Name : Sarthi S Darji
Roll No :12

Q 31 >
Project Idea: Build a simple game where players control characters with different abilities, each represented by prototype objects.
   - Description: Participants will define prototype objects for different character types (e.g., warrior, mage) with shared properties and methods. They'll understand how to use prototype objects to create multiple instances with shared behavior and characteristics.

 Html code :

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
 <head>
   <meta charset="utf-8">
   <title>Project 31</title>
   <link rel="stylesheet" href="style.css">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
 </head>
 <body>
   <div class="wrapper">
     <ul class="cards">
       <li class="card">
         <div class="view front-view">
           <img src="images/que_icon.svg" alt="icon">
         </div>
         <div class="view back-view">
           <img src="images/img-1.png" alt="card-img">
         </div>
       </li>
       <li class="card">
         <div class="view front-view">
           <img src="images/que_icon.svg" alt="icon">
         </div>
         <div class="view back-view">
           <img src="images/img-6.png" alt="card-img">
         </div>
       </li>
       <li class="card">
         <div class="view front-view">
```

```html
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-3.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-2.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-1.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-5.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-2.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-6.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-3.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-4.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-5.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-4.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-4.png" alt="card-img">
```

```
         </div>
       </li>
       <li class="card">
         <div class="view front-view">
           <img src="images/que_icon.svg" alt="icon">
         </div>
         <div class="view back-view">
           <img src="images/img-4.png" alt="card-img">
         </div>
       </li>
       <li class="card">
         <div class="view front-view">
           <img src="images/que_icon.svg" alt="icon">
         </div>
         <div class="view back-view">
           <img src="images/img-4.png" alt="card-img">
         </div>
       </li>
       <li class="card">
         <div class="view front-view">
           <img src="images/que_icon.svg" alt="icon">
         </div>
         <div class="view back-view">
           <img src="images/img-4.png" alt="card-img">
         </div>
       </li>
     </ul>
   </div>

   <script src="script.js"></script>

 </body>
</html>
```

Css code :

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swa
p');
```

