

HTML 캔버스를 이용 한 게임 구현

2018875036 송준엽

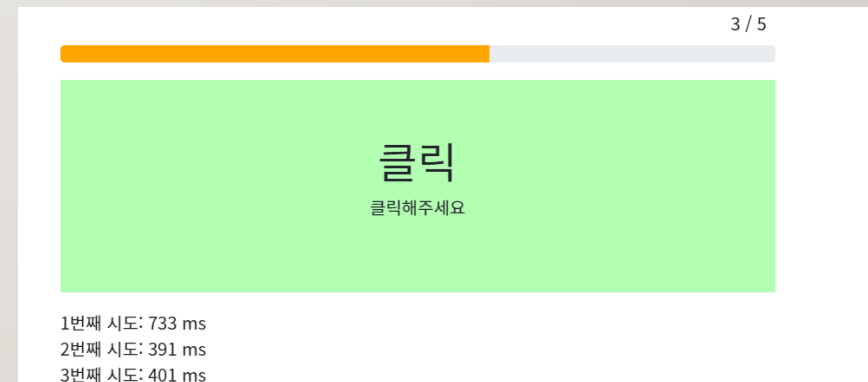
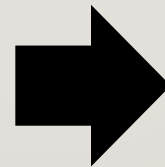
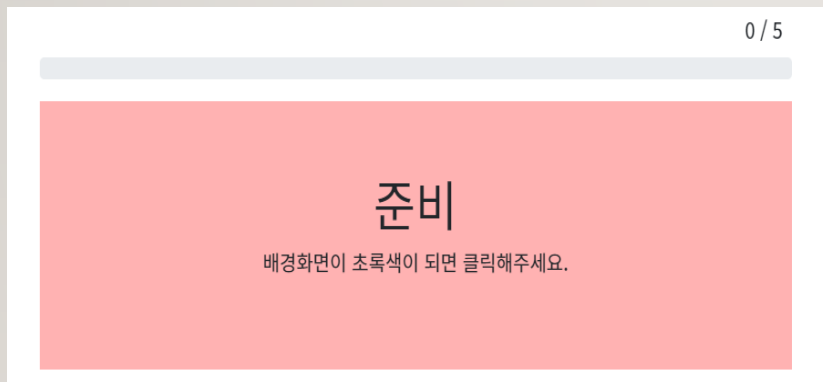


2 개요

- 1. 기존 구현된 프로그램 설명
- 2. 추가 기능 고안 및 UI순서도 설명
- 3. 추가 기능 설명
- 4. 플레이 영상
- 5. 참고

3 | 기존 구현된 프로그램 설명

- 어떤 프로그램을 참고 할까 하다가 제가 즐겨보는 유튜버가 반응 속도 테스트를 하기 위해 웹에서 검색해서 이용한 것이 생각나서 반응속도 테스트기를 참고 하기로 했습니다.



4 | 기존 구현된 프로그램 설명

1) 돔 설명 및 캔버스 설명

- Html과 CSS부분은 생략하고 주요 script 부분만 설명 하겠습니다.

```
<div id = "description"></div>
  <canvas id="canvas" class="waiting"></canv>
  <div id="show"></div>

<script>
  const canvas= document.querySelector('#canvas');
  const ctx = canvas.getContext("2d");
  const description = document.querySelector('#description');
  ...
```

“canvas”의 id로 돔
으로 접근하고
class로 상태를 나
타냅니다

canvas.getContext(“
2d”)함수를 이용해
2D상의 위치를 정
의 합니다.

5 | 기존 구현된 프로그램 설명

2) LOAD 이벤트

```
window.addEventListener('load', function(){  
    description.textContent = 'Click to start';  
    console.log("onLoad");  
});
```

‘load’이벤트 리스너에 설명란의 텍스트를 초기화 하는 함수를 바인딩합니다.

6 | 기존 구현된 프로그램 설명

3.1) CLICK이벤트

```
canvas.addEventListener('click', function () {  
  if (canvas.classList.contains('waiting')) {  
    canvas.classList.remove('waiting');  
    canvas.classList.add('ready');  
    description.textContent = 'wait for green';  
  
    timer = setTimeout(function () {  
      start = new Date();  
      canvas.click();  
    }, Math.floor(Math.random() * 1000) + 2000);  
  }  
}
```

‘click’이벤트리스너에 현재 캔버스의 상태를 판단(class)해서 ‘waiting’중이라면 ‘ready’로 상태를 변환하고 랜덤한 시간 후에 클릭을 기다리는 상태로 변환합니다.

