

Canvas 게임 UPGRADE

블록 피하기

학번 : 2021864039

이름 : 김태현



프로그램 목표

1.

변경 사항

3.

추가 개선점

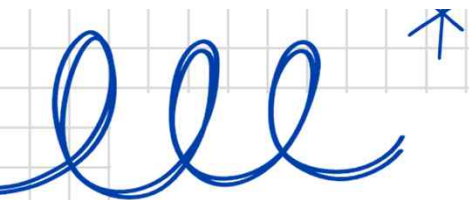
2.

소스 코드 분석

4.

출처





변경 사항

1. 인트로 화면 추가

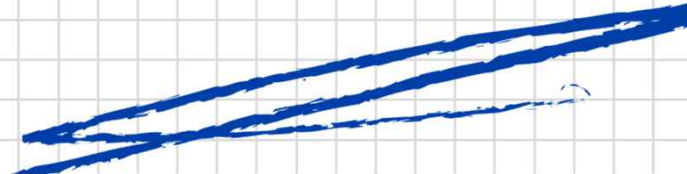
2. 종료 화면 추가

3. 캐릭터 이미지 변경

4. 재시작 기능

5. 게임 속도 변경

6. 조작법 변경



INTRO SCREEN GAME OVER SCREEN

```
#intro-screen, #game-over-screen {  
  position: absolute;  
  width: 480px;  
  height: 270px;  
  background: url('intro_screen.jpg') center/cover;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  z-index: 999;  
}  
  
/*game_over_screen의 추가 속성*/  
#game-over-screen {  
  background: □#00000095; /* 반투명한 배경색 */  
  color: ■#ff0000;  
  display: none;  
}
```

Align-items : center

- 컨텐츠 아이템의 내부 상하관계 정렬 상태를 설정
- 가운데로 정렬

Z-index

- 창을 순서대로 배치하는 스타일

Justify-content : center

- 플렉스 요소의 수평 방향 정렬 방식을 설정
- 플렉스 요소는 플렉스 컨테이너의 가운데에서부터 배치

#game-over-screen

Game-over-screen 속성을 추가적으로 설정

PRESS START

변경점

```


<body>
<div id="intro-screen" onclick="startGame()">
  <!-- <h3>화면을 클릭하여 게임을 시작하세요</h3> -->
</div>
<div id="game-over-screen" onclick="restartGame()">
  <h2 id="final-score"></h2>
</div>
```

캐릭터 이미지 등록

Intro-screen onclick = "startGame()"
 이미지를 클릭하면 게임실행

Game-over-screen onclick = "restartGame()"
 Game이 종료되고 화면을 클릭하면 재실행

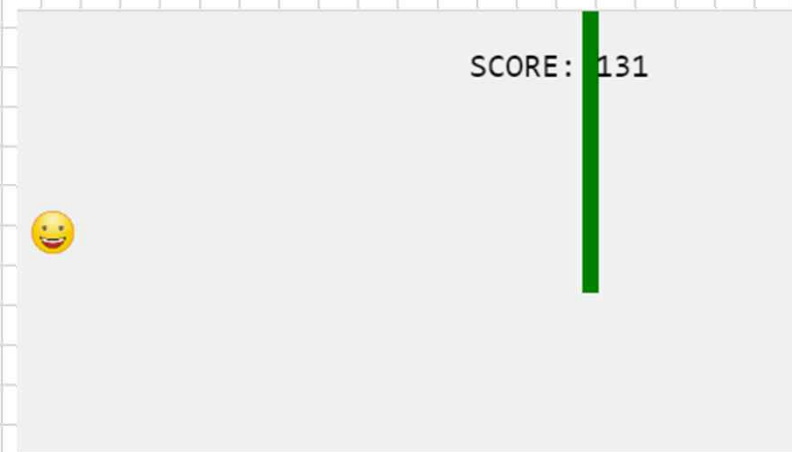
StartGame 변경점

```
function startGame() {  
  var introScreen = document.getElementById('intro-screen');  
  introScreen.style.display = 'none';  
  myGamePiece = new component(30, 30, "red", 10, 120, "image");  
  mySound = new sound("bounce.mp3");  
  myMusic = new sound("gametheme.mp3");  
  myMusic.play();  
  myScore = new component("20px", "Consolas", "black", 280, 40, "text");  
  myGameArea.start();  
}
```

introScreen = document.getElementById('intro-screen')
Introscreen 호출

introScreen.style.display = 'none'
Intro screen 숨김

myGamePiece
캐릭터를 이미지로 대체



* showGameOverScreen restartGame

showGameOverScreen()

게임이 끝나고 최종 점수를 나타낸다

restartGame()

게임을 재시작하는 함수

GetElementById로 game-over-screen에 접근하여
Text 등록



```
function showGameOverScreen() {  
    var gameOverScreen = document.getElementById('game-over-screen');  
    gameOverScreen.style.display = 'flex';  
    document.getElementById('final-score').innerText =  
        "GAME OVER\n" + "SCORE : " + myGameArea.frameNo + "\nClick to RESTART";  
    myMusic.stop();  
}  
  
function restartGame() {  
    var gameOverScreen = document.getElementById('game-over-screen');  
    gameOverScreen.style.display = 'none';  
    myGameArea.clear();  
    myObstacles = [];  
    myGamePiece.changeImage("happy.png");  
    startGame();  
}
```



