

휴먼컴퓨터 인터페이스

과제 #0. HTML Canvas 연습

2020.04.08

컴퓨터 소프트웨어 학과

2016726009 | 임현우

01

요구조건과 제약조건 만족도 및 추가 구현사항 Summary



	직선	원	사각형	다각형	베지어곡선	텍스트	비트맵 이미지	추가기능
구현여부	○	○	○	○	○	○	○	○ (키보드 입력)
객체 수	모눈의 가로줄, 세로줄 개수	1	사각형 입력 명령 횟수 만큼	5	베지어곡선 입력 명령 횟수 만큼	6	2	

① 직선

구현 방법

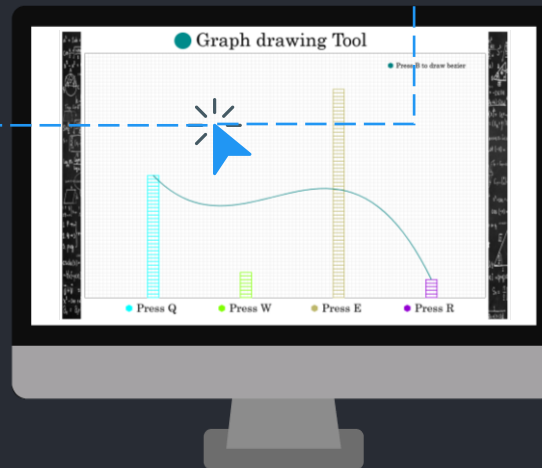
//30.5 = margin of grid

```
function drawGrid(context, color, stepx, stepy){
    context.strokeStyle = color;
    context.lineWidth = 0.5;

    for(var i=stepx+50.5; i<context.canvas.width-50.5; i+= stepx){
        context.beginPath();
        context.moveTo(i,60.5);
        context.lineTo(i, context.canvas.height-60.5);
        context.stroke();
    }

    for(var i=stepy+50.5; i<context.canvas.height-50.5; i+=stepy){
        context.beginPath();
        context.moveTo(60.5,i);
        context.lineTo(context.canvas.width-60.5,i);
        context.stroke();
    }
}
```

결과



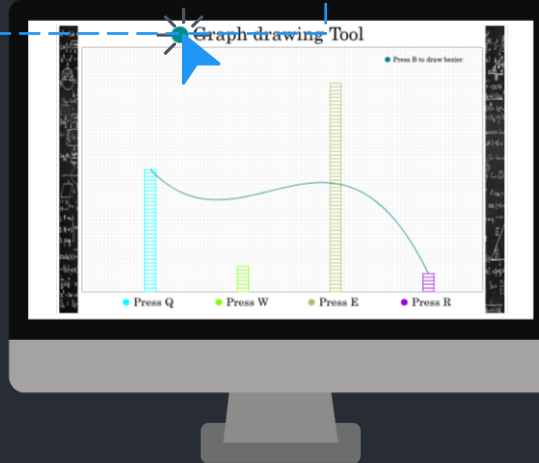
직선(모눈)

강의자료의 drawGrid 함수를 변형해 canvas 전체가 아닌 필요한 부분만 직선 객체를 이용해 모눈을 그리도록 하였다.

② 원

```
function drawHeadLineCircle(){  
  context.fillStyle = 'darkcyan';  
  context.beginPath();  
  context.arc(325,25,23,0,Math.PI*2);  
  context.fill();  
  context.fillStyle = 'black';  
}
```

구현 방법



결과

● Graph drawing Tool

Arc함수를 이용해 원을 그렸으며 fillStyle을 변경하는 라인이 있어 함수의 마지막 라인에 fillStyle을 다시 black으로 변경하는 코드를 삽입하였다.

