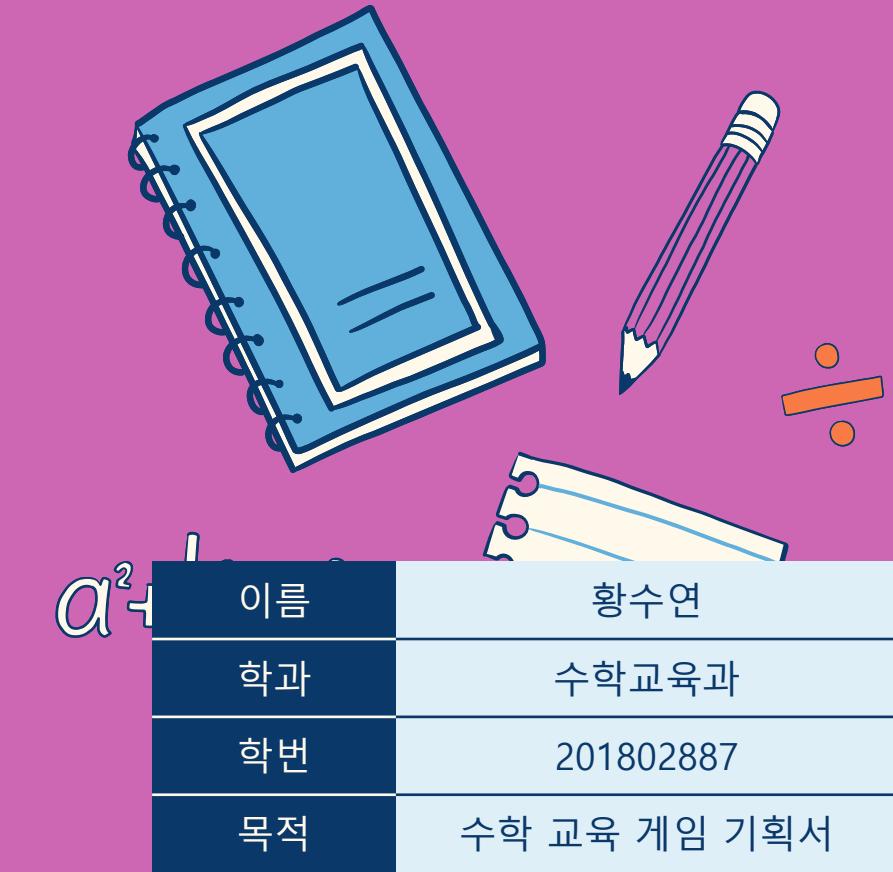


$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# 벡터 던지기

2015 개정 교육과정  
기하-(2)평면벡터 단원

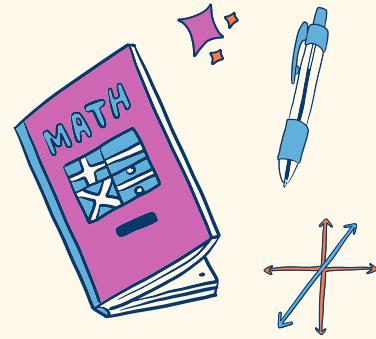


# Document History

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# 목차

1. 게임 개요
  2. 주요 타겟
  3. 게임 플레이 방법
  4. 게임에 필요한 벡터 이론
  5. UI 및 시스템 기획
- 
2. 주요 타겟
  2. 주요 타겟
  2. 주요 타겟
  2. 주요 타겟
  2. 주요 타겟



# 1. 게임 개요

## 1) 기획서 목적

- 수학 교육 게임 개발을 위해 개발 목적, 플레이 방법, 세부 기획, 개발 일지 등을 작성한다.

## 2) 게임 개요

- 수학 기하 단원의 벡터의 이해를 돋기 위한 교육용 소프트웨어 게임
- 화살표 모양의 그림을 제시된 벡터만큼 이동시켜 화면의 끝에 도달시켜서 점수를 얻는 게임
- 장르: 교육용, 아케이드, 슈팅
- 플랫폼: AOS (iOS는 맥북이 없는 관계로 당장 구현 불가)
- 그래픽: 2D 캐주얼 그래픽

## 2. 주요 타겟

- 벡터단원의 이론 학습 이후 **매체 및 도구 활용 학습**을 선택하는 학생
- **정보 처리 능력**을 함양하기 위해 교육용 게임을 조작 및 탐구 활동을 통해 수학의 개념과 원리를 이해하려는 학생
- **태도 및 실천 능력**을 함양하기 위해 수학 학습에 적극적으로 참여하고, 도전하여 학습 동기를 되새기려는 학생
- **의사소통 능력**을 수정할 것함양하기 위해 수학 학습에 적극적으로 참여하고, 도전하여 학습 동기를 되새기려는 학생
- 수준별 학습을 진행하여, 내용의 깊이에 있어 더 깊숙이 알고 싶은 학생일 경우,

**추론 능력**을 함양하기 위해

게임에서 벡터가 어떻게 사용되었는지,

관찰과 탐구과정을 통해 귀납, 유추 등

개연적 추론을 통해 수학적 사실을

적절한 근거에 기초하여 정당화하려는 학생

### 3. 게임 플레이 방법

1) 하단에서 벡터를 던진다.

- ① 게임 화면의 하단에는 세 군데 화살표를 던질 곳이 있다.
- ② 화살표를 던질 때는 세 군데 중 하나를 선택하여 던질 수 있다.

2) 상단에는 화살표(벡터)나 결과값이 숫자가 되는 타일이 현재, 다음, 다다음번에 나올 것이 띄워진다.

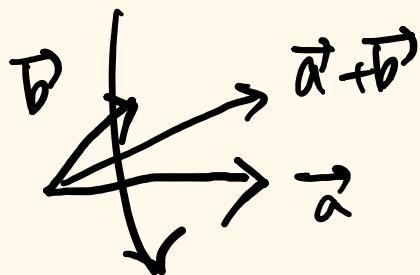
- ① 게임 화면의 상단에는 벡터, 벡터의 크기, 벡터의 내적 등이 나온다.
- ② 현재 타일, 다음타일, 다다음타일까지 보인다.
- ③ 결과값이 벡터인 경우, 화살표를 던진다.
- ④ 결과값이 숫자인 경우, 화살표에 가중치를 넣는다. (어떤 화살표에 3만큼의 가중치)

3) 화살표를 화면의 끝에 밀어 넣어 가중치만큼의 점수를 획득한다.