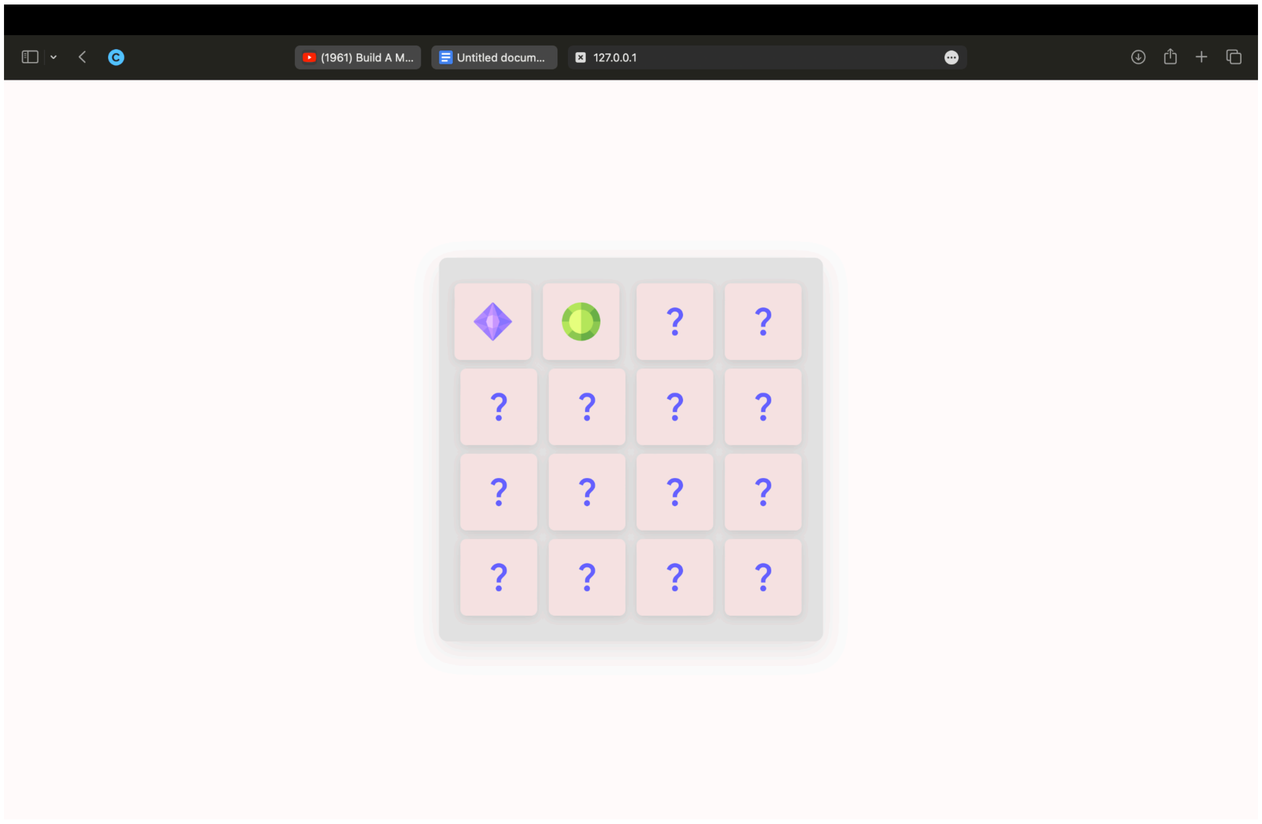
```css
*{
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: 'Poppins', sans-serif;
}
body{
 display: flex;
 align-items: center;
 justify-content: center;
 min-height: 100vh;
 background: #fffcfc;
}
.wrapper{
 padding: 25px;
 border-radius: 10px;
 background: #e4e2e2;
 box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}
.cards, .card, .view{
 display: flex;
 align-items: center;
 justify-content: center;
}
.cards{
 height: 400px;
 width: 400px;
 flex-wrap: wrap;
 justify-content: space-between;
}
.cards .card{
 cursor: pointer;
 list-style: none;
 user-select: none;
 position: relative;
 perspective: 1000px;
 transform-style: preserve-3d;
 height: calc(100% / 4 - 10px);
 width: calc(100% / 4 - 10px);
}
.card.shake{
 animation: shake 0.35s ease-in-out;
```

```css
}
@keyframes shake {
 0%, 100%{
   transform: translateX(0);
 }
 20%{
   transform: translateX(-13px);
 }
 40%{
   transform: translateX(13px);
 }
 60%{
   transform: translateX(-8px);
 }
 80%{
   transform: translateX(8px);
 }
}
.card .view{
 width: 100%;
 height: 100%;
 position: absolute;
 border-radius: 7px;
 background: #f6e5e5;
 pointer-events: none;
 backface-visibility: hidden;
 box-shadow: 0 3px 10px rgba(0,0,0,0.1);
 transition: transform 0.15s linear;
}
.card .front-view img{
 width: 19px;
}
.card .back-view img{
 max-width: 45px;
}
.card .back-view{
 transform: rotateY(-180deg);
}
.card.flip .back-view{
 transform: rotateY(0);
}
.card.flip .front-view{
 transform: rotateY(180deg);
}

@media screen and (max-width: 700px) {
 .cards{
   height: 350px;
   width: 350px;
 }
 .card .front-view img{
   width: 17px;
 }
 .card .back-view img{
   max-width: 40px;
 }
}

@media screen and (max-width: 530px) {
 .cards{
   height: 300px;
   width: 300px;
 }
 .card .front-view img{
   width: 15px;
 }
 .card .back-view img{
   max-width: 35px;
 }
}
```