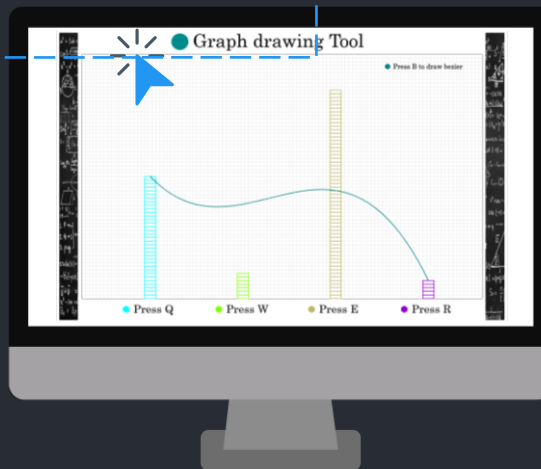
02

HTML Canvas 객체 종류 각각에 대한 구현 방법과 결과

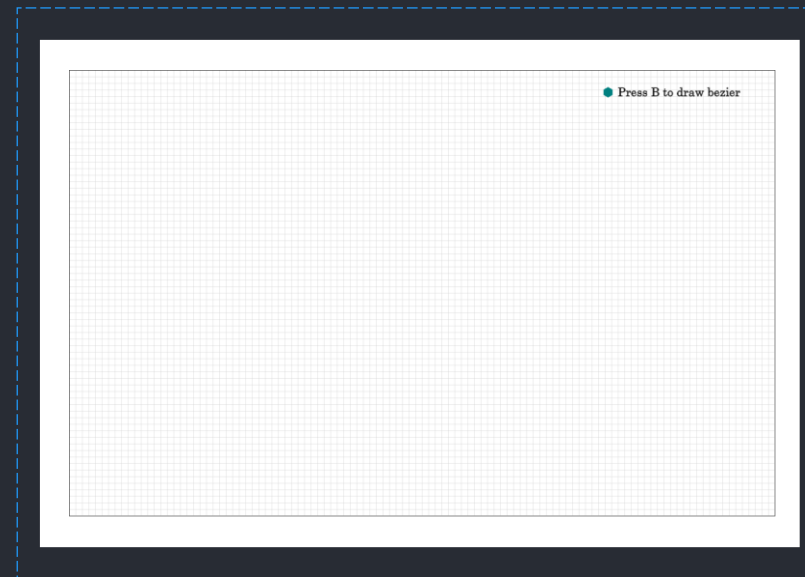
③ 사각형

구현 방법

```
//u must call stroke or fill after this function called  
function drawRect(xpos, ypos, width, height){  
    context.beginPath();  
    context.rect(xpos,ypos,width,height);  
}  
drawRect(60.5, 60.5, canvas.width-120, canvas.height-120);
```



결과



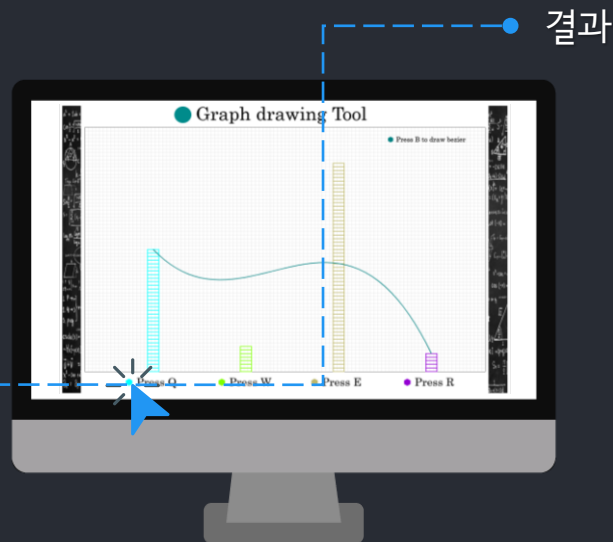
모눈의 외곽선을 강조하기 위해서 테두리가 검은색인 사각형객체를 draw하였다.

④ 다각형

구현 방법

```
function drawPolygon(posx, posy, rad, sides,color) {
  var angleDiv = 2*Math.PI/sides;
  var angle = 0;

  context.beginPath();
  context.moveTo(posx, posy - rad);
  for(var i=0; i<=sides; i++) {
    context.lineTo(posx + rad*Math.cos(angleDiv*i + Math.PI/2),
                  posy - rad*Math.sin(angleDiv*i + Math.PI/2));
  }
  context.fillStyle = color;
  context.fill();
  context.fillStyle = 'black';
}
```



Press Q

Press W

Press E

Press R

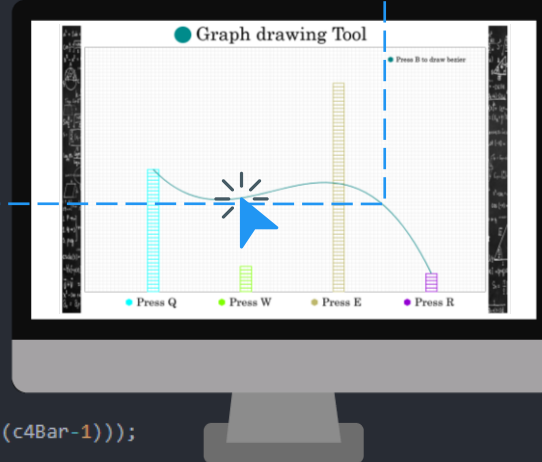
Press B to draw bezier

중심의 x,y좌표, 반지름, 변 개수, 색을 입력 받으면 다각형을 그리는 함수를 작성하였다. Canvas의 y축의 방향이 일반적인 좌표평면의 반대이므로 sin, cos함수를 이용해 좌표를 정할때 Math.PI/2를 더해 주어서 함수를 완성하였다.