- ① 화살표를 벡터의 방향만큼 밀어 넣는다.
- ② 이를 반복하여 화면의 끝에 도달하게 하면 점수를 얻을 수 있다.
- ③ 게임 화면에 배치된 화살표에 숫자를 드래그 앤 드롭하면 화살표에 가중치가 들어간다.
- ④ 화살표의 가중치만큼 화살표가 화면의 끝에 도달했을 경우 점수를 얻을 수 있다.

#### 4. 게임에 필요한 벡터 이론

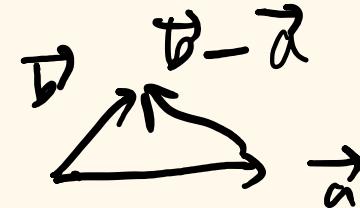
$\vec{a}$ : 벡터  $a$



$\vec{b}$ : 벡터  $b$



$|\vec{a}|, |\vec{b}|$ : 각 벡터의 크기



: 빡빡의 평행성.

(+, +) →

(-, +) ↗

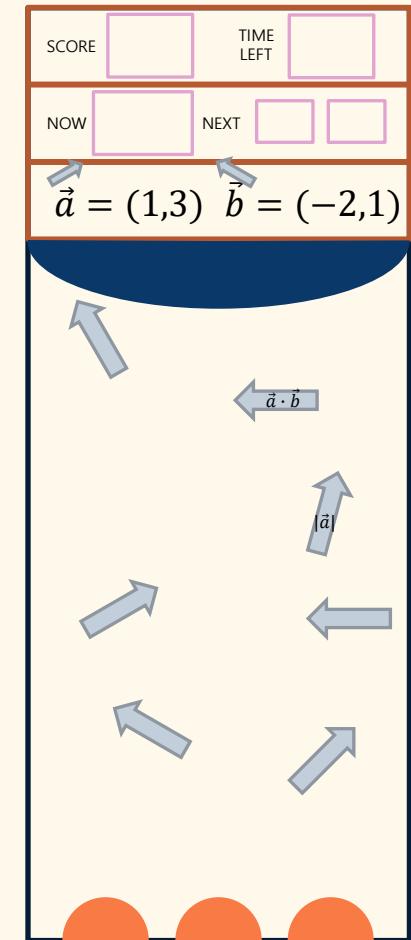
$\vec{a} \cdot \vec{b}$ :  $\vec{a}$ 와  $\vec{b}$ 의 내적  $\rightarrow |\vec{b}| \cos \theta$  방향 cosine 가능.

$$(a_1, a_2) (b_1, b_2) \rightarrow a \cdot b = a_1 b_1 + a_2 b_2$$

## 5. UI 및 시스템 기획

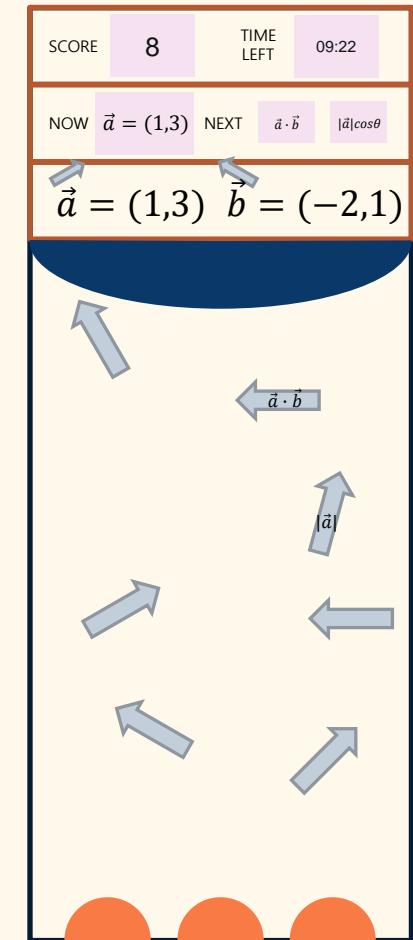
### 1) UI 배치 러프안

- ① 스마트폰으로 구현하기 위해 22:9 비율의 2640 x 1080 해상도를 기준으로 기획한다.
- ② 배경은 흰색에 가까운 계열로 설정한다.
- ③ 버튼은 검은색에 가까운 색상이며, 각 버튼에 따라 세부적으로 설정한다.
- ④ 글씨는 기본적으로 맑은 고딕에 검은색이며, 크기는 세부적으로 따로 설정한다.



## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

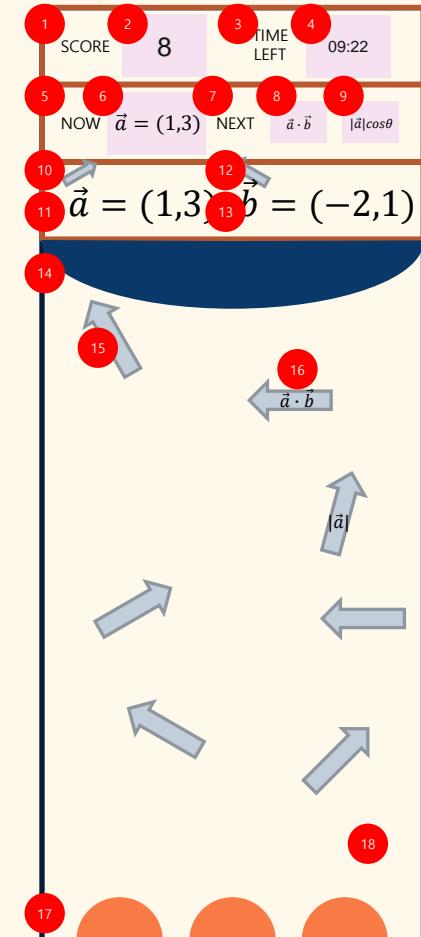


## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

UI 설명 표

NO	UI 이름	UI 타입 (컴포넌트)	설명
1	SCORE	문자 (String)	<ul style="list-style-type: none"> <li>획득한 점수를 알아볼 수 있도록 실제 점수 옆에 “SCORE”라고 표시</li> </ul>
2	점수	문자 (String)	<ul style="list-style-type: none"> <li>획득한 점수를 int형으로 표시</li> </ul>
3	TIME LEFT	문자 (String)	<ul style="list-style-type: none"> <li>남은 시간을 알아볼 수 있도록 실제 남은 시간 옆에 “TIME LEFT”라고 표시</li> </ul>
4	남은 시간	문자 (String)	<ul style="list-style-type: none"> <li>남은 시간을 타이머 기능을 사용하여 표시</li> </ul>
5	NOW	문자 (String)	<ul style="list-style-type: none"> <li>현재 타일을 알아볼 수 있도록 실제 현재 타일 옆에 “NOW”라고 표시</li> </ul>
6	현재 타일	문자 (String)	<ul style="list-style-type: none"> <li>벡터 수식으로 이루어진 현재 타일</li> </ul>

