
숫자 카드 게임
게임 시작

숫자 카드 게임 규칙

1)카드 구성

플레이어와 컴퓨터는 각각 1~9까지의 숫자 카드를 가집니다.
같은 카드는 한 번만 사용 가능

2)게임 진행

플레이어가 카드를 클릭하면 그 카드를 내게 됩니다.
숫자가 더 큰 쪽이 그 라운드의 승자가 됩니다.
숫자가 같으면 무승부

3)라운드규칙

게임은 최대 8라운드까지 진행됩니다.
먼저 4라운드를 이긴 쪽이 즉시 최종 승리
8라운드가 끝났는데 승수가 같다면 무승부

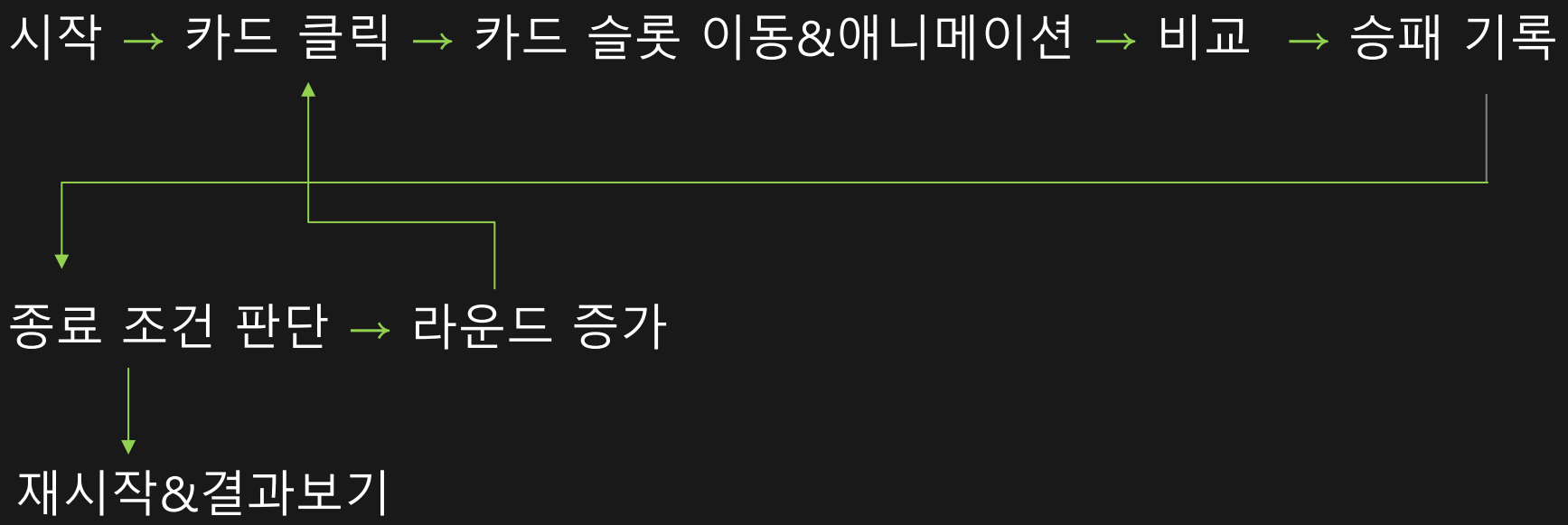
-HTML로 UI 구조 분리(intro , status , battlefield , player-hand)

-CSS로 배경/버튼/전광판 구성

-JS로 게임 상태 관리 + 이벤트 제어

자바스크립트의 비동기 흐름과 DOM 제어 CSS 애니메이션 구현에 초점

동작 흐름



변수 선언 및 객체 생성

```
<script>
const bgImg =[
  './game-image/bg-img/cat.png',
  './game-image/bg-img/catCry.png',
  './game-image/bg-img/catLose.png',
  './game-image/bg-img/catSmile2.png',
  './game-image/bg-img/catWin.png'
]
const preloaded = bgImg.map(src => {
  const img = new Image();
  img.src = src;
  return img;
});


let playerCards, computerCards, playerWins, computerWins, round, gameOver;

const cardImages = {
  1: './game-image/number/number1.png',
  2: './game-image/number/number2.png',
  3: './game-image/number/number3.png',
  4: './game-image/number/number4.png',
  5: './game-image/number/number5.png',
  6: './game-image/number/number6.png',
  7: './game-image/number/number7.png',
  8: './game-image/number/number8.png',
  9: './game-image/number/number9.png'
};
```

// 이미지를 미리 로드 후 저장과정

// preloaded는 image 객체를 생성해
이미지들을 비동기로 로드

// cardImages 객체에 1~9의 키를만들고 이미지
주소를 대입

핵심 기능 1)초기화 및 시작 로직

```
function initGame() {
  playerCards = [1,2,3,4,5,6,7,8,9];     //상태 변수들을 선언하고 초기화를 위해 초기값 설정
  computerCards = [1,2,3,4,5,6,7,8,9];
  playerWins = 0;
  computerWins = 0;
  round = 1;
  resultAlert='';
  gameOver = false;

  document.getElementById('round').textContent = round;
  document.getElementById('player-wins').textContent = playerWins;
  document.getElementById('computer-wins').textContent = computerWins;
  document.getElementById('game-result').textContent = '';
  document.getElementById('round-result').textContent = '';
  document.getElementById('player-choice-img').src = './game-image/number/numberback.png'
  document.getElementById('computer-choice-img').src = './game-image/number/numberback.png'



  document.body.style.backgroundImage = `url('${preloaded[0].src}')`;
  document.getElementById('player-cards').innerHTML = renderPlayerCards();
}                                                    // 카드 이미지 생성을 위한 renderPlayerCards() 호출
```