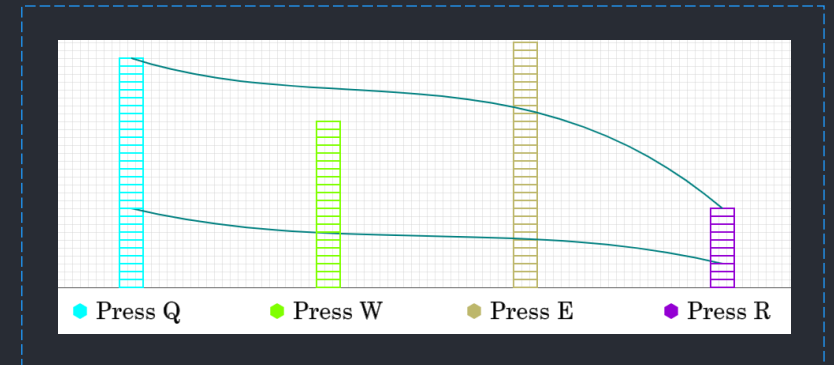
⑤ 베지어 곡선

구현 방법

```
if(keyValue == 'b'){  
    context.lineWidth = 2;  
    context.strokeStyle = 'teal';  
    context.beginPath();  
    context.moveTo(245,canvas.height-70-(10*(c1Bar-1)));  
    context.bezierCurveTo(495,canvas.height-70-(10*(c2Bar-1)),745,  
        canvas.height-70-(10*(c3Bar-1)),995,canvas.height-70-(10*(c4Bar-1)));  
    context.stroke();  
    context.strokeStyle = 'black';  
}
```



결과



키입력이 발생하면 양 끝 포인트와 컨트롤 포인트를 막대그래프의 끝점으로 갖는 베지어 곡선을 그리도록 하였다.

⑥ 텍스트

```
function drawText(text, xpos, ypos,font){  
    context.font = font;  
    context.fillText(text,xpos,ypos);  
    context.strokeText(text,xpos,ypos);  
    context.font = '10px century';  
}
```

```
function drawHeadline(){  
    drawText('Graph drawing Tool', 360,40,'50px century');  
}
```

구현 방법

결과

Graph drawing Tool

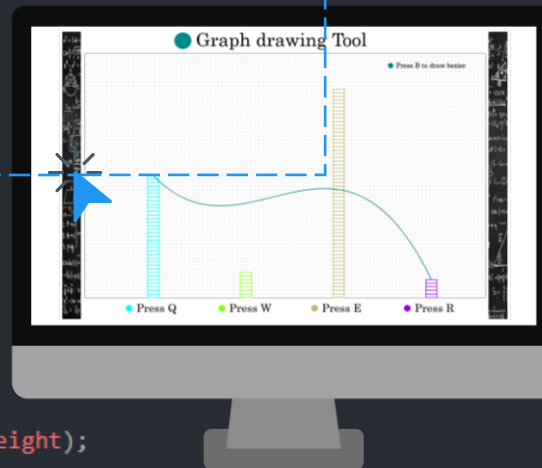


입력할 텍스트, x,y 위치, 폰트를 인자로 받는 drawText함수를 정의하여 텍스트를 그리도록 하였다.

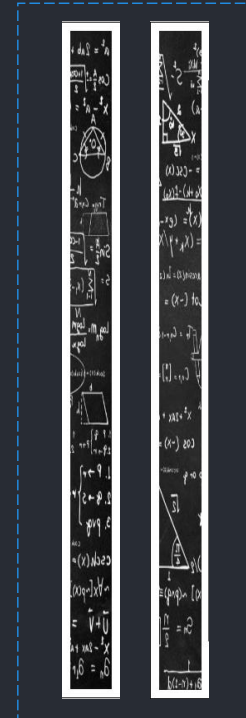
⑦ 비트맵 이미지

```
var leftImage = new Image();
leftImage.src = 'resource/left.jpeg';
leftImage.onload = function(e){
    context.drawImage(leftImage,0,0,50.5,canvas.height);
}
var rightImage = new Image();
rightImage.src = 'resource/right.jpeg';
rightImage.onload = function(e){
    context.drawImage(rightImage,canvas.width-50.5,0,50.5,canvas.height);
}
```

