

## <컴퓨터 그래픽스 과제2>

AI학과 2143993 백수민

### Collision Detection 실습 보고서

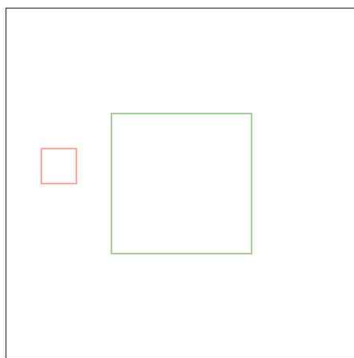
#### 1. 기본과제(3점)

##### box\_box\_collision() 함수 내부 완성하기

-> 빨간색 박스를 키보드를 이용해서 이동 시키는데, 초록색 박스와 겹쳐지면 두 박스의 안이 색으로 칠해진다. 다음은 코드 실행 결과를 보여준다.

#### [AIE305] Computer Graphics Lab

##### Collision Detection



학번 : 2143993

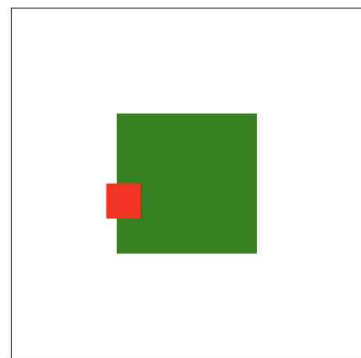
이름 : 백수민

기본과제 (3점)

- ☒ main\_230920.js 코드 확인
- ☐ 빨간색 박스의 위치를 키보드를 이용해서 이동시켜보기
- ☐ box\_box\_collision() 함수 내부를 완성하기

#### [AIE305] Computer Graphics Lab

##### Collision Detection



학번 : 2143993

이름 : 백수민

기본과제 (3점)

- ☒ main\_230920.js 코드 확인
- ☒ 빨간색 박스의 위치를 키보드를 이용해서 이동시켜보기
- ☒ box\_box\_collision() 함수 내부를 완성하기

실행에 사용된 draw\_image() 함수와 box\_box\_collision() 함수 코드 내용.

```
35
36 function draw_image() {
37     let isFill = false; //충돌하는지 안 하는지
38     if (box_box_collision(boxPts[0], boxPts[1], boxPts[2], boxPts[3]))
39         isFill = true;
40
41     ctx.strokeStyle = "green"; //선
42     ctx.fillStyle = "green"    //도형 내부
43     draw_box(boxPts[0], boxPts[1], isFill)
44
45     ctx.strokeStyle = "red"
46     ctx.fillStyle = "red"
47     draw_box(boxPts[2], boxPts[3], isFill)
48 }
49
50 function box_box_collision(pMin, pMax, qMin, qMax) {
51     if (
52         pMin.x < qMax.x && pMax.x > qMin.x && pMin.y < qMax.y && pMax.y > qMin.y
53     )
54         return true;
55     return false
56 }
57
```

## 2. 도전과제(1점)

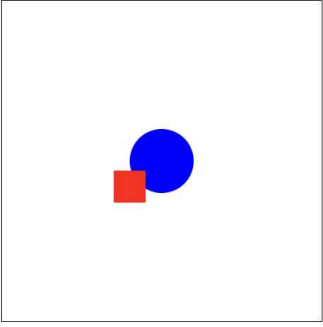
### 원과 박스의 충돌을 감지하는 코드를 작성하여 테스트해보기

-> 원과 박스가 겹쳐지면 두 도형이 색으로 칠해짐으로써 충돌을 감지함.

```
js > main_230920.js > @ box_box_collision
53 // draw_circle(circleCtr,50,false);
54
55
56 function draw_challenge() {
57   let isFill = false; //충돌하는지 안 하는지
58
59   let circleCtr = new THREE.Vector2(250,250);
60   let rad = 50;
61
62   if (circle_box_collision(circleCtr, rad, boxPts[2], boxPts[3]))
63   {
64     isFill = true;
65   }
66
67   ctx.strokeStyle = "blue"
68   ctx.fillStyle = "blue"
69   draw_circle(circleCtr, rad, isFill);
70
71   ctx.strokeStyle = "red"
72   ctx.fillStyle = "red"
73   draw_box(boxPts[2], boxPts[3], isFill)
74
75 }
76
77 function box_box_collision(pMin, pMax, qMin, qMax) {
78   if (
79     pMin.x < qMax.x && pMax.x > qMin.x && pMin.y < qMax.y && pMax.y > qMin.y
80   )
81     return true;
82   return false
83 }
84
85 function circle_box_collision(circle, r, qMin, qMax) {
86   let circle_x = Math.max(qMin.x, Math.min(circle.x, qMax.x));
87   let circle_y = Math.max(qMin.y, Math.min(circle.y, qMax.y));
88   //거리 계산
89   let dis_x = circle.x - circle_x;
90   let dis_y = circle.y - circle_y;
91   let dis_sq = dis_x ** 2 + dis_y ** 2;
92
93   if (dis_sq <= r ** 2)
94   {
95     return true;
96   }
97
98   return false
99 }
100
```

**[AIE305] Computer Graphics Lab**

Collision Detection



학번 : 2143993  
이름 : 백수민  
기본과제 (3점)  
main\_230920.js 코드 확인  
별간색 박스의 위치를 키보드를 이용해서 이동시켜보기  
box\_box\_collision() 함수 내부를 완성하기  
도전과제 (1점)  
원과 박스의 충돌을 감지하는 코드를 작성하여 테스트해보기

### 원의 중심을 사각형

Math.max(qMin.x, Math.min(circle.x, qMax.x)) 코드는

circle.x를 qMin.x와 qMax.x 사이의 값으로 제한하기 위한 코드로,

circle.x < qMin.x => qMin.x가 선택

circle.x > qMax.x => qMax.x가 선택

두 값 중에서 가장 큰 값을 반환한다. y도 동일하게 진행한다.

### 거리 계산

원의 중심과 사각형 내의 가장 가까운 점 사이의 수평 및 수직 거리로 dis\_x와 dis\_y를 구한다. 그리고 피타고리스 정리를 활용해 제곱값의 합을 dis\_sq로 선언한다.

### 원의 반지름과 거리 비교

dis\_sq는 원의 중심과 사각형 내의 가장 가까운 점 사이의 거리의 제곱 값으로 원의 반지름의 제곱값과 비교한다.

만약 dis\_sq가 반지름의 제곱값 보다 작거나 같다 => 원과 사각형이 충돌이라고 간주하여 True를 반환한다.