게임프로그래밍 발표

2019775031 성재원

시작 메뉴

목차

1. 게임 소개

2. 코드 설명

게임 소개

빨간 네모를 클릭으로 조종해서
장애물과 충돌하지 않고
오래 버티는 게임

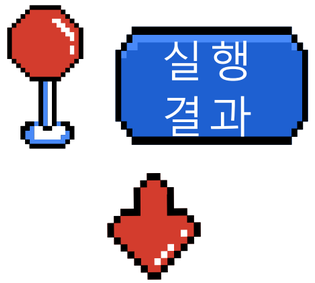
점수 표시

점수는 게임 오버되기 전까지
일정하게 증가

클릭으로 방향 조절

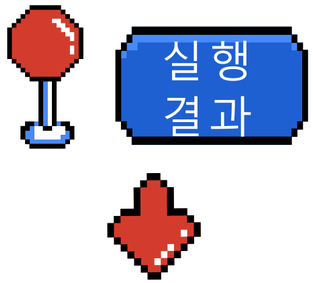
상하좌우 버튼 클릭으로
네모를 움직임.

코드 소개




```
<!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>

function startGame() {
    myGameArea.start();
}

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]);
    }
}

</script>

<p>We have created a game area! (or at least an empty canvas)</p>

</body>
</html>
```

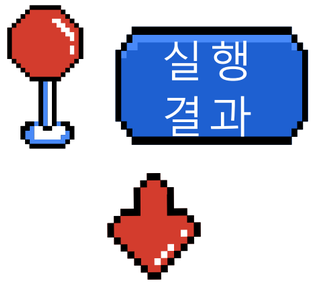
코드 소개



```
    var myGamePiece;

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

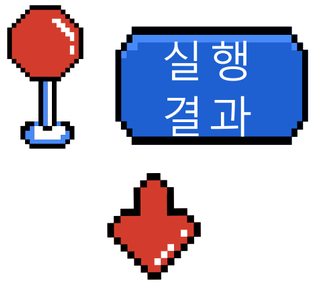
```
var myGameArea = {
  canvas : document.createElement("canvas"),
  start : function() {
    this.canvas.width = 480;
    this.canvas.height = 270;
    this.context = this.canvas.getContext("2d");
    document.body.insertBefore(this.canvas, document.body.childNodes[0]);
    this.interval = setInterval(updateGameArea,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

function component(width, height, color, x, y) {
  this.width = width;
  this.height = height;
  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);
  }
}

function updateGameArea() {
  myGameArea.clear();
  myGamePiece.update();
}
```

코드 소개




```
function updateGameArea() {
  myGameArea.clear();
  myGamePiece.x += 1;
  myGamePiece.update();
}
```

코드 소개



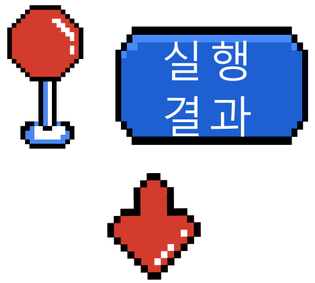
위로

왼쪽 오른쪽

아래에

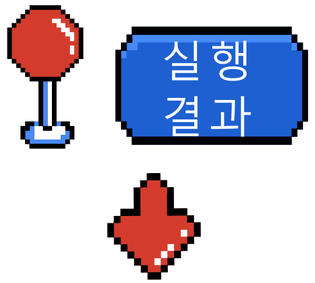
```
<script>
function component(width, height, color, x, y) {
  this.width = width;
  this.height = height;
  this.speedX = 0;
  this.speedY = 0;
  this.x = x;
  this.y = y;
  this.update = function() {
    ctx = myGameArea.context;
    ctx.fillStyle = color;
    ctx.fillRect(this.x, this.y, this.width, this.height);
  }
  this.newPos = function() {
    this.x += this.speedX;
    this.y += this.speedY;
  }
}

function updateGameArea() {
  myGameArea.clear();
  myGamePiece.newPos();
  myGamePiece.update();
}

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

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

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

function moveright() {
  myGamePiece.speedX += 1;
}
</script>

<button onclick="moveup()">UP</button>
<button onclick="movedown()">DOWN</button>
<button onclick="moveleft()">LEFT</button>
<button onclick="moveright()">RIGHT</button>
```

코드 소개



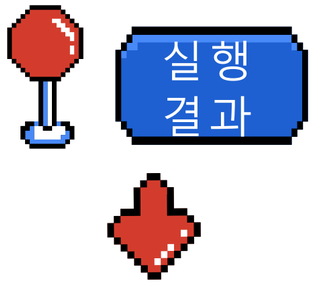

```
function stopMove() {
  myGamePiece.speedX = 0;
  myGamePiece.speedY = 0;
}
</script>

