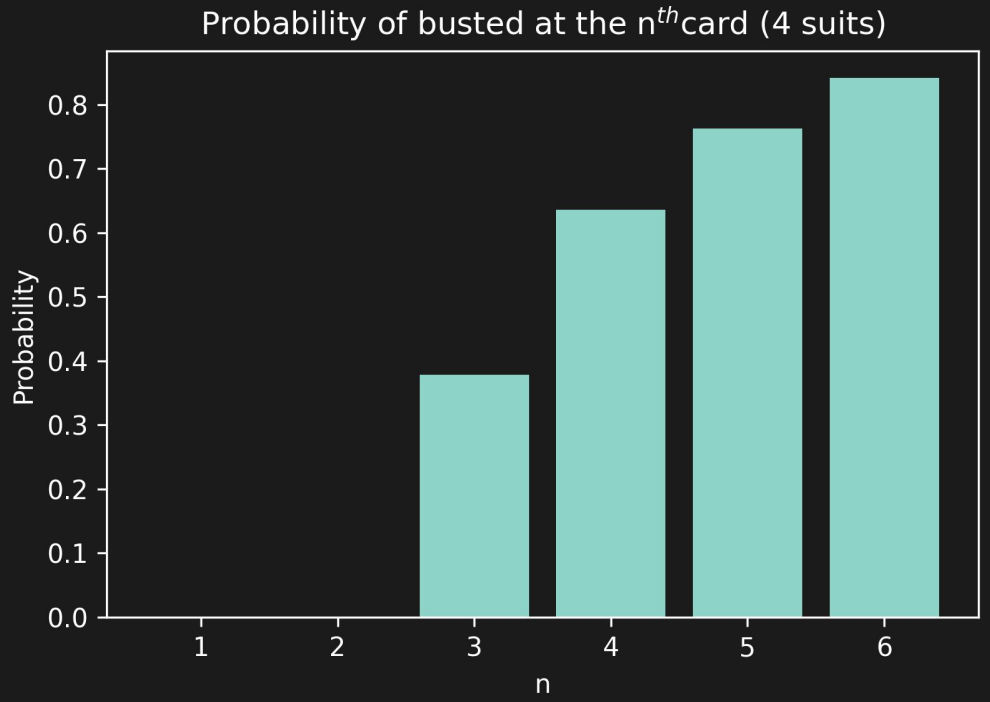

1
Add all the value of the cards together
Return the hand value
End
2
Add all the value of the cards together
Check if the value exceeds 11 or not
Yes
No
Add 10 to the value of the hand

Calculating the Chance of busting after hitting

1. Input the array of all the card known by the player
2. Create a deck of cards that did not contain known cards
3. Determine all possible hand combinations and calculate the odds of busting

Problem exploration#2

Q: How long the game last?



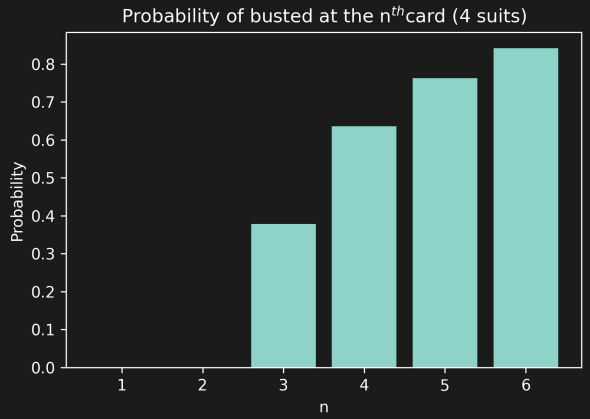
- This is from calculation in python notebook using brute-force method.

proof_of_concept.ipynb

~ 3-4 hits

Q: How long the game last?

