OBB 계산

1. 사각형 두 개의 Up, Right 방향 벡터를 구함
2. 거리를 구한 원점끼리의 거리(벡터값으로 가지고 있음)
3. 충돌에 사용되는 사각형 2개의 UP과 Right 방향 벡터에 \* Size/2 값을 구함
4. 연산 1

C = 절댓값(거리,r1의 Right 방향벡터와의 내적 계산 결과)

A = 절댓값((r2Up,r1의 Right 방향벡터의 내적 절댓값)+(r2Right,r1의 Right 방향벡터 내적 절댓값))

B = r1 x축의 길이/2

1. 연산 2

C = 절댓값(거리,r1의 Up 방향 벡터 내적 계산 결과)

A = 절댓값((r2Up,r1의 Up 방향벡터와의 내적 절댓값)

+(r2Right,r1의 Up 방향벡터와의 내적 절댓값))

B = r1 y축의 길이

1. 연산 3

C = 절댓값(거리,r2의 Right 방향 벡터 내적 계산 결과)

A = 절댓값((r1Up,r2의 Right 방향벡터의 내적 절댓값)

+(r1Right,r2의 Up 방향벡터의 내적 절댓값))

B = r2 x축의 길이

1. 연산 4

C = 절댓값(거리,r2의 Up 방향 벡터 내적 계산 결과)

A = 절댓값((r1Up,r2의 Up 방향벡터와의 내적 절댓값)

+(r1Right,r2의 Up 방향벡터와의 내적 절댓값))