<button onmousedown="moveup()" onmouseup="stopMove()" ontouchstart="moveup()">UP</button>
<button onmousedown="movedown()" onmouseup="stopMove()" ontouchstart="movedown()">DOWN</button>
<button onmousedown="moveleft()" onmouseup="stopMove()" ontouchstart="moveleft()">LEFT</button>
<button onmousedown="moveright()" onmouseup="stopMove()" ontouchstart="moveright()">RIGHT</button>
```

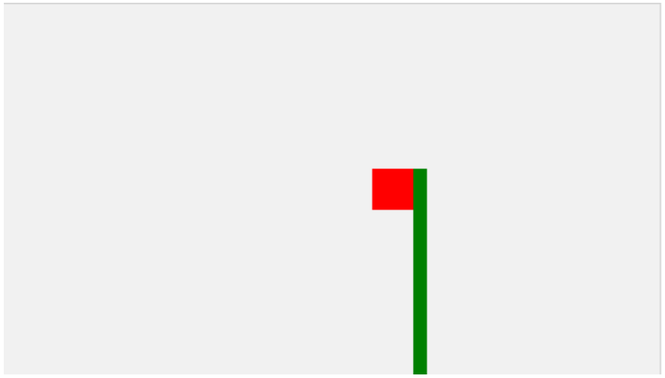
코드 소개



```
var myGamePiece;
var myObstacle;

function startGame() {
  myGamePiece = new component(30, 30, "red", 10, 120);
  myObstacle = new component(10, 200, "green", 300, 120);
  myGameArea.start();
}

function updateGameArea() {
  myGameArea.clear();
  myObstacle.update();
  myGamePiece.newPos();
  myGamePiece.update();
}
```

코드 소개



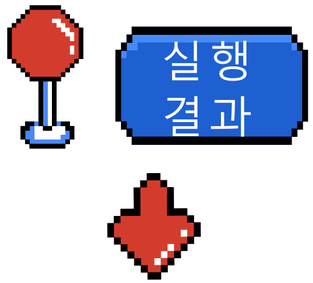


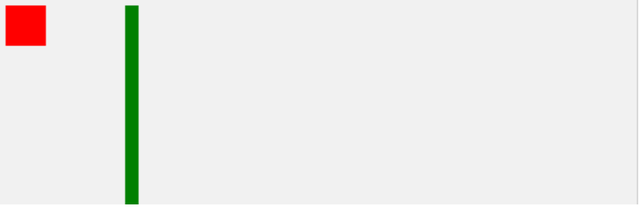
```
  stop : function() {
    clearInterval(this.interval);
  }
}
```

```
function updateGameArea() {
  if (myGamePiece.crashWith(myObstacle)) {
    myGameArea.stop();
  } else {
    myGameArea.clear();
    myObstacle.update();
    myGamePiece.newPos();
    myGamePiece.update();
  }
}
```

```
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;
}
```

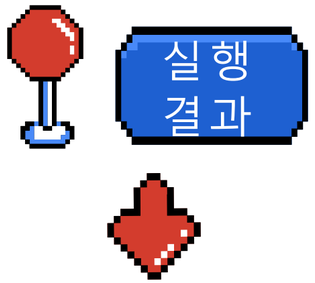
코드 소개





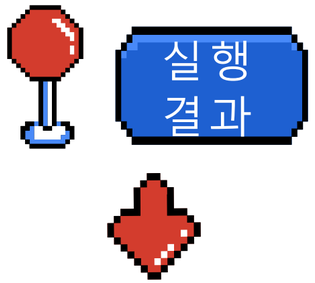
```
function updateGameArea() {
  if (myGamePiece.crashWith(myObstacle)) {
    myGameArea.stop();
  } else {
    myGameArea.clear();
    myObstacle.x += -1;
    myObstacle.update();
    myGamePiece.newPos();
    myGamePiece.update();
  }
}
```

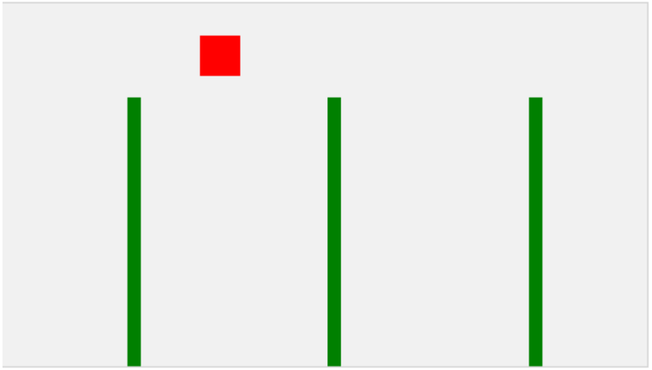
코드 소개




```
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;