Brute force on python



Generate combinations of n cards

Evaluate its score and classified as bust or not bust

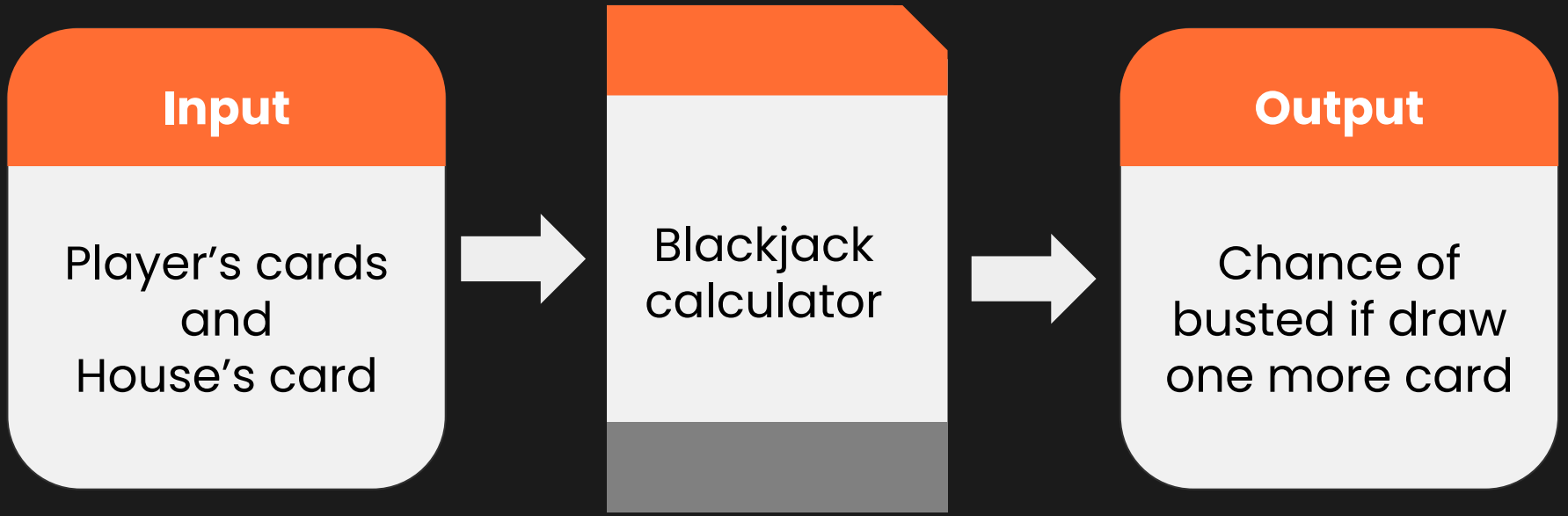
Probability of **bust with *n* cards** = #bust/#combinations

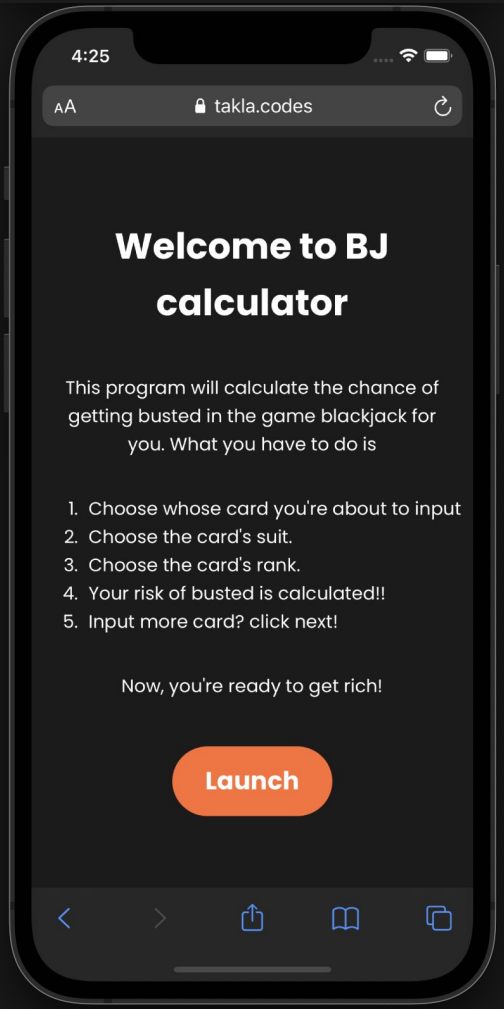
Probability of bust when deal an n^{th} cards