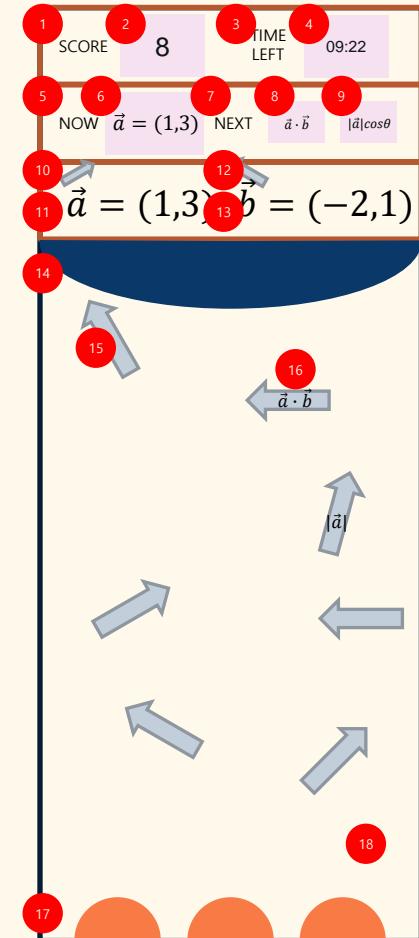


## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

UI 설명 표

NO	UI 이름	UI 타입 (컴포넌트)	설명
7	NEXT	문자 (String)	<ul style="list-style-type: none"> <li>다음 타일을 알아볼 수 있도록 실제 다음 타일 옆에 “NEXT”라고 표시</li> </ul>
8	다음 타일	문자 (String)	<ul style="list-style-type: none"> <li>벡터 수식으로 이루어진 다음 타일</li> </ul>
9	다다음 타일	문자 (String)	<ul style="list-style-type: none"> <li>벡터 수식으로 이루어진 다다음 타일</li> </ul>
10 12	화살표	이미지 (Image)	<ul style="list-style-type: none"> <li>벡터의 방향(좌상향, 우상향)을 알려주는 화살표</li> </ul>
11	벡터 $\vec{a}$	문자 (String)	<ul style="list-style-type: none"> <li>현재 벡터 <math>\vec{a}</math>의 성분을 나타낸다.</li> </ul>
13	벡터 $\vec{b}$	문자 (String)	<ul style="list-style-type: none"> <li>현재 벡터 <math>\vec{b}</math>의 성분을 나타낸다.</li> </ul>

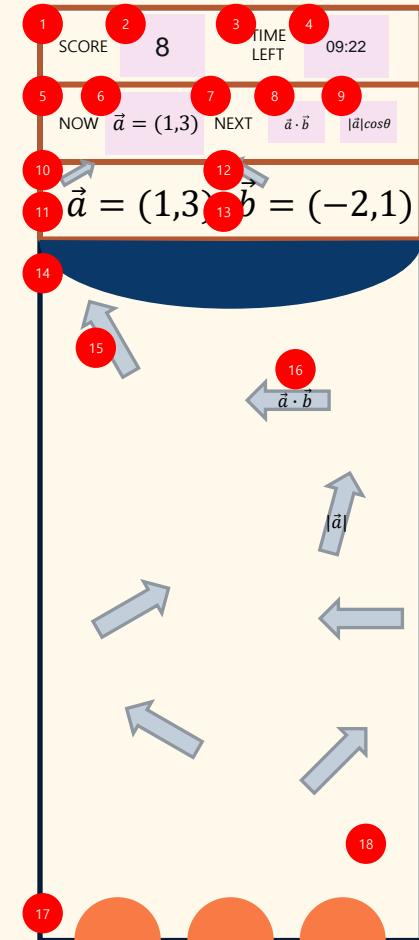


## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

UI 설명 표

NO	UI 이름	UI 타입 (컴포넌트)	설명
14	HOLE	이미지 (Image)	<ul style="list-style-type: none"> <li>화살표를 HOLE까지 밀어 넣으면 해당 화살표의 가중치에 따라 점수를 획득한다.</li> </ul>
15	화살표	이미지 (Image)	<ul style="list-style-type: none"> <li>타일의 결과값이 벡터일 경우, 발사되는 화살표 이미지</li> </ul>
16	가중치	문자 (String)	<ul style="list-style-type: none"> <li>타일의 결과값이 숫자일 경우, 화살표에 그 값을 드래그 앤 드롭하여 가중치가 화살표에 입력되고, 그 값을 나타내는 문자이다.</li> </ul>
17	발사 버튼	버튼 (Button)	<ul style="list-style-type: none"> <li>타일의 결과값이 벡터일 경우, 세 버튼 중 하나를 눌러 화살표를 벡터의 방향만큼 발사한다.</li> </ul>
18	배경 이미지	이미지 (Image)	<ul style="list-style-type: none"> <li>게임 화면의 배경 이미지</li> </ul>



## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (1) SCORE

- 획득한 점수를 알아볼 수 있도록 실제 점수 옆에 "SCORE"라고 표시
- 설정 내용

string ID	SCORE
출력 내용	SCORE
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (2) 점수

- 획득한 점수를 int형으로 표시한다.
- 설정 내용

int ID	score_int
출력 내용	`\${score_int}`
설정 내용	<ul style="list-style-type: none"><li>size: 14</li><li>색상: 검은색 계열</li><li>글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (3) TIME LEFT

- 남은 시간을 알아볼 수 있도록 실제 남은 시간 옆에 "TIME LEFT"라고 표시
- 설정 내용

string ID	TIME LEFT
출력 내용	TIME LEFT
설정 내용	<ul style="list-style-type: none"><li>size: 14</li><li>색상: 검은색 계열</li><li>글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (4) 남은 시간