핵심 기능 2)카드 렌더링

```
playerCards = [1,2,3,4,5,6,7,8,9];
```

```
function renderPlayerCards() {
  return playerCards.map(c =>
    `<img src="${cardImages[c]}" onclick="playRound(${c})">`
  ).join('');
}
```

// .map()메서드를 이용한 1~9의 카드 이미지 생성

// 카드 이미지를 누를 시 playRound 함수 호출

```
const cardImages = {
  1: './game-image/number/number1.png',
  2: './game-image/number/number2.png',
  3: './game-image/number/number3.png',
  4: './game-image/number/number4.png',
  5: './game-image/number/number5.png',
  6: './game-image/number/number6.png',
  7: './game-image/number/number7.png',
  8: './game-image/number/number8.png',
  9: './game-image/number/number9.png'
};
```

핵심 기능 3)라운드 진행(playRound)

```
function renderPlayerCards() {
  return playerCards.map(c =>
    `<img src="${cardImages[c]}" onclick="playRound(${c})">`
  ).join('');
}


function playRound(playerCard) {
  if (gameOver) return; //gameover 값이 true면 리턴


  playerCards = playerCards.filter(c => c !== playerCard);     // .filter()메서드를 이용한 사용한 카드 제거 후 렌더링
  document.getElementById('player-cards').innerHTML = renderPlayerCards();

  const computerCard = computerChoose();

  animateBattle(playerCard, computerCard, () => {
    let roundResult = '';


    if (playerCard > computerCard) {
      playerWins++;
      roundResult = '플레이어 승!';
      document.body.style.backgroundImage = `url('${preloadedImages[1].src}')`;
```

```
function computerChoose() {  // 컴퓨터 카드 카드 선택 함수
  const randomIndex = Math.floor(Math.random() * computerCards.length);
  const choice = computerCards[randomIndex];
  computerCards = computerCards.filter(c => c !== choice);
  return choice;
}
```

핵심 기능 4)배틀 진행

```
function animateBattle(playerCard, computerCard, callback) {
  const playerImg = document.getElementById('player-choice-img');
  const computerImg = document.getElementById('computer-choice-img');

  playerImg.src = cardImages[playerCard];
  //computerImg.src = cardImages[computerCard];
  computerImg.src = './game-image/number/numberback.png';

  playerImg.style.transform = 'translateY(-20px) scale(1.6)'; //y축을 -20감소 시켜 위로 이동 ,scale을 1.6으로 설정해 커지게 설정
  computerImg.style.transform = 'translateY(-20px) scale(1.6)';

  setTimeout(() => {
    playerImg.style.transform = 'scale(1)'; //0.8초 뒤에는  style.transform에는 scale을 1로 변환값만 남도록 설정
    computerImg.style.transform = 'scale(1)';
    callback();
  }, 800);
}
```

// 카드 슬롯에 플레이어 선택 카드 이미지 대입

// 애니메이션 효과를 먼저 주기 위해
setTimeout 비동기 예약 함수 를 이용
->이 안에 callback을 작성

핵심 기능 5)점수/결과 표시

```
animateBattle(playerCard, computerCard, () => {
  let roundResult = '';  // 콜백 이후 진행되는 코드
  if (playerCard > computerCard) {
    playerWins++;
    roundResult = '플레이어 승!';
    document.body.style.backgroundImage = `url('${preloaded[1].src}')`;;
  } else if (playerCard < computerCard) {   // css는 객체를 읽지 못하므로 preloaded[i]에
    computerWins++;                           .src를 붙여서 문자열로 반환
    roundResult = '고양이 승!';
    document.body.style.backgroundImage = `url('${preloaded[3].src}')`;
  } else {
    roundResult = '무승부!';
    document.body.style.backgroundImage = `url('${preloaded[0].src}')`;
  }
  document.getElementById('round-result').textContent = roundResult;
  document.getElementById('player-wins').textContent = playerWins;
  document.getElementById('computer-wins').textContent = computerWins;

  resultAlert+= `${round}라운드 결과: 플레이어:${playerCard} -고양이:${computerCard} =>${roundResult}\n`;
```

핵심 기능 6)최종 결과 처리

```
    if (playerWins === 4 || computerWins === 4 || round >= 8) {
      gameOver = true; // 최종 승자 결정시 표시와 gameOver 설정
      let finalResult = '';
      if (playerWins > computerWins) {
        finalResult = '플레이어 최종 승리!';
        document.body.style.backgroundImage = `url('${preloaded[2].src}')`;
      } else if (playerWins < computerWins) {
        finalResult = '고양이 최종 승리!';
        document.body.style.backgroundImage = `url('${preloaded[4].src}')`;
      } else {
        finalResult = '최종 무승부!';
        document.body.style.backgroundImage = `url('${preloaded[0].src}')`;
      }
      document.getElementById('game-result').textContent = finalResult;
      resultAlert += `\n ${finalResult}`;
      document.getElementById('restart-btn').style.display = 'inline-block';
      document.getElementById('result-btn').style.display = 'inline-block';
      return;                                    // 결과 버튼,재시작 버튼을 보여짐.
    }
    round++;
    document.getElementById('round').textContent = round;
  });

}
```

Q&A

Thank you