= Probability of (bust with *n* | *n-1* card not busted)

$$= \frac{\text{Probability of (bust with } n\text{) - Probability of (busted with } n\text{-}1\text{)}}{\text{Probability of (not busted with } n\text{-}1\text{)}}$$

The program's design





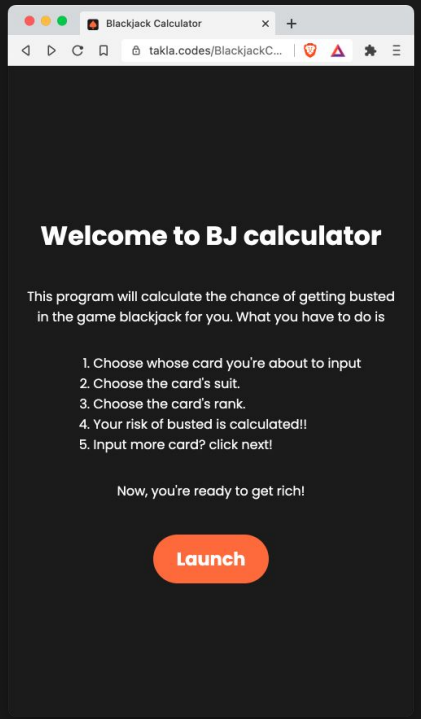
Final product

You can play it here at

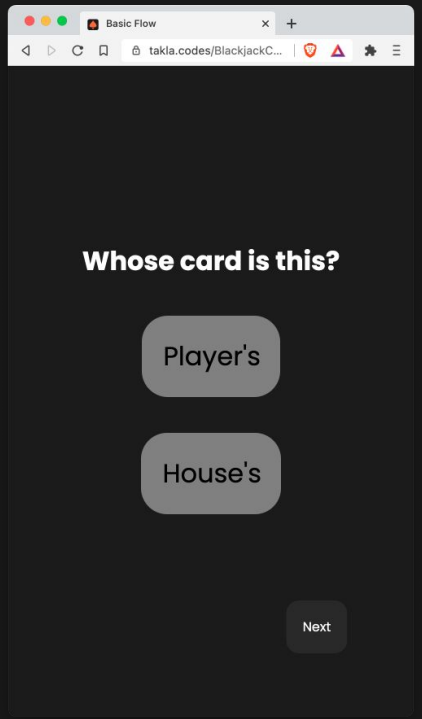
URL: bit.ly/BJcalculator

Work best on smartphone

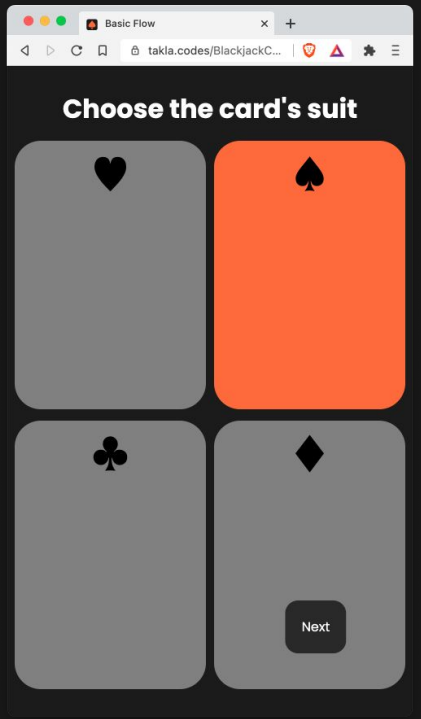
Dynamics environments