- 남은 시간을 타이머 기능을 사용하여 표시
- 최초 씬에서 플레이 할 시간을 지정하고, 그 시간이 초 단위로 줄어들며 표시된다.
- 남은 시간이 0분 0초가 되면, 게임을 종료하고 엔딩 씬으로 전환된다.
- 설정 내용

int ID?	Timer(? -> 타이머 표시 방법 아직 모름)
출력 내용	00:00(?-> 타이머 표시 방법 아직 모름)
설정 내용	<ul style="list-style-type: none"><li>size: 20</li><li>색상: 검은색 계열</li><li>글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (5) NOW

- 현재 타일을 알아볼 수 있도록 실제 현재 타일 옆에 "NOW"라고 표시
- 설정 내용

string ID	NOW
출력 내용	NOW
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (6) 현재 타일

- 이번 텐의 수식을 알려준다.
- 수식은 평면 벡터 단원에 나오는 벡터들로 구성된다.
- 타일의 종류는 결과값이 벡터인 것과 숫자인 것 두 가지이다.
- 타일(수식)의 종류는 다음 페이지에서 상세 설명한다.
- 설정 내용

string ID	NOW_Tile
출력 내용	벡터 수식(다음 페이지에서 상세 설명)
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## (6) 정재 배열

나오는 모든 벡터의 타입과 종류



-  $\vec{a}$ : 우상향 벡터 (+, +)       $t: 1 \sim 9 \text{ char}$       -: - $1 \sim 9 \text{ char}$

-  $\vec{b}$ : 좌상향 벡터 (-, +)      Random. num (1, 9)

-  $\vec{a} + \vec{b}$ : 우상향 벡터 (+, +)      ↳ 1에서 9 까지의 char 중 난수 생성

-  $\vec{a} - \vec{b}$ : 좌상향 벡터 (-, +)

---

$|\vec{a}|$ :  $\vec{a}$ 의 크기

$\vec{a} \cdot \vec{b}$ : 벡터  $a$ 와 벡터  $b$  내적

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (7) NEXT

- 현재 타일을 알아볼 수 있도록 실제 현재 타일 옆에 "NEXT"라고 표시
- 설정 내용

string ID	NEXT
출력 내용	NEXT
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (8) 다음 타일

- 다음 턴의 수식을 알려준다.
- 수식은 평면 벡터 단원에 나오는 벡터들로 구성된다.
- 타일의 종류는 결과값이 벡터인 것과 숫자인 것 두 가지이다.
- 타일(수식)의 종류는 [해당 페이지](#)에서 상세 설명한다.
- 설정 내용

string ID	NEXT_Tile
출력 내용	벡터 수식( <a href="#">해당 페이지</a> 에서 상세 설명)
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (9) 다다음 타일

- 다다음 턴의 수식을 알려준다.
- 수식은 평면 벡터 단원에 나오는 벡터들로 구성된다.
- 타일의 종류는 결과값이 벡터인 것과 숫자인 것 두 가지이다.
- 타일(수식)의 종류는 [해당 페이지](#)에서 상세 설명한다.
- 설정 내용

string ID	NEXTNEXT_Tile
출력 내용	벡터 수식( <a href="#">해당 페이지</a> 에서 상세 설명)
설정 내용	<ul style="list-style-type: none"><li>• size: 14</li><li>• 색상: 검은색 계열</li><li>• 글자 굵게 처리</li></ul>

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (10) 화살표

- $\hat{a}$ 가 우상향 방향의 벡터임을 알려주는 이미지
- 배치는 (11) 벡터  $a$ 의 상단에 배치하며,  $\hat{a}$  부분의 바로 위에 배치한다.
- 각도는 지면 기준  $30^\circ$ , 즉,  $\theta = \frac{\pi}{6}$ 의 각도 정도로 배치한다.
- 정확히  $\theta = \frac{\pi}{6}$ 로 배치하지 않아도 무방하며, 플레이어가 우상향 방향의 벡터임을 알 수 있을 정도로만 기울여져 있으면 된다.
- 벡터  $a$ , 벡터  $b$ , 게임 플레이시 발사되는 화살표는 같은 이미지 파일을 사용한다.
- 설정 내용

	설정 내용
예시 이미지	<ul style="list-style-type: none"><li>• 흰색 바탕에 검은색 테두리인 화살표</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (11) 벡터 $\hat{a}$

- $\hat{a}$ 가 우상향 방향의 벡터임을 알려주는 이미지
- 배치는 (11) 벡터  $a$ 의 상단에 배치하며,  $\hat{a}$  부분의 바로 위에 배치한다.
- 각도는 지면 기준  $30^\circ$ , 즉,  $\theta = \frac{\pi}{6}$ 의 각도 정도로 배치한다.
- 정확히  $\theta = \frac{\pi}{6}$ 로 배치하지 않아도 무방하며, 플레이어가 우상향 방향의 벡터임을 알 수 있을 정도로만 기울여져 있으면 된다.
- 벡터  $a$ , 벡터  $b$ , 게임 플레이시 발사되는 화살표는 같은 이미지 파일을 사용한다.
- 설정 내용

	설정 내용
예시 이미지	<ul style="list-style-type: none"><li>• 흰색 바탕에 검은색 테두리인 화살표</li></ul> 

## (ii) 풀이

정수와 실수와 지수로 같은 값을 갖는다.

즉, 각각의 부호와 소수점 아래의 수와 소수점 위에 수를 갖는다.

따라서 a 값은 원소형 타입을 지닌 탑재되어며, 상수는 1~9 까지 char

타입에  $\vec{a} = ( , )$  + 내용 깊이, b 값은 a의 탑재 타입을 갖는  
여러 가지 가능하다.