구현 방법



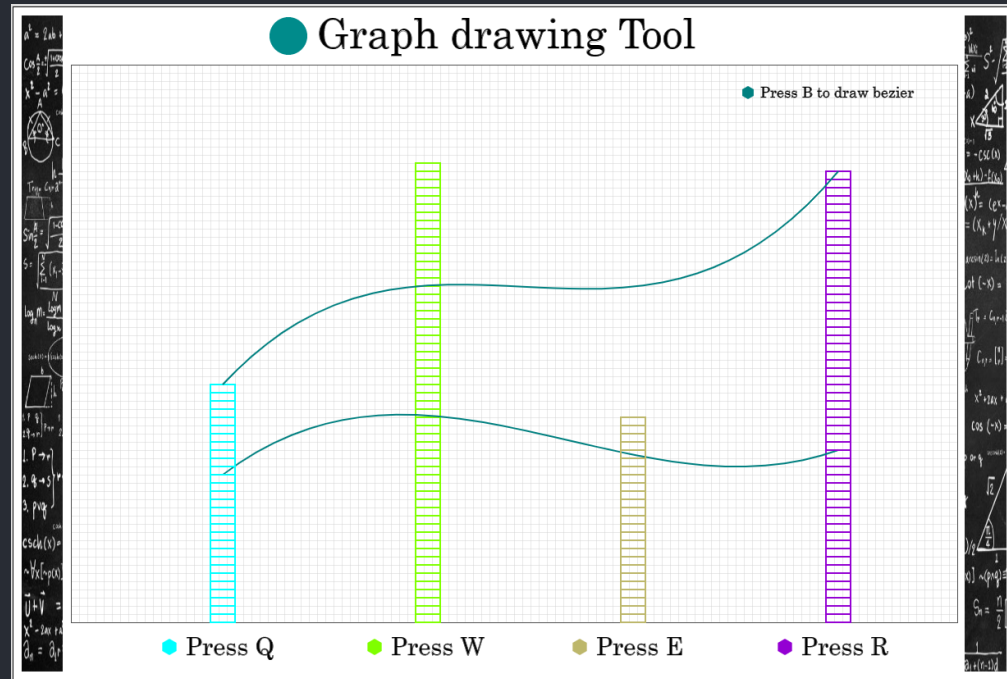
결과



강의자료에 포함 되어있는 함수를 변형해 하부 폴더에 있는 jpeg 파일을 불러올 수 있도록 하였고, canvas의 양 끝에 이미지를 불러오도록 하였다.