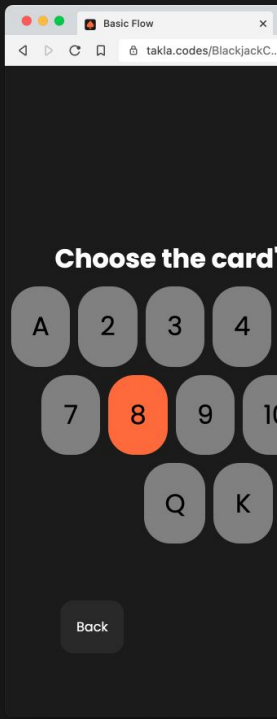
index.html



main.html

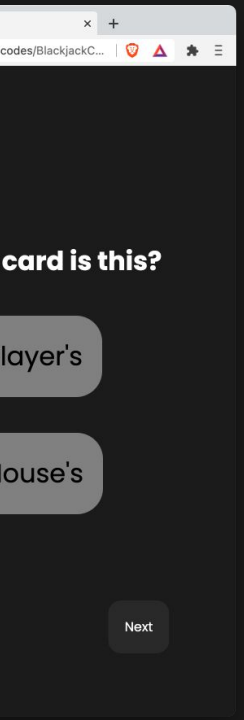


main.html

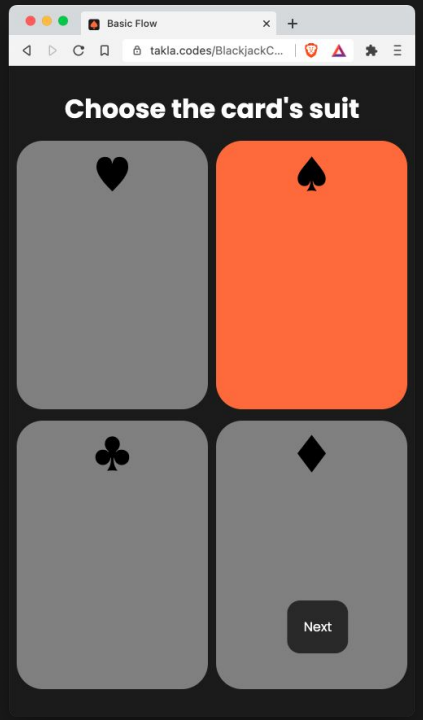


main.htm

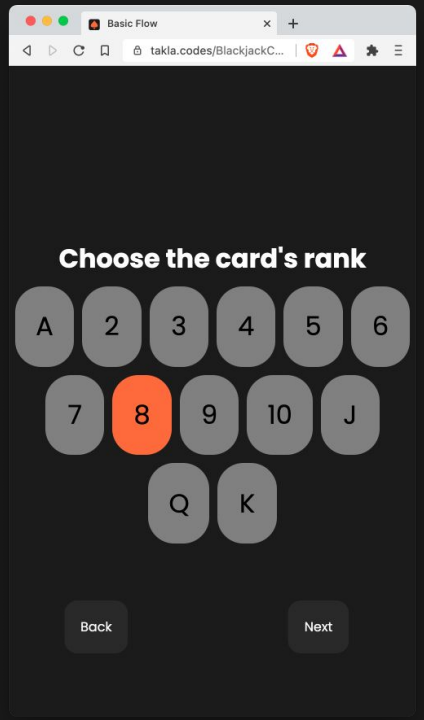
Dynamics environments



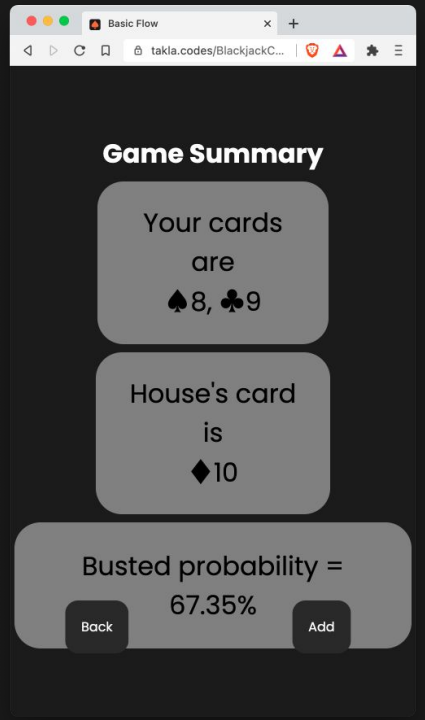
html



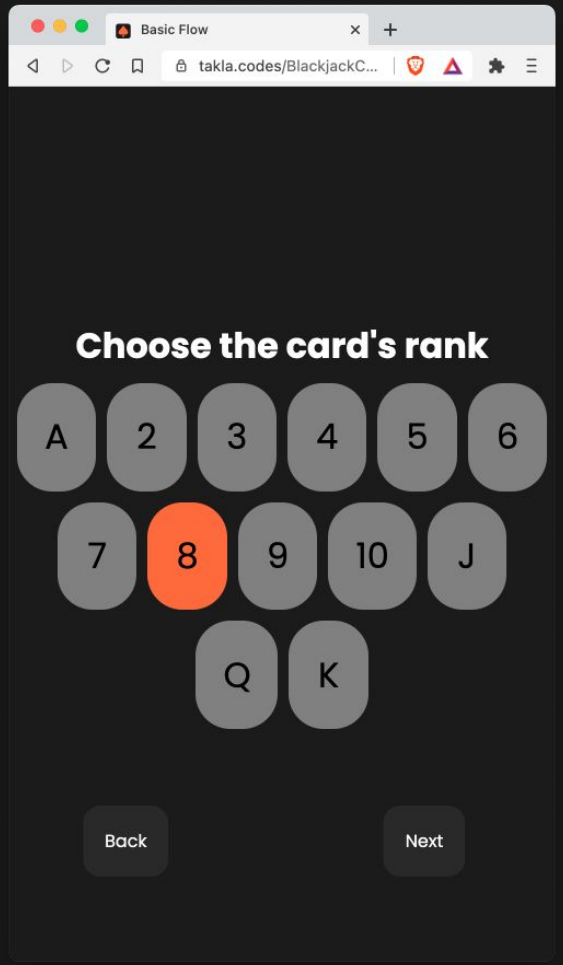
main.html



main.html



main.html



Dynamics environments creation

Define each page as css class

To show apply class “stage”

To hide apply class “stage-hide”