Js code:

```
const cards = document.querySelectorAll(".card");

let matched = 0;
let cardOne, cardTwo;
let disableDeck = false;

function flipCard({target: clickedCard}) {
   if(cardOne !== clickedCard && !disableDeck) {
       clickedCard.classList.add("flip");
       if(!cardOne) {
           return cardOne = clickedCard;
       }
       cardTwo = clickedCard;
       disableDeck = true;
       let cardOneImg = cardOne.querySelector(".back-view img").src,
       cardTwoImg = cardTwo.querySelector(".back-view img").src;
       matchCards(cardOneImg, cardTwoImg);
   }
}

function matchCards(img1, img2) {
   if(img1 === img2) {
       matched++;
       if(matched == 8) {
           setTimeout(() => {
               return shuffleCard();
           }, 1000);
       }
       cardOne.removeEventListener("click", flipCard);
       cardTwo.removeEventListener("click", flipCard);
       cardOne = cardTwo = "";
       return disableDeck = false;
   }
   setTimeout(() => {
       cardOne.classList.add("shake");
       cardTwo.classList.add("shake");
   }, 400);

   setTimeout(() => {
       cardOne.classList.remove("shake", "flip");
       cardTwo.classList.remove("shake", "flip");
       cardOne = cardTwo = "";
       disableDeck = false;
   }, 1200);
}

function shuffleCard() {
   matched = 0;
   disableDeck = false;
   cardOne = cardTwo = "";
   let arr = [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8];
   arr.sort(() => Math.random() > 0.5 ? 1 : -1);
   cards.forEach((card, i) => {
       card.classList.remove("flip");
       let imgTag = card.querySelector(".back-view img");
       imgTag.src = `images/img-${arr[i]}.png`;
       card.addEventListener("click", flipCard);
   });
}

shuffleCard();

cards.forEach(card => {
   card.addEventListener("click", flipCard);
});
```

Output screenshot:



Explanation of Java Scripts

- `const cards = document.querySelectorAll(".card");`
  This line finds all the elements on the webpage that have a class of "card" and stores them in a constant variable named `cards`.

- `let matched = 0;`
  This sets up a counter named `matched` and starts it at 0. It will keep track of how many pairs of cards have been matched.

- `let cardOne, cardTwo;`
  These are variables to keep track of the first and second cards that are flipped over.

- `let disableDeck = false;`
  This is a flag used to prevent more cards from being flipped over when two cards are already flipped and being checked for a match.

- `function flipCard({target: clickedCard}) { ... }`
  This function is called when a card is clicked. It checks if the clicked card can be flipped and then flips it.

- Inside `flipCard`:
  - `if(cardOne !== clickedCard && !disableDeck) { ... }`
    This checks if the clicked card is not already the first card flipped and if the deck is not disabled.

  - `clickedCard.classList.add("flip");`
    This adds the "flip" class to the clicked card, which will usually make it flip over visually.

  - `if(!cardOne) { return cardOne = clickedCard; }`
    If no card has been flipped yet, this makes the clicked card the first card.

  - `cardTwo = clickedCard;`
    If the first card is already flipped, this makes the clicked card the second card.

  - `disableDeck = true;`
    This disables the deck to prevent flipping more cards until the current pair is processed.

  - `let cardOneImg = cardOne.querySelector(".back-view img").src, cardTwoImg = cardTwo.querySelector(".back-view img").src;`
    This gets the image sources from the flipped cards to compare them.

  - `matchCards(cardOneImg, cardTwoImg);`
    This calls another function to check if the two flipped cards match.

- `function matchCards(img1, img2) { ... }`
  This function checks if the two images on the flipped cards are the same.

- Inside `matchCards`:
  - `if(img1 === img2) { ... }`
    This checks if the images match.

  - `matched++;`
    If they match, increase the matched counter.

  - `if(matched == 8) { ... }`
    If all pairs are matched, call the `shuffleCard` function after a delay.

  - `cardOne.removeEventListener("click", flipCard);`
    This removes the click event from the matched cards so they can't be flipped again.

  - `cardOne = cardTwo = "";`
    This resets the variables for the next turn.

  - `disableDeck = false;`
    This re-enables the deck so more cards can be flipped.

- `function shuffleCard() { ... }`
  This function resets the game and shuffles the cards.

- Inside `shuffleCard`:
  - `matched = 0;`
    This resets the matched counter for a new game.

  - `let arr = [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8];`
    This creates an array with pairs of numbers from 1 to 8.

  - `arr.sort(() => Math.random() > 0.5 ? 1 : -1);`
    This shuffles the array in a random order.

  - `cards.forEach((card, i) => { ... });`
    This goes through each card and assigns a new image based on the shuffled array.

- `shuffleCard();`
  This calls the `shuffleCard` function to start the game.

- `cards.forEach(card => { card.addEventListener("click", flipCard); });`
  This adds a click event to each card that will call the `flipCard` function when a card is clicked