7 | 기존 구현된 프로그램 설명

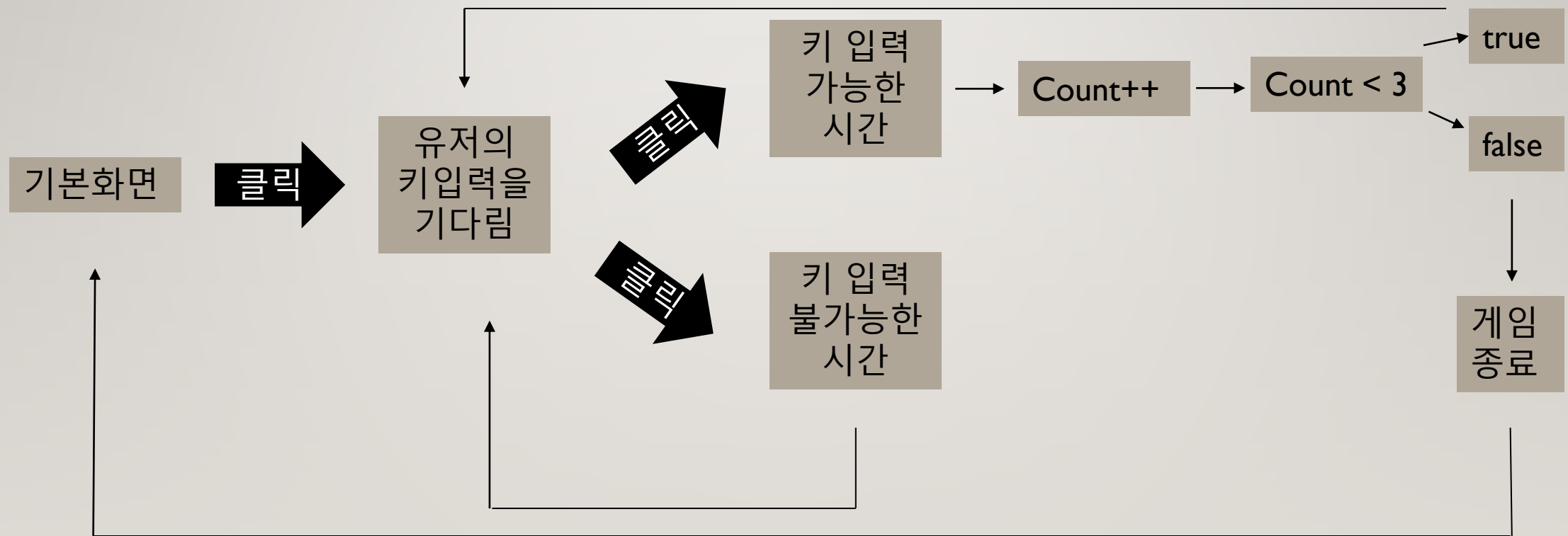
3.2) CLICK이벤트

```
else if (canvas.classList.contains('ready')) {  
  if (!start) {  
    clearTimeout(timer);  
    canvas.classList.remove('ready');  
    canvas.classList.add('waiting');  
    description.textContent = 'Clicked too soon';  
  }  
}
```

‘click’이벤트리스너에 현재 캔버스의 상태를 판단(class)해서 ‘ready’중이고 아직 클릭을 하면 안될 때 클릭을 하면 호출 됩니다.

