

게임 프로그래밍

HTML 게임

소 프 트 웨 어 학 과 2 0 2 1 8 6 3 0 6 7 정 혜 인

2 0 2 3 / 1 1 / 0 8

CONTENTS



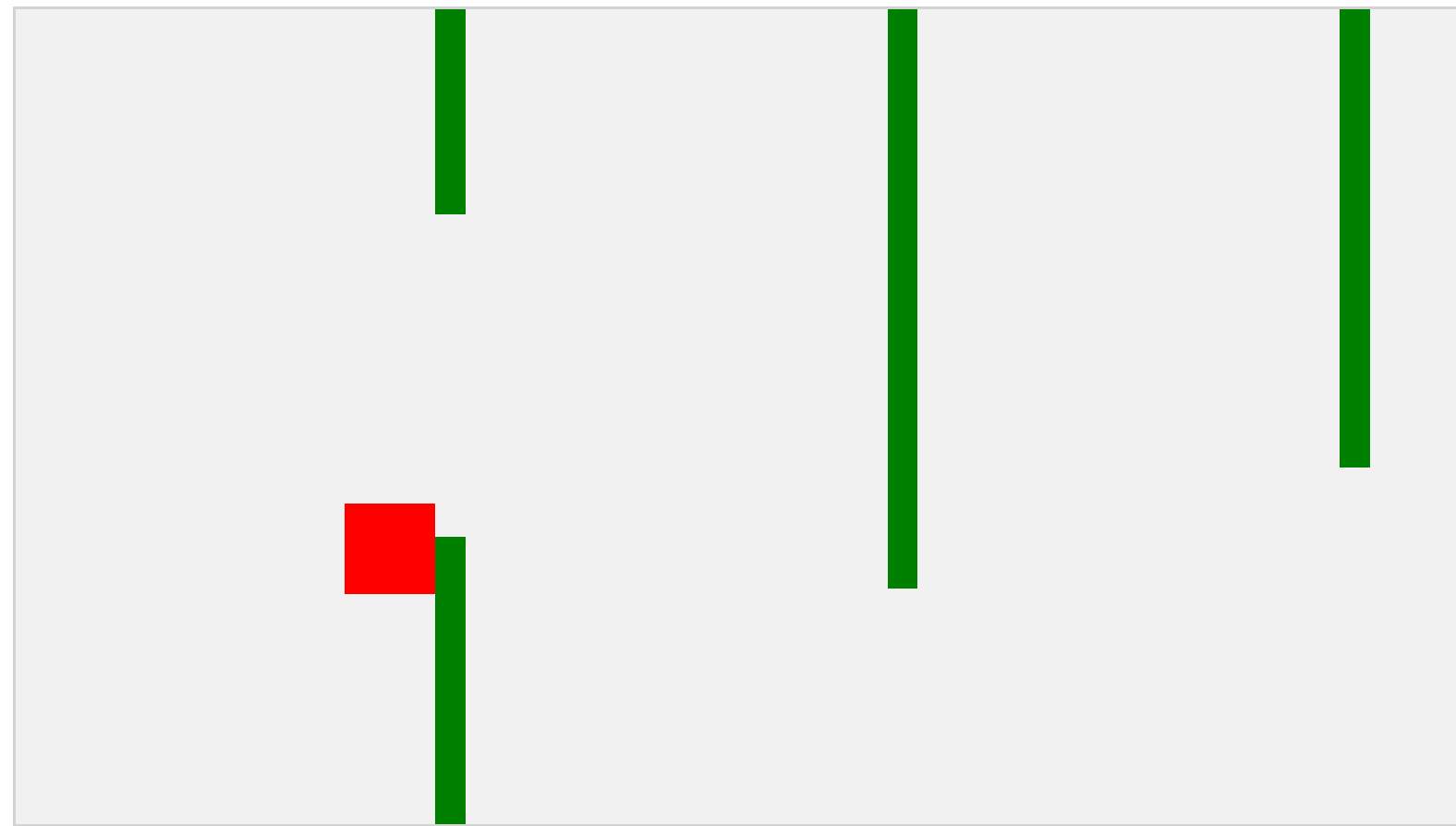
01

게임 소개

02

코드 분석

게임 소개



UP

LEFT

RIGHT

DOWN

- 캐릭터를 위, 아래, 왼쪽, 오른쪽으로 조작하여 일정한 간격으로 생성되는 장애물을 피하는 게임

Turn up the volume. Do you hear a "dunk" when the red square hits an obstacle?

코드 분석

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<style>
canvas {
    border:1px solid #d3d3d3;
    background-color: #f1f1f1;
}
</style>
</head>
<body onload="startGame()">
<script>
```

```
var myGamePiece;
var myObstacles = [];
var mySound;
```

- <!DOCTYPE html> : HTML 문서의 문서 유형을 정의한다.
- <html> : HTML 문서의 시작을 나타내며 전체 HTML 구조를 감싼다.
- <head> : HTML 문서의 머리말 부분을 정의한다.
- <meta name="viewport" content="width=device-width, initial-scale=1.0"/> :
모바일 기기에서 화면 폭을 기기 폭에 맞게 조절하고 초기 확대 수준을 1.0으로 설정한다.

코드 분석

```
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);
  },
  stop : function() {
    clearInterval(this.interval);
  },
  clear : function() {
    this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
  }
}
```

- var myGameArea = {...} : 객체를 정의하여 게임 영역을 관리한다.
- stop : function() {...} : stop 메소드는 clearInterval(this.interval); 구문을 실행하여 this.interval에 저장된 인터벌을 제거하여 게임 루프를 중지시킨다.

코드 분석