■ Key event를 활용한 막대그래프 그리기

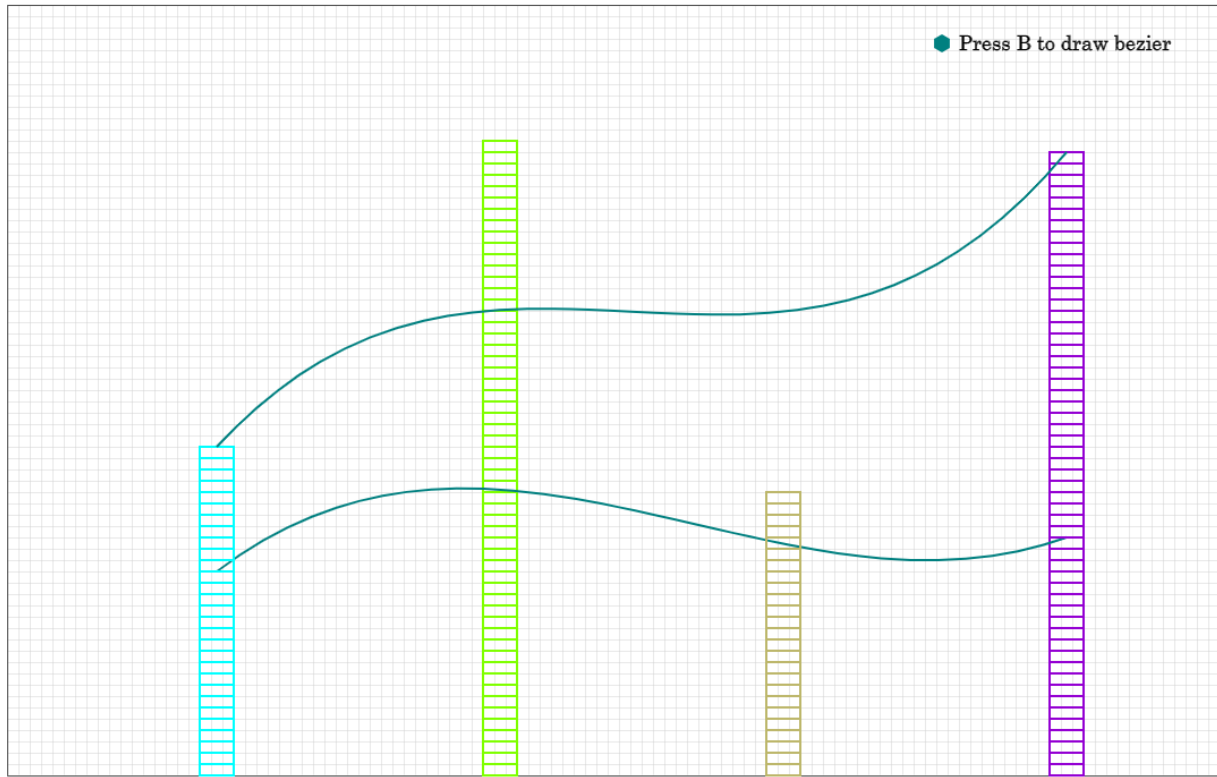
```
document.addEventListener('keydown', (event) => {
  const keyValue = event.key;
  if(keyValue == 'q'){
    context.lineWidth = 2;
    context.strokeStyle = 'aqua';
    drawRect(230, canvas.height-70-(10*c1Bar), 30, 10);
    context.stroke();
    c1Bar++;
  }
  if(keyValue == 'w'){
    context.lineWidth = 2;
    context.strokeStyle = 'chartreuse';
    drawRect(480, canvas.height-70-(10*c2Bar), 30, 10);
    context.stroke();
    c2Bar++;
  }
  if(keyValue == 'e'){
    context.lineWidth = 2;
    context.strokeStyle = 'darkkhaki';
    drawRect(730, canvas.height-70-(10*c3Bar), 30, 10);
    context.stroke();
    c3Bar++;
  }
  if(keyValue == 'r'){
    context.lineWidth = 2;
    context.strokeStyle = 'darkviolet';
    drawRect(980, canvas.height-70-(10*c4Bar), 30, 10);
    context.stroke();
    c4Bar++;
  }
  if(keyValue == 'b'){
    context.lineWidth = 2;
    context.strokeStyle = 'teal';
    context.beginPath();
    context.moveTo(245, canvas.height-70-(10*(c1Bar-1)));
    context.bezierCurveTo(495, canvas.height-70-(10*(c2Bar-1)), 745,
      canvas.height-70-(10*(c3Bar-1)), 995, canvas.height-70-(10*(c4Bar-1)));
    context.stroke();
    context.strokeStyle = 'black';
  }
}, false);
```



EventListener를 등록하여 'keydown'이벤트 발생시 막대 그래프 길이를 늘여가도록 하였다. Q,W,E,R을 입력 시 각각의 그래프 길이가 늘어나며, B키 입력 시 그려진 막대그래프를 이용해 베지어 곡선 또한 출력하도록 하였다.

● Graph drawing Tool

● Press B to draw bezier



키보드의 Q, W, E, R, B 키를 이용해서
막대 그래프와 베지어 곡선을 그릴 수
있는 graph drawing tool을 제작 하
였다.

논의

- ① keyboard 이벤트를 활용해서 동적으로 그래프를 그리는 데에 성공하였다. 하지만, 막대 그래프의 길이를 줄어둘게 하는 기능을 추가하는 과정에서 코드의 복잡성이 높아지고 많은 객체들을 일일이 추가하는 경우가 발생해 추가하지 않은 버전으로 과제를 제출하게 되었다.
- ② 자체적인 평가를 하자면 과제에서 요구하는 모든 요건을 충족하였으며, 동적인 콘텐츠(키보드 입력)를 제작하였기 때문에 양호한 수준이라고 보여진다.
- ③ 향후 개선점
- 막대 그래프의 길이를 줄어둘게 하는 키 입력 추가
 - 막대 그래프 개수의 증가 및 감소 기능 추가
 - 막대 그래프 하단의 Press Q, Press W, Press E, Press R 텍스트를 사용자가 변경할 수 있도록 하는 기능 추가



감사합니다