8 | 기존 구현된 프로그램 설명

4) 순서도



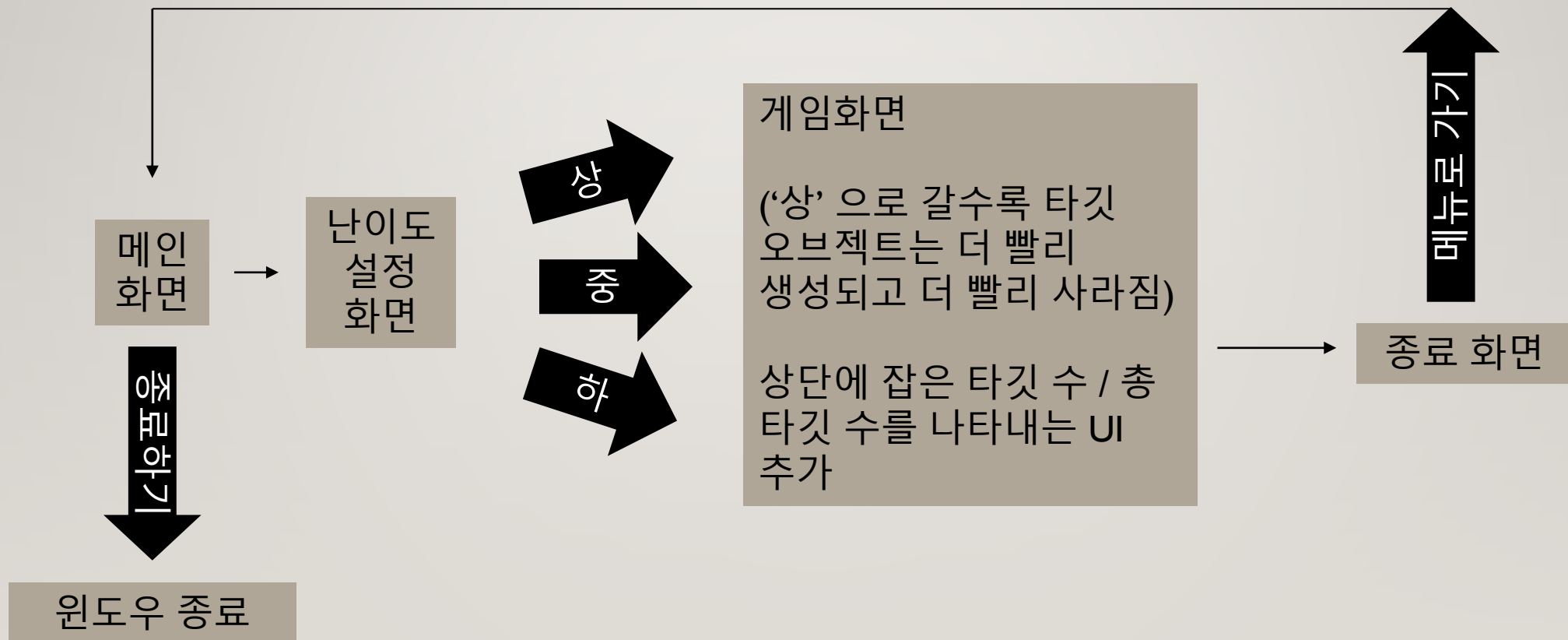
9 2. 추가기능 고안 및 UI순서도 설명

1) 구현할 추가기능 설명

- 추가기능은 유저의 클릭을 확인 및 스크립트 내 시간을 다루는 것을 넘어 유저의 클릭 위치, 시간, 그리기 등을 사용하여 FPS게임의 사격장을 html 캔버스를 이용하여 구현해 보려고 합니다. 기획단계에서 모든 기능은 구현 가능하다고 가정하고 설계 했습니다.

10 2. 추가기능 고안 및 UI순서도 설명

2) UI순서도 설명



II 3 추가기능 설명

I) 캔버스 추가 및 스크립트 할당

```
<canvas id="background" class="main" width="1200" height="700" style="margin-left: 20px; margin-top: 20px;"></canvas>
```

```
<script>
```

```
//돔 오브젝트 저장
```

```
const canvas = document.getElementById("background");
```

```
const ctx = canvas.getContext("2d");
```

```
const imageElem = new Image();
```

```
let startButtonPos = [200, 400, 600, 600];
```

```
let exitButtonPos = [700, 400, 1100, 600];
```

I2 3 추가기능 설명

2.1) 메인 메뉴 그리기

```
function DrawSquareAndText(x, y, width, height, text){  
    ctx.fillStyle = 'black';  
    ctx.font = '48px serif';  
    ctx.textAlign = "center"  
    ctx.strokeRect(x, y, width, height);  
    ctx.fillText(text, x + (width - text.length * ctx.font.length) / 2, y + height /  
2);  
}
```

I3 3 추가기능 설명

2.2) 메인 메뉴 그리기

```
//메인메뉴 그려지는 코드
function MainMenu(){
    ctx.strokeRect(0, 0, 1200, 700);
    DrawSquareAndText(400, 100, 400, 200, '에임 연습하기');

    DrawSquareAndText(startButtonPos[0], startButtonPos[1],
        startButtonPos[2] - startButtonPos[0], startButtonPos[3] -
        startButtonPos[1], '게임 시작');
    DrawSquareAndText(exitButtonPos[0], exitButtonPos[1],
        exitButtonPos[2] - exitButtonPos[0], exitButtonPos[3] -
        exitButtonPos[1], '종료하기');
}
```