```
function component(width, height, color, x, y) {
  this.width = width;
  this.height = height;
  this.speedX = 0;
  this.speedY = 0;
  this.x = x;
  this.y = y;
  this.update = function() {
    ctx = myGameArea.context;
    ctx.fillStyle = color;
    ctx.fillRect(this.x, this.y, this.width, this.height);
  }
  this.newPos = function() {
    this.x += this.speedX;
    this.y += this.speedY;
  }
  this.crashWith = function(otherobj) {
    var myleft = this.x;
    var myright = this.x + (this.width);
    var mytop = this.y;
    var mybottom = this.y + (this.height);
    var otherleft = otherobj.x;
    var otherright = otherobj.x + (otherobj.width);
    var othertop = otherobj.y;
    var otherbottom = otherobj.y + (otherobj.height);
    var crash = true;
    if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright))
    {
      crash = false;
    }
    return crash;
  }
}
```

- component 함수 : 캐릭터나 장애물 등을 생성하고 제어하기 위해 사용한다.
- this.update = function() {...} : 게임 캔버스에 객체를 그린다.
- this.newPos = function() {...} : 객체의 새로운 위치를 업데이트한다.
- this.crashWith = function(otherobj) {...} : 이 객체가 다른 객체와 충돌하는지를 확인한다.

코드 분석

```
function updateGameArea() {
  var x, height, gap, minHeight, maxHeight, minGap, maxGap;
  for (i = 0; i < myObstacles.length; i += 1) {
    if (myGamePiece.crashWith(myObstacles[i])) {
      mySound.play();
      myGameArea.stop();
      return;
    }
  }
  myGameArea.clear();
  myGameArea.frameNo += 1;
  if (myGameArea.frameNo == 1 || everyinterval(150)) {
    x = myGameArea.canvas.width;
    minHeight = 20;
    maxHeight = 200;
    height = Math.floor(Math.random()*(maxHeight-minHeight+1)+minHeight);
    minGap = 50;
    maxGap = 200;
    gap = Math.floor(Math.random()*(maxGap-minGap+1)+minGap);
    myObstacles.push(new component(10, height, "green", x, 0));
    myObstacles.push(new component(10, x - height - gap, "green", x, height + gap));
  }
  for (i = 0; i < myObstacles.length; i += 1) {
    myObstacles[i].x -= 1;
    myObstacles[i].update();
  }
  myGamePiece.newPos();
  myGamePiece.update();
}
```

- var x, height, gap, minHeight, maxHeight, minGap, maxGap : 장애물의 위치, 크기, 간격 변수
- for (i = 0; i < myObstacles.length; i += 1) {...} : 배열의 모든 장애물을 순회하면서 충돌 여부를 확인한다.
- if (myGamePiece.crashWith(myObstacles[i])) {...} : myGamePiece 객체와 장애물이 충돌하는지를 확인한다.

코드 분석

```
function sound(src) {  
    this.sound = document.createElement("audio");  
    this.sound.src = src;  
    this.sound.setAttribute("preload", "auto");  
    this.sound.setAttribute("controls", "none");  
    this.sound.style.display = "none";  
    document.body.appendChild(this.sound);  
    this.play = function(){  
        this.sound.play();  
    }  
    this.stop = function(){  
        this.sound.pause();  
    }  
}
```

- function sound(src) : 주어진 오디오 파일 경로(src)를 사용하여 HTML5 <audio> 엘리먼트를 동적으로 생성하고 제어한다.

코드 분석

```
function everyinterval(n) {  
    if ((myGameArea.frameNo / n) % 1 == 0) {return true;}  
    return false;  
}
```

```
function moveup() {  
    myGamePiece.speedY = -1;  
}
```

```
function movedown() {  
    myGamePiece.speedY = 1;  
}
```

```
function moveleft() {  
    myGamePiece.speedX = -1;  
}
```

```
function moveright() {  
    myGamePiece.speedX = 1;  
}
```

```
function clearmove() {  
    myGamePiece.speedX = 0;  
    myGamePiece.speedY = 0;  
}
```

- function everyinterval(n) : 현재 게임 프레임 번호 (myGameArea.frameNo)를 n으로 나눈 나머지를 계산하고, 그 결과가 0이면 true를 반환하고, 그렇지 않으면 false를 반환한다.
- function move..() : 게임 캐릭터를 움직이게 하는 함수

코드 분석

```
function startGame() {
    myGamePiece = new component(30, 30, "red", 10, 120);
    mySound = new sound("bounce.mp3");
    myGameArea.start();
}

</script>
<div style="text-align:center;width:480px;">
    <button onmousedown="moveup()" onmouseup="clearmove()" ontouchstart="moveup()">UP</button><br><br>
    <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>
    <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>
    <button onmousedown="movedown()" onmouseup="clearmove()" ontouchstart="movedown()">DOWN</button>
</div>
<p>Turn up the volume. Do you hear a "dunk" when the red square hits an obstacle?</p>
</body>
</html>
```

- <div style="text-align:center;width:480px;">: 이 <div> 엘리먼트는 버튼과 게임 설명 텍스트를 가운데 정렬하고, 폭을 480px로 설정한다.
- <button ... > : 버튼을 생성하고 이 버튼은 마우스로 클릭하거나 터치 이벤트를 통해 move...() 함수를 호출하고, 마우스를 떼면 clearmove() 함수를 호출한다.

출처

—

[ChatGPT](#)

https://www.w3schools.com/graphics/game_sound.asp

THANK
YOU EVERYONE

감 사 합 니 다