이제 탑재 타입에  $\vec{a} +$  끝 때까지 유지되며,

별의 연산시 갖는 맥락(별의 a의 역할)으로 처리된다

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (12) 화살표

- $\vec{b}$ 가 좌상향 방향의 벡터임을 알려주는 이미지
- 배치는 (11) 벡터  $b$ 의 상단에 배치하며,  $\vec{b}$  부분의 바로 위에 배치한다.
- 각도는 지면 기준  $150^\circ$ , 즉,  $\theta = \frac{5\pi}{6}$ 의 각도 정도로 배치한다.
- 정확히  $\theta = \frac{5\pi}{6}$ 로 배치하지 않아도 무방하며, 플레이어가 좌상향 방향의 벡터임을 알 수 있을 정도로만 기울여져 있으면 된다.
- 벡터  $a$ , 벡터  $b$ , 게임 플레이시 발사되는 화살표는 같은 이미지 파일을 사용한다.
- 설정 내용

	설정 내용
예시 이미지	<ul style="list-style-type: none"><li>• 흰색 바탕에 검은색 테두리인 화살표</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (13) 벡터b

- $\hat{a}$ 가 우상향 방향의 벡터임을 알려주는 이미지

별의 연상시

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (14) HOLE

- 화살표를 HOLE까지 밀어 넣으면 해당 화살표의 가중치에 따라 점수를 획득한다.
- 화살표와 HOLE이 충돌 판정이 나면, 화살표는 사라진다.
- 가중치는 사라진 화살표에 기록된 벡터 수식을 현재  $\vec{a}, \vec{b}$ 에 따라 계산하며, 그 값을 SCORE에 더한다.
- 설정 내용

	설정 내용
예시 이미지	<ul style="list-style-type: none"><li><b>활꼴 모양</b></li><li>어두운 남색으로, 포켓 당구대의 홀과 같은 느낌으로 제작한다.</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (15) 화살표

- 게임 화면에 배치되는 화살표이다.
- 이 화살표는 현재 타일의 결과값이 벡터일 경우 발사되는 화살표이다.
- 화살표가 발사되는 장소는 "(17) 발사 버튼"이며, 이 곳 중 한 곳에서 발사된다.
- 화살표끼리 서로 충돌하며, 서로 밀쳐내는 물리 작용을 가지고 있다.
- 화살표와 HOLE이 충돌 판정이 나면, 화살표는 사라진다.
- 벡터  $a$ , 벡터  $b$ , 게임 플레이시 발사되는 화살표는 같은 이미지 파일을 사용한다.
- 설정 내용

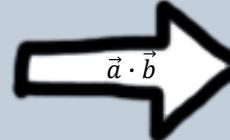
	설정 내용
예시 이미지	<ul style="list-style-type: none"><li>• 흰색 바탕에 검은색 테두리인 화살표</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (16) 가중치

- 화살표 안에 넣을 수 있는 가중치
- 현재 타일의 결과값이 숫자일 경우, 화살표에 그 값을 드래그 앤 드롭하여 가중치가 화살표에 입력되고, 그 값을 나타내는 문자이다.
- 가중치 수식 값은 현재 벡터  $a$ , 벡터  $b$ 에 따라 변화한다.
- HOLE에 화살표가 닿았을 경우, 가중치는 현재 화면에 보이는 벡터  $a$ , 벡터  $b$ 값으로 계산하여 SCORE값에 더한다.
- 화살표와 마찬가지로, 화살표와 HOLE이 충돌 판정이 나면, 화살표에 있는 가중치 문자 또한 사라진다.
- 설정 내용

string ID	WEIGHT
출력 내용	현재 타일 중 결과 값이 숫자인 수식
설정 내용	<ul style="list-style-type: none"><li>size: 10</li><li>색상: 검은색 계열</li><li>글자 굵게 처리</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (17) 발사 버튼

- 현재 타일의 결과값이 벡터일 경우, 화살표를 발사할 수 있는 장소이다.
- 발사 버튼은 화면 최하단에 3개가 배치되며, 각 기능은 동일하다.
- 현재 타일의 결과값이 벡터일 경우, 3개의 발사 버튼 중 하나의 버튼을 클릭하면, 화살표가 발사된다.
- 발사되는 방향이나 벡터의 성분 값은 현재 벡터  $a$ , 벡터  $b$ , 벡터의 연산에 따라 결정된다.
- 해당 물리량은 클라이언트와 논의 하에 상세 결정한다.
- 설정 내용

	설정 내용
예시 이미지	<ul style="list-style-type: none"><li>• 반원 모양</li><li>• 주황색</li></ul> 

## 5. UI 및 시스템 기획

### 2) UI 배치 상세안

#### (18) 배경 이미지

- 게임 화면의 배경 이미지
- 2640 x 1080 해상도 기준으로 제작하며, 다른 기종과의 호환에 문제가 있을 수 있으므로, 이보다 더 크게 제작한다.
- 게임 화면의 가장자리는 화살표가 튕겨 나갈 수 있도록 설정하며,  
튕겨 나가는 물리양은 클라이언트와 논의하여 상세 결정한다.

	설정 내용
이미지	<ul style="list-style-type: none"><li>• 2640 x 1080 해상도 기준보다 크게 제작</li><li>• 기본적으로 흰색 계열의 단색이나, 아트팀의 재량에 따라 무늬를 넣을 수 있다.</li><li>• 다만 무늬는 게임 플레이에 방해되지 않도록 작고, 투명도가 높도록 제작한다.</li><li>• 오른쪽 이미지는 예시 이미지이다.</li></ul>

# 수학 수업에서 쓰일 수 있는 예시



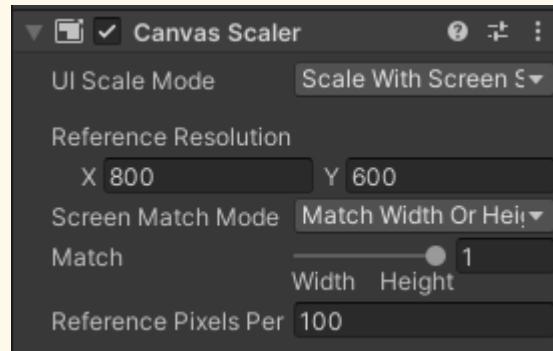
- 1) 평면 벡터 단원의 이론 수업을 교사가 진행한다. (이번 차시 이전 어플을 미리 다운 받도록 안내)
- 2) 학생들이 각자의 스마트폰을 이용하여 교육용 소프트웨어 게임에 접속한다.
- 3) 교사가 특정 시간동안 게임을 하도록 지시하고, 이후 함께 토의하는 시간을 가진다.
  - ① 교사가 게임을 할 시간을 10분으로 지정한다.
  - ② 학생들은 10분간 게임을 실행한다.
  - ③ 학생들이 10분 이후에 학생들은 각자의 점수를 말한다.
  - ④ 학생들이 벡터의 계산을 어떻게 했는지, 게임 속 벡터가 어떻게 쓰였는지 토의하는 시간을 가진다.