14 3 추가기능 설명

2.1) 난이도 설정화면 그리기

//난이도메뉴 그려지는 코드

```
function DifficultyMenu() {  
    ctx.clearRect(0,0,1200,700)  
    ctx.strokeRect(0, 0, 1200, 700);  
  
    DrawSquareAndText(difficultyButtonPos[0][0], difficultyButtonPos[0][1],  
                      difficultyButtonPos[0][2] - difficultyButtonPos[0][0],  
                      difficultyButtonPos[0][3] - difficultyButtonPos[0][1], '어려움');  
    DrawSquareAndText(difficultyButtonPos[1][0], difficultyButtonPos[1][1],  
                      difficultyButtonPos[1][2] - difficultyButtonPos[1][0],  
                      difficultyButtonPos[1][3] - difficultyButtonPos[1][1], '보통');  
    DrawSquareAndText(difficultyButtonPos[2][0], difficultyButtonPos[2][1],  
                      difficultyButtonPos[2][2] - difficultyButtonPos[2][0],  
                      difficultyButtonPos[2][3] - difficultyButtonPos[2][1], '쉬움');  
}
```

15 3 추가기능 설명

3.0) 게임화면 그리기

```
let targetPosAndMakeTime = []; //x, y, makeTime, liveTime
let targetGenSpeed = [500, 1000]; //min, max
let targetLiveTime = [1000, 2000]; //min, max
let lastSpawnTime = new Date().getTime();

let gamePlayinterval;
const maxTarget = 10;
let targetCount = 0;
let caughtCount = 0;
```

16 3 추가기능 설명

3.1) 게임화면 그리기

```
function DrawGameScreen() {  
    //인터벌 종료를 판단하는 문장  
    if (targetCount >= maxTarget) {  
        clearInterval(gamePlayinterval);  
  
        ctx.drawImage(imageElem, 0, 0, canvas.offsetWidth,  
            canvas.offsetHeight);  
        return;  
    }  
    ctx.drawImage(imageElem, 0, 0, canvas.offsetWidth,  
        canvas.offsetHeight);  
    //모기 이미지 그리기  
    for(let i = 0; i < targetPosAndMakeTime.length; i++){  
        ctx.drawImage(targetImg, targetPosAndMakeTime[0], targetPosAndMakeTime[1], 30,  
            30);  
    }  
}
```

17 3 추가기능 설명

3.2) 게임화면 그리기

```
//모기 삭제 코드
if(targetPosAndMakeTime[i][2].getTime() + targetPosAndMakeTime[i][3] <= new
Date().getTime()){
    targetPosAndMakeTime.slice(i, 1);
    continue;
}

ctx.drawImage(targetImg, targetPosAndMakeTime[i][0], targetPosAndMakeTime[i][1],
30, 30);
}
```


18 3 추가기능 설명

3.3) 게임화면 그리기

```
let targetGenSpeedTime = Math.floor(Math.random() * (targetGenSpeed[1] - targetGenSpeed[0])) +
    targetGenSpeed[0];
    //새 모기 생성
if (new Date() - lastSpawnTime >= targetGenSpeedTime) {
    let tmpX = Math.floor(Math.random() * 1140) + 30;
    let tmpY = Math.floor(Math.random() * 540) + 30;
    let tmpLiveTime = Math.floor(Math.random() * (targetLiveTime[1] - targetLiveTime[0])) +
        targetLiveTime[0];

    targetPosAndMakeTime.push([tmpX, tmpY, new Date(), tmpLiveTime])
    lastSpawnTime = new Date();

    targetCount++;
}

scoreBoard.textContent = targetCount + " / " + maxTarget;
}
```