Add/Remove as a user click the button next

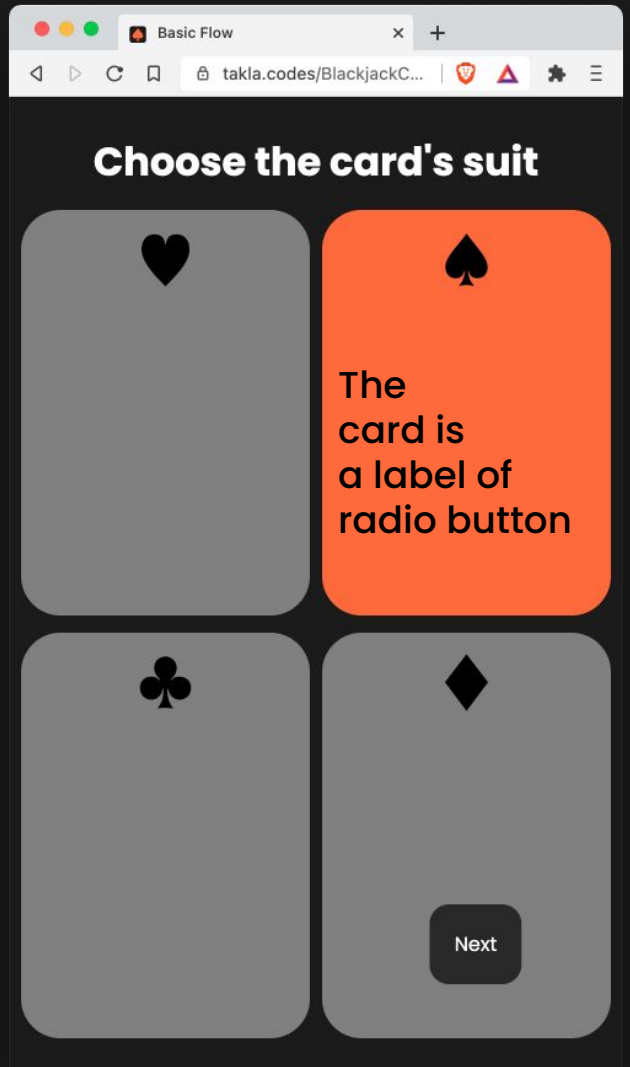
(This control by JS)


```
.stage{
    Flexbox to align content in the center
    Set its size to match with windows.
    (Set width=100vw and height=100vh)
    Add display = hide when click Next
}

.prompt{
    Bold font with white color
}

.card{
    Rounded corner by border-radius: 2em.
    Padding for aesthetic.
}

.next-button{
    Fixed position on screen for ease of use.
}
```