## 6. 개발 일지

### 1) SCORE UI 배치

#### (1) Canvas Scaler

- Match를 1로 설정하여 SCORE 및 상단 UI 4개 부분이 작게 보이던 현상을 해결하였다.
- 스마트폰의 해상도가 달라져 UI 배치가 달라질 것을 보완하기 위해  
UI Scale Mode를 Scale With Screen Size로 바꿨다.

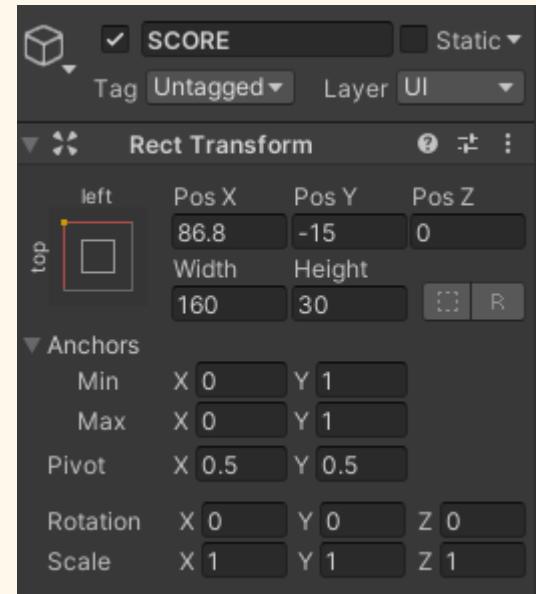
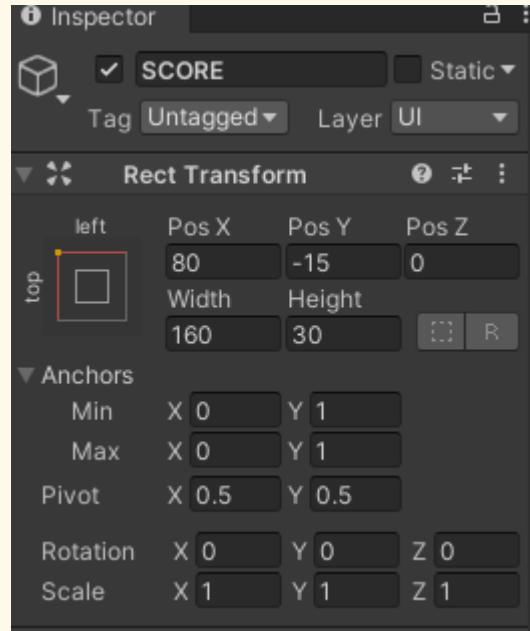


## 6. 개발 일지

### 1) SCORE UI 배치

#### (2) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 최좌측, 최상단에 배치하였다.
- 이후 Pos X를 조정하였다.

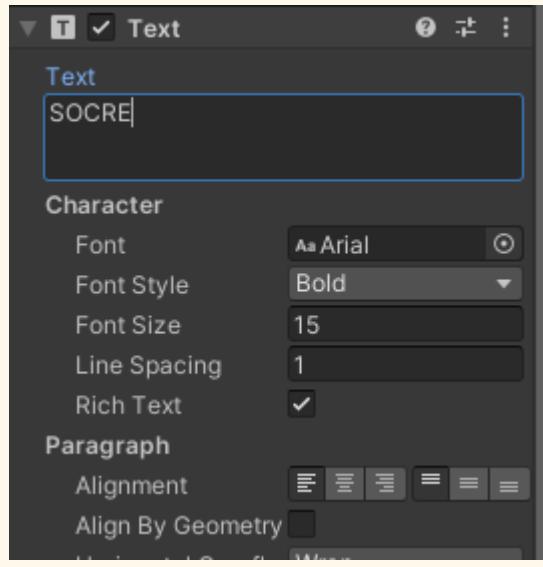


## 6. 개발 일지

### 1) SCORE UI 배치

#### (3) Text

- Text에 SCORE라고 입력
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정

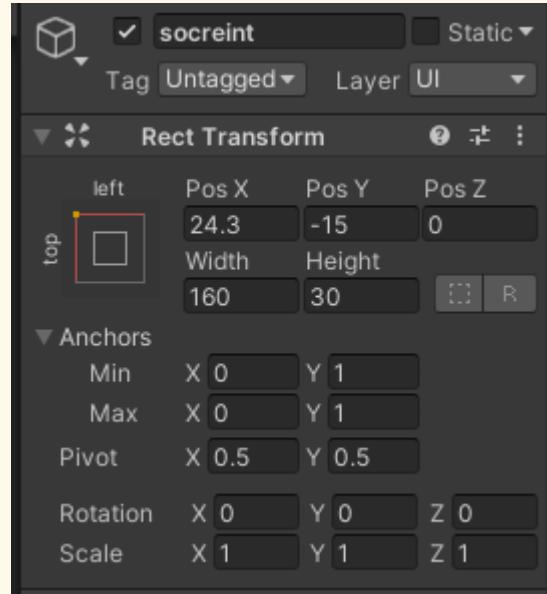


## 6. 개발 일지

### 2) 점수 UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 중앙, 최상단에 배치하였다.
- 이후 Pos X를 조정하였다.

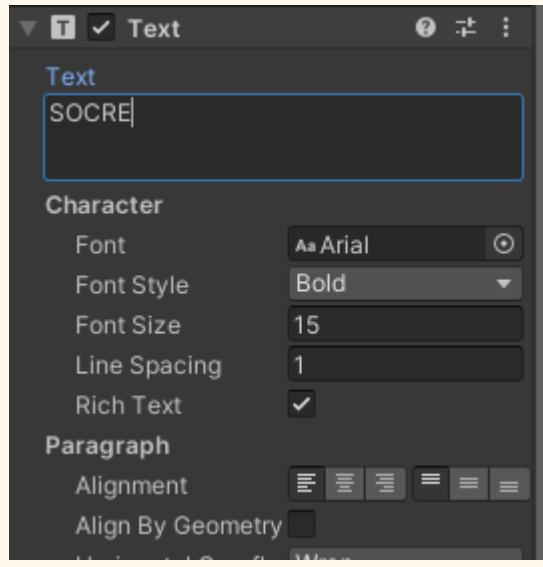


## 6. 개발 일지

### 2) 점수 UI 배치

#### (2) Text

- Text에 우선 숫자입력 → 이후 변수화를 해야 한다.
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정