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

코드 소개

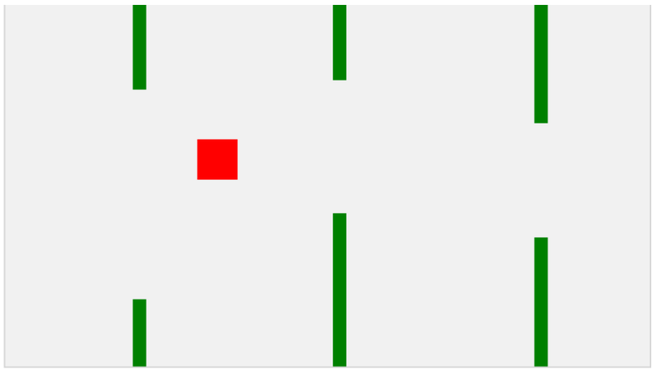




```
var myGamePiece;
var myObstacles = [];

function updateGameArea() {
  var x, y;
  for (i = 0; i < myObstacles.length; i += 1) {
    if (myGamePiece.crashWith(myObstacles[i])) {
      myGameArea.stop();
      return;
    }
  }
  myGameArea.clear();
  myGameArea.frameNo += 1;
  if (myGameArea.frameNo == 1 || everyinterval(150)) {
    x = myGameArea.canvas.width;
    y = myGameArea.canvas.height - 200
    myObstacles.push(new component(10, 200, "green", x, y));
  }
  for (i = 0; i < myObstacles.length; i += 1) {
    myObstacles[i].x += -1;
    myObstacles[i].update();
  }
  myGamePiece.newPos();
  myGamePiece.update();
}
```

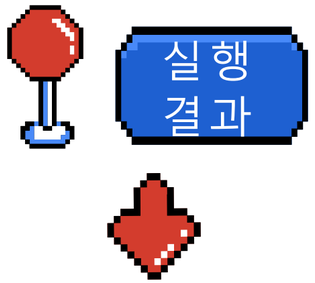
코드 소개

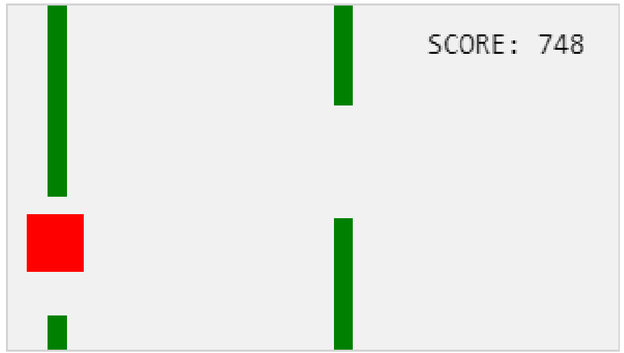
실행 결과



```
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));
}
```

코드 소개

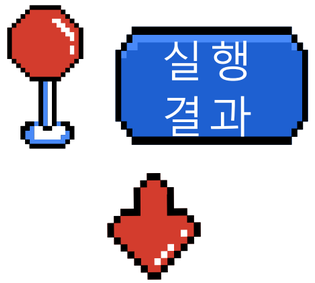


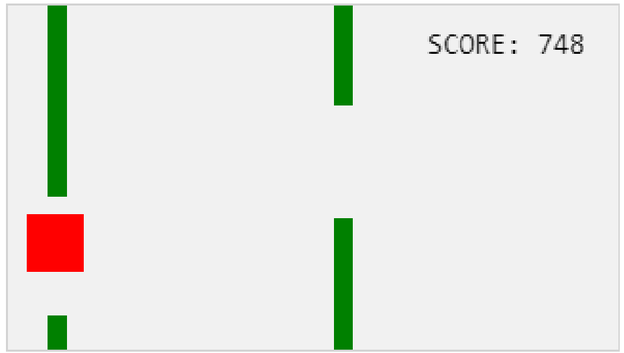



```
var myGamePiece;
var myObstacles = [];
var myScore;

function startGame() {
  myGamePiece = new component(30, 30, "red", 10, 160);
  myScore = new component("30px", "Consolas", "black", 280, 40, "text");
  myGameArea.start();
}
```

```
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.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 {
      ctx.fillStyle = color;
      ctx.fillRect(this.x, this.y, this.width, this.height);
    }
  }
...
}
```

코드 소개






```
function updateGameArea() {
  var x, height, gap, minHeight, maxHeight, minGap, maxGap;
  for (i = 0; i < myObstacles.length; i += 1) {
    if (myGamePiece.crashWith(myObstacles[i])) {
      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].speedX = -1;
    myObstacles[i].newPos();
    myObstacles[i].update();
  }
  myScore.text = "SCORE: " + myGameArea.frameNo;
  myScore.update();
  myGamePiece.newPos();
  myGamePiece.update();
}
```

감사합니다!

발표를 종료하시겠습니까?