Collect input output

Use JS as usual

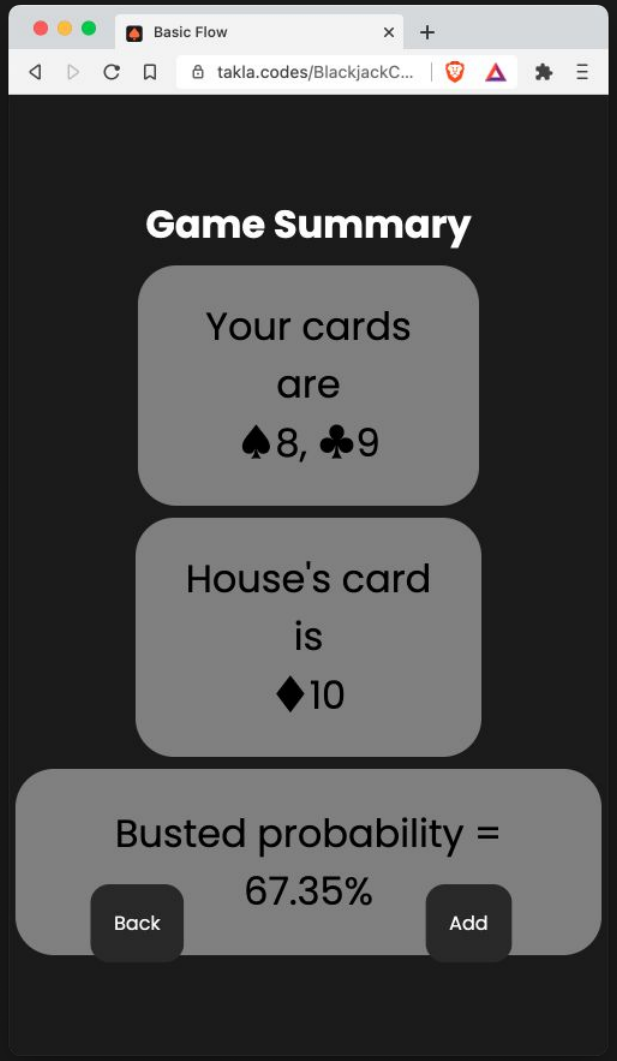
Collect data from radio button

Color change using Pseudo-class “focus” (css)

Hide the radio button by css

Get value from radio button by looping in array.

Collects input when user click “Next”



Summary page

Show user's inputs

Show probability of getting busted if player deal the next card.

Color pallet

Define these colors in css for convenient

Accent color

```
--accent-color:rgb(255, 109, 51);
```

Background color

```
--base-color: #1b1b1b;
```

Text color

```
--text-color:white;
```


Go get rich !

Cards, deck data structure

```
Card = {Value: "K", Suit: "Hearts", Weight: 10} //type = object

Deck = Array of cards
```

CSS and SASS

```
Originally wrote in .SASS file then use VSCode plugin to convert to .css
```

Outline

- Motivation (Chanon)
  - Get rich
- Problem statement
- Input/Output
- Problem exploration
  - Demo (Probabilities calculation) (Chanon)
  - Ipynb proof of concept
- Final product
  - Technicalities
    - UIUX
  - Color pallet
- Application Demo
- End, we all become billionaires.

Points()

Algorithm

Code

Hard

Duration: 10 mins