19 3 추가기능 설명

4) 추가된 점

메인화면, - 윈도우 닫기

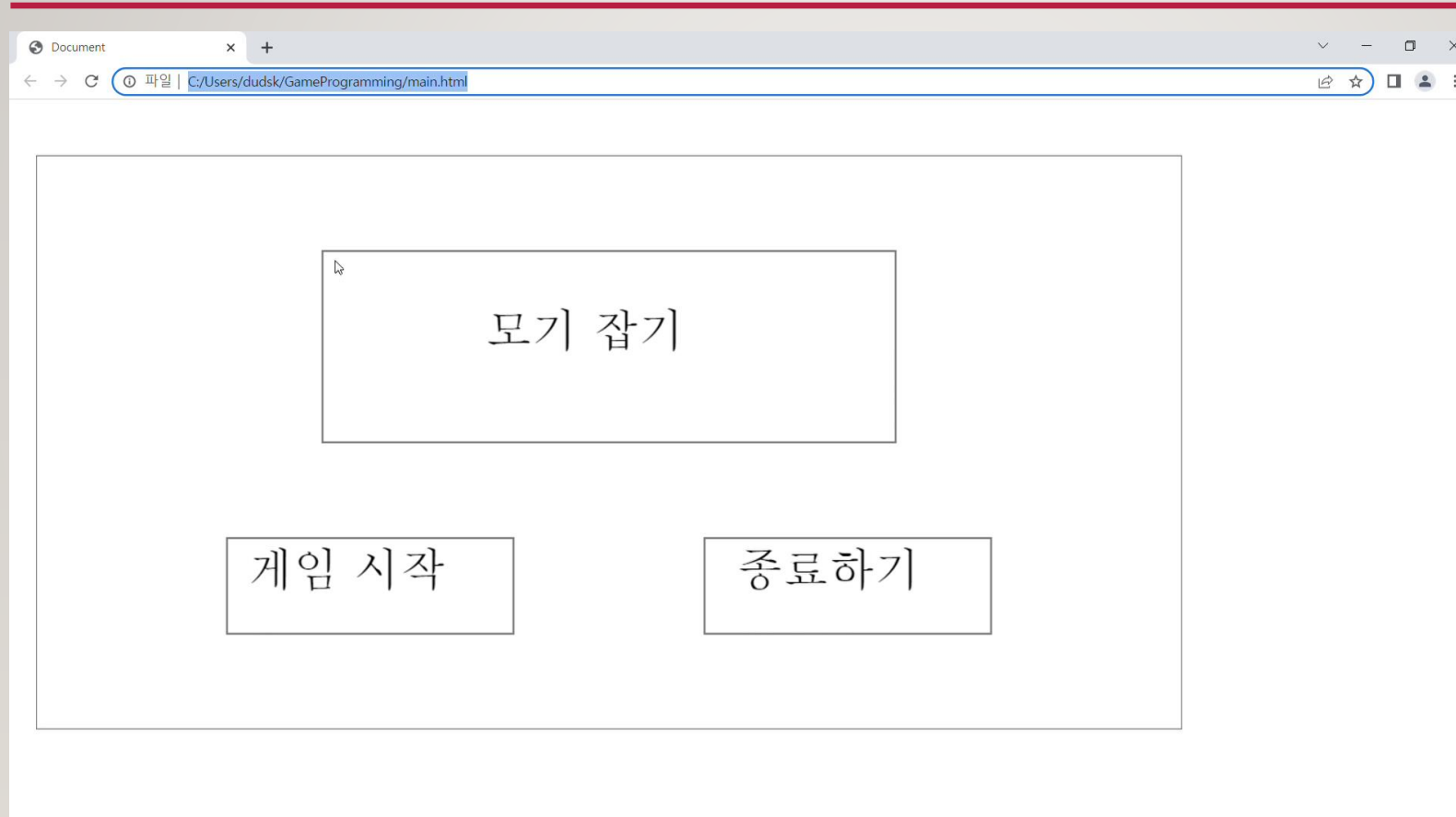
난이도 선택 화면 - 상, 중, 하

게임 화면 - 배경 이미지 넣기, 랜덤하게 게임 오브젝트 생성 및 삭제
클릭시 마우스 커서의 위치를 읽어옴

스코어 화면

그 외 - 마우스 커서 변경, 윈도우 닫기

20 4. 플레이 영상



유튜브 링크 :
<https://youtu.be/TfBjOal2tys>

21 5 참고

- 반응속도 테스트기(기본 소스): <https://goddino.tistory.com/108>
- Div태그로 구현된 영역 canvas로 교체 : <https://aljjabaegi.tistory.com/602>
- 캔버스에 이미지 삽입 : [https://webisfree.com/2018-07-02/\[html5\]-canvas-%EC%9A%94%EC%86%8C%EC%97%90-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%B6%94%EA%B0%80%ED%95%98%EA%B8%B0](https://webisfree.com/2018-07-02/[html5]-canvas-%EC%9A%94%EC%86%8C%EC%97%90-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%B6%94%EA%B0%80%ED%95%98%EA%B8%B0)
- 윈도우창 종료 : <https://7942yongdae.tistory.com/76>
- 마우스 커서 변경 :
<http://superkts.pe.kr/helper/view.php?seq=66&PHPSESSID=044e30d45dcda45351db8d983a717ce2>