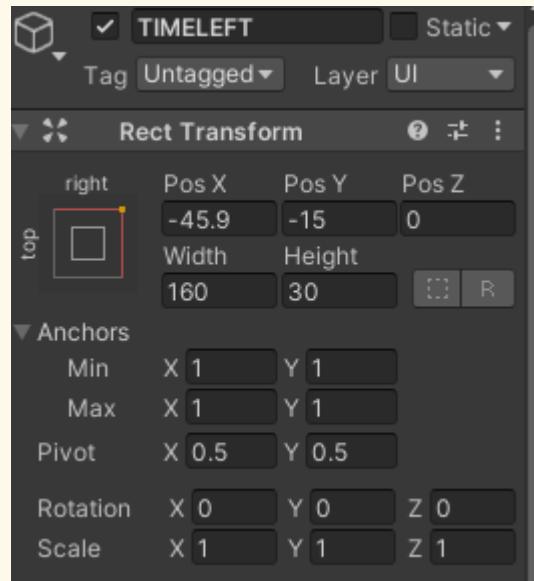


## 6. 개발 일지

### 3) TIME LEFT UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 중앙 최상단에 배치하였다.
- 이후 Pos X를 조정하였다.

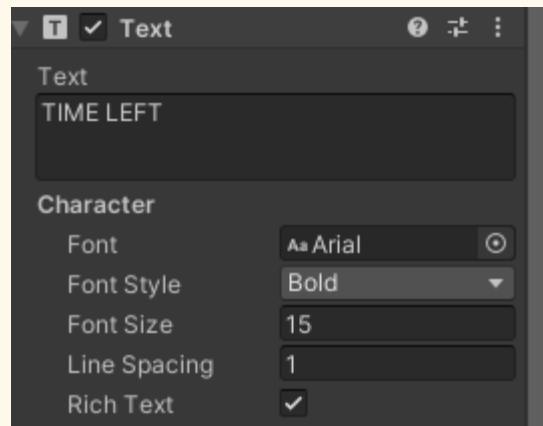


## 6. 개발 일지

### 3) TIME LEFT UI 배치

#### (2) Text

- Text에 TIME LEFT 입력
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정

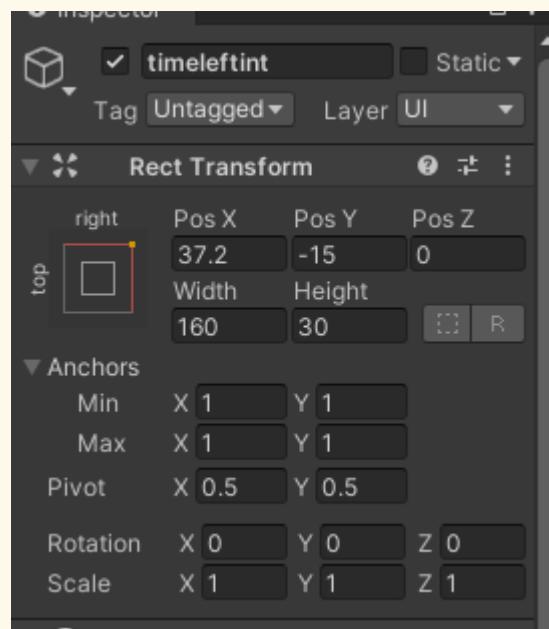


## 6. 개발 일지

### 4) 남은 시간 UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 최우측, 최상단에 배치하였다.
- 이후 Pos X를 조정하였다.

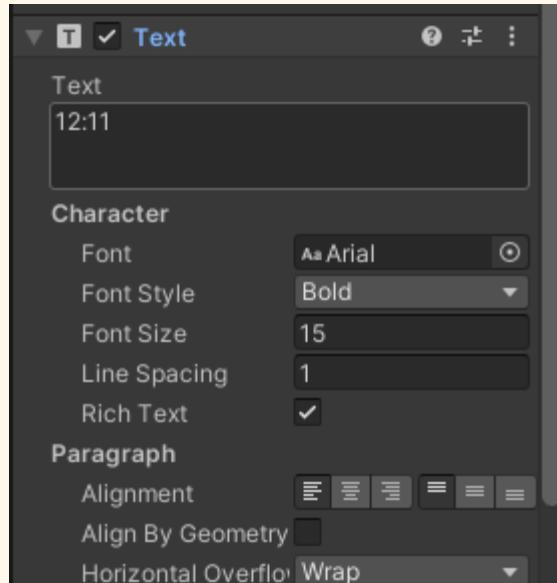


## 6. 개발 일지

### 4) 남은 시간 UI 배치

#### (2) Text

- Text에 우선 숫자입력 → 이후 변수화를 해야 한다.
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정