Stop




```
var myGameArea = {  
  canvas: document.createElement("canvas"),  
  start: function () {  
    this.canvas.width = 480;  
    this.canvas.height = 270;  
    this.context = this.canvas.getContext("2d");  
    document.body.insertBefore(this.canvas, document.body.childNodes[0]);  
    this.frameNo = 0;  
    this.interval = setInterval(updateGameArea, 20);  
  },  
  clear: function () {  
    this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);  
  },  
  stop: function () {  
    clearInterval(this.interval);  
    showGameOverScreen();  
  }  
}
```

Stop: function

showGameOverScreen() 추가

캐릭터가 충돌되어 게임이 종료되면
게임오버화면을 불러온다.



component

```
function component(width, height, color, x, y, type) {
  this.type = type;
  this.width = width;
  this.height = height;
  this.speedX = 0;
  this.speedY = 0;
  this.x = x;
  this.y = y;
  this.image = document.getElementById("happy");
  if(type == "image"){
    this.happyImage = document.getElementById("happy_img");
    this.image = this.happyImage;
    this.image.style.width = this.width + "px";
    this.image.style.height = this.height + "px";
  }
  this.update = function () {
    ctx = myGameArea.context;
    if (this.type == "text") {
      ctx.font = this.width + " " + this.height;
      ctx.fillStyle = color;
      ctx.fillText(this.text, this.x, this.y);
    }else if (type == "image") {
      ctx.drawImage(this.image, this.x, this.y, this.width, this.height);
    }else {
      ctx.fillStyle = color;
      ctx.fillRect(this.x, this.y, this.width, this.height);
    }
  }
}
```

GetElementById로 캐릭터 이미지 불러온다.
이미지의 크기를 블록의 이미지와 동일하게 한다.

Update

text, image 그 외의 경우에는 사각형을 그린다.

게임 속도 조절

```
if(myGameArea.frameNo <= 500){
    newInterval = 20;
}else if(myGameArea.frameNo <= 1000){
    clearInterval(myGameArea.interval);
    newInterval = 15;
}else if(myGameArea.frameNo <= 1500){
    clearInterval(myGameArea.interval);
    newInterval = 12;
}else if(myGameArea.frameNo <= 2000){
    clearInterval(myGameArea.interval);
    newInterval = 10;
}else if(myGameArea.frameNo <= 2500){
    clearInterval(myGameArea.interval);
    newInterval = 8;
}else if(myGameArea.frameNo <= 3000){
    clearInterval(myGameArea.interval);
    newInterval = 5;
}else{
    newInterval = 1;
}
if(myGameArea.interval !== newInterval){
    clearInterval(myGameArea.interval);
    myGameArea.interval = setInterval(updateGameArea, newInterval);
}
```

myGameArea.frameNo = SCORE 에 따라
게임속도를 점진적으로 빠르게한다.

현재 Interval을 중지하고 newInterval로 설정한다.



사용자 입력처리

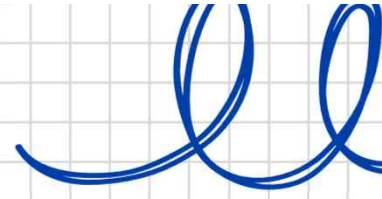
키보드 이벤트 처리

캐릭터의 이동을 버튼 → 키보드 입력으로 변경

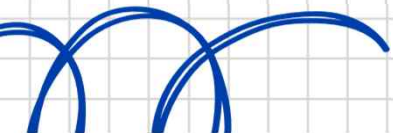
```
document.addEventListener("keydown", function (e) {  
    // 키보드 이벤트 처리: 방향키로 블록 움직임  
    switch (e.keyCode) {  
        case 37: // 왼쪽 화살표  
            moveleft();  
            break;  
        case 39: // 오른쪽 화살표  
            moveright();  
            break;  
        case 38: // 위쪽 화살표  
            moveup();  
            break;  
        case 40: // 아래쪽 화살표  
            movedown();  
            break;  
    }  
});  
  
document.addEventListener("keyup", function (e) {  
    // 키보드 이벤트 처리: 방향키 떼 때 블록 정지  
    switch (e.keyCode) {  
        case 37:  
        case 39:  
            clearmoveX();  
            break;  
        case 38:  
        case 40:  
            clearmoveY();  
            break;  
    }  
});
```



게임 플레이 화면

A black rectangular area representing a game screen. In the center, the text "PRESS START" is written in a white, pixelated, uppercase font. A small white mouse cursor arrow is positioned over the letter 'A' in "START".

PRESS START





개선점

1. 게임 배경화면 추가
- 2.캐릭터 충돌 시 이미지 변경
- 3.랭킹 시스템
- 4.시작 전 난이도 선택
- 5.캐릭터 목숨(장애물에 충돌하면 무시 또는 튕겨짐)



출 처

[Position](#)

[justify_content](#)

[align-items](#)

[Z-index](#)

[GetElementById](#)

Chat-GPT