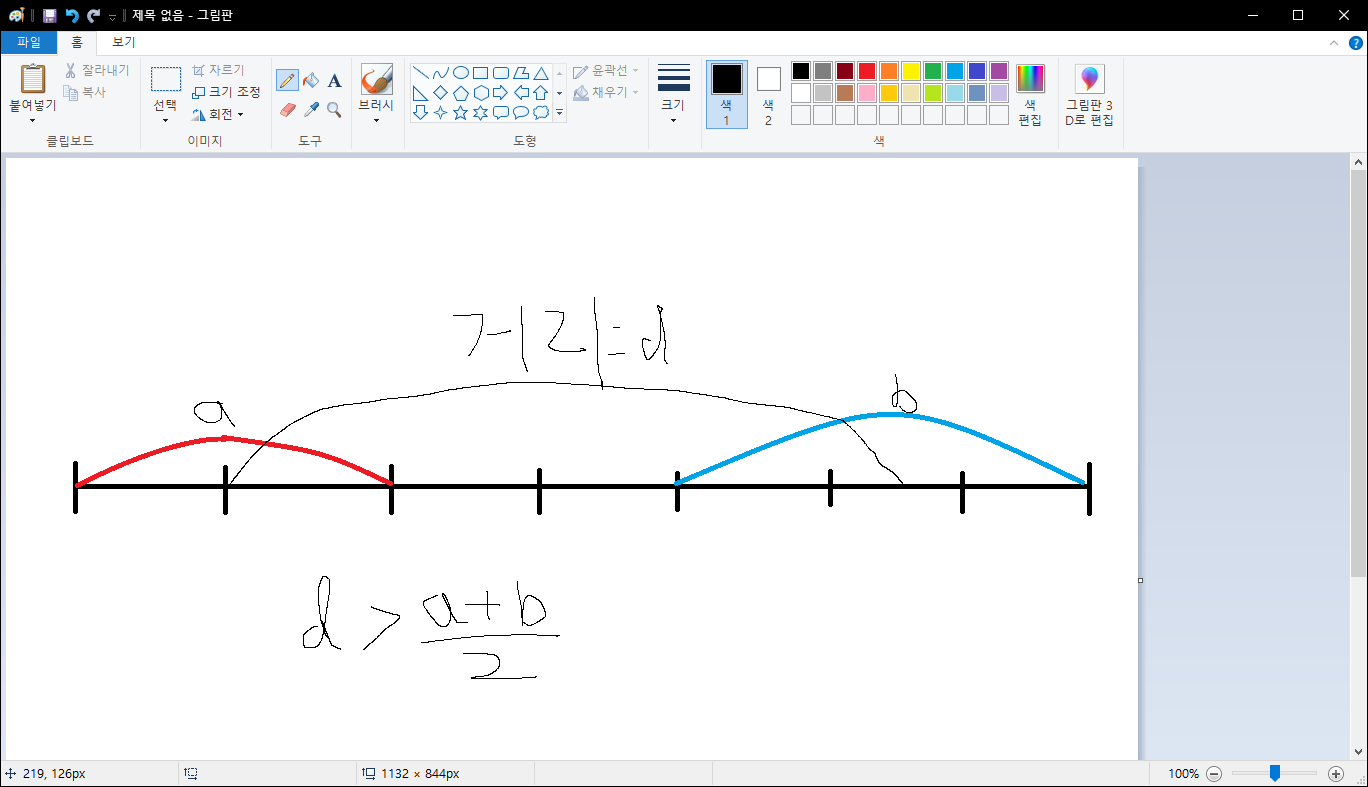
B = r2 y축의 길이

이렇게 보면 위의 식이 뭘 의미하는 것인지 하나도 모를 수도 있다.

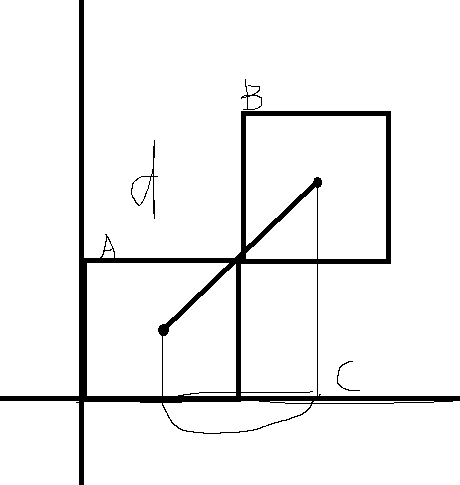
하지만 조금씩 나눠서 OBB 충돌을 이해해보면 이해할 수 있다.

만약 1차원에서 길이가 다른 두 선분의 충돌 여부를 파악하기 위해서 어떤 식으로 생각해야 하는지 생각해보면

A=2 B=3이라고 하면 아래의 이미지처럼 충돌을 확인할 수 있는 식이 나온다.



그렇다면 이제 차원을 한 단계 올려서 생각해보자



이제 다시 위에 있는 식을 요약해보면

각 연산의 의미는 투영할 기준이 되는 방향벡터를 정해놓은 것으로 생각할 수 있고 다각형의 방향 벡터를 임의의 축으로 연산을 진행하고 진행 과정에서 위와 같은 C>A+B라는 결과에 도달하면 false를 반환하여 충돌되지 않음을 확인하는 방식이다.

OBB 계산

1. 사각형 두개의 Up, Right 방향 벡터를 구함
2. 거리를 구함 원점끼리의 거리(벡터값으로 가지고 있음)
3. 충돌에 사용되는 사각형 2개의 UP과 Right 방향 벡터에 대응하는 변의 길이/2를 구함
4. 연산 1

C = 거리를 r1의 Right 방향벡터를 축으로 투영한 거리값

A = r2Up, r2Right을 r1의 Right 방향벡터를 축으로 투영한 거리값

B = r1의 x의 투영 거리

1. 연산 2

C = 거리를 r1의 up 방향벡터를 축으로 투영한 거리값

A = r2Up, r2Right을 r1의 Up 방향벡터를 축으로 투영한 거리값

B = r1의 y의 투영 거리

1. 연산 3

C = 거리를 r2의 Right 방향벡터를 축으로 투영한 거리값

A = r1Up, r1Right을 r2의 Right 방향벡터를 축으로 투영한 거리값

B = r2의 x의 투영 거리

1. 연산 4

C = 거리를 r2의 up 방향벡터를 축으로 투영한 거리값

A = r1Up, r1Right을 r2의 Up 방향벡터를 축으로 투영한 거리값

B = r2의 y의 투영 거리

이렇게 이해하면 이제 OBB 방식 과정을 좀더 쉽게 이해할수있다.

참고 링크 : https://justicehui.github.io/other-algorithm/2018/06/23/OBB/