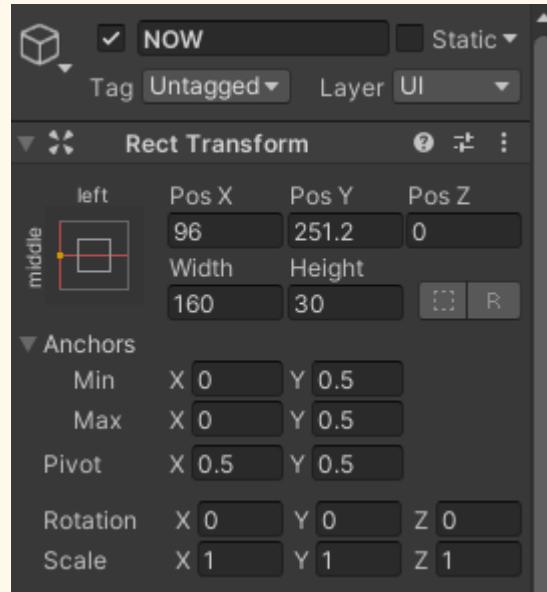
SOCRE 12345 TIME LEFT 12:11

## 6. 개발 일지

### 5) NOW UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 최좌측, 중앙에 배치하였다.
- 이후 Pos X, Pos Y를 조정하였다.

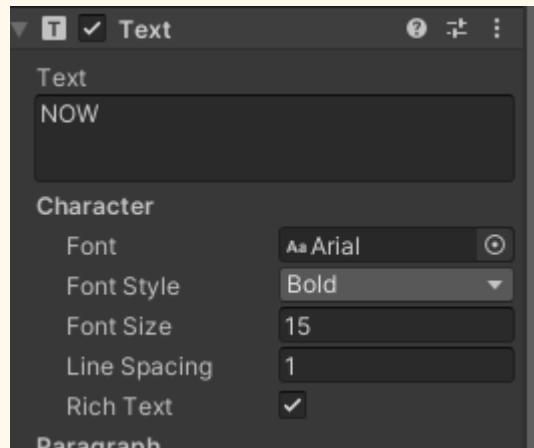


## 6. 개발 일지

### 5) NOW UI 배치

#### (2) Text

- NOW라고 입력
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정

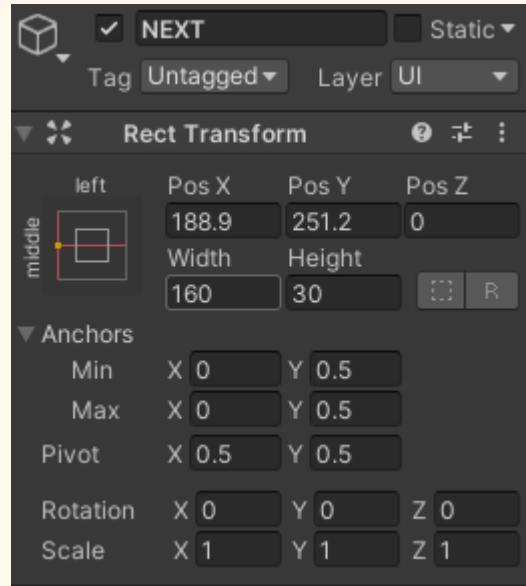


## 6. 개발 일지

### 6) NEXT UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 중앙에 배치하였다.
- 이후 Pos X, Pos Y를 조정하였다.

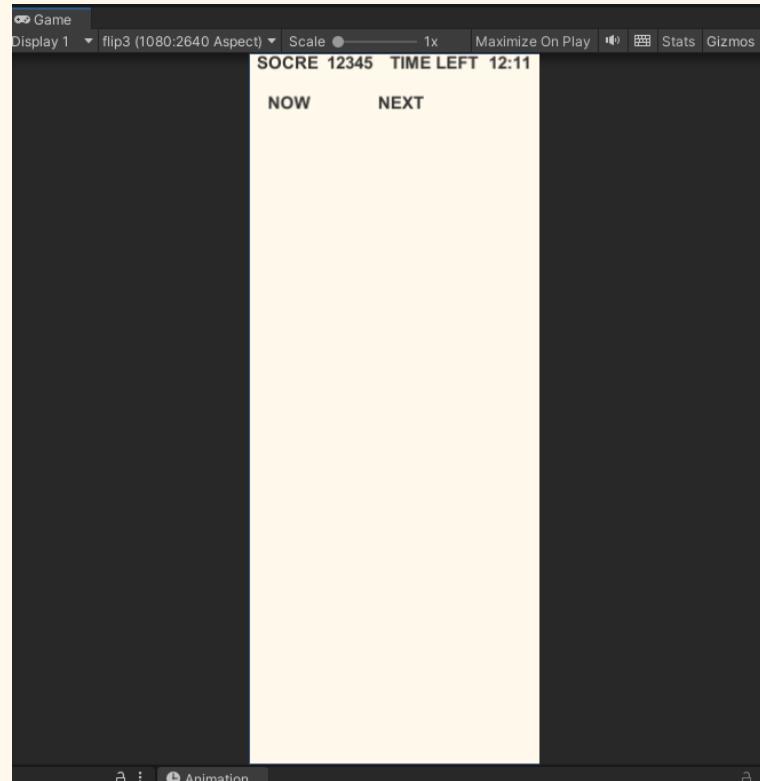
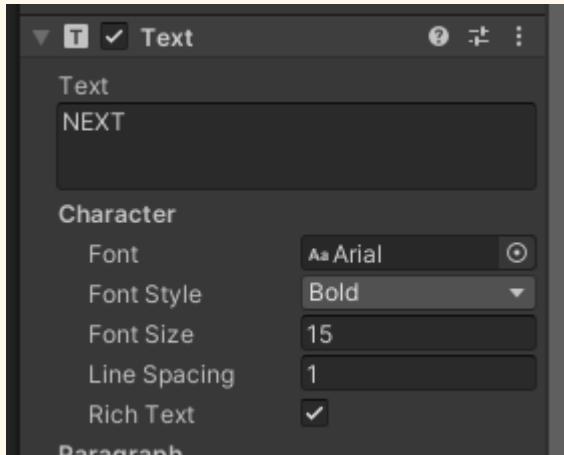


## 6. 개발 일지

### 6) NEXT UI 배치

#### (2) Text

- NEXT라고 입력
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정



## 6. 개발 일지

### 7) 발사 버튼 UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 중앙, 최하단에 배치하였다.
- 이후 Width와 Height를 조정하였다.

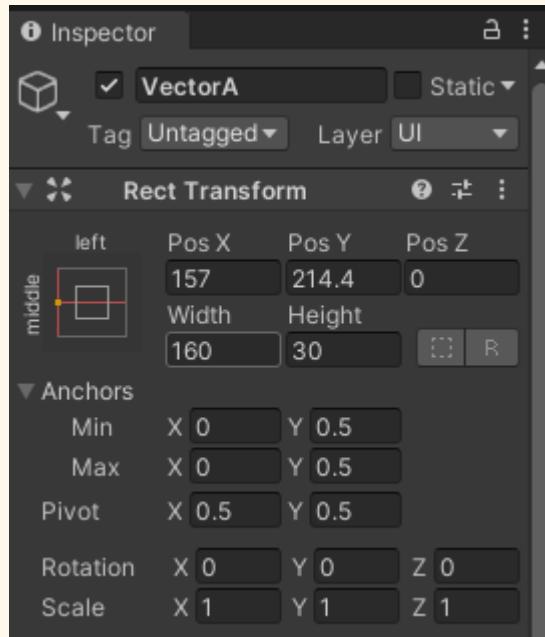


## 6. 개발 일지

### 8) 벡터 UI 배치

#### (1) Rect Transform

- Alt 키를 누른채로 top, left 사이의 박스를 클릭하여 최우측, 최상단에 배치하였다.
- 이후 Pos X, Pos Y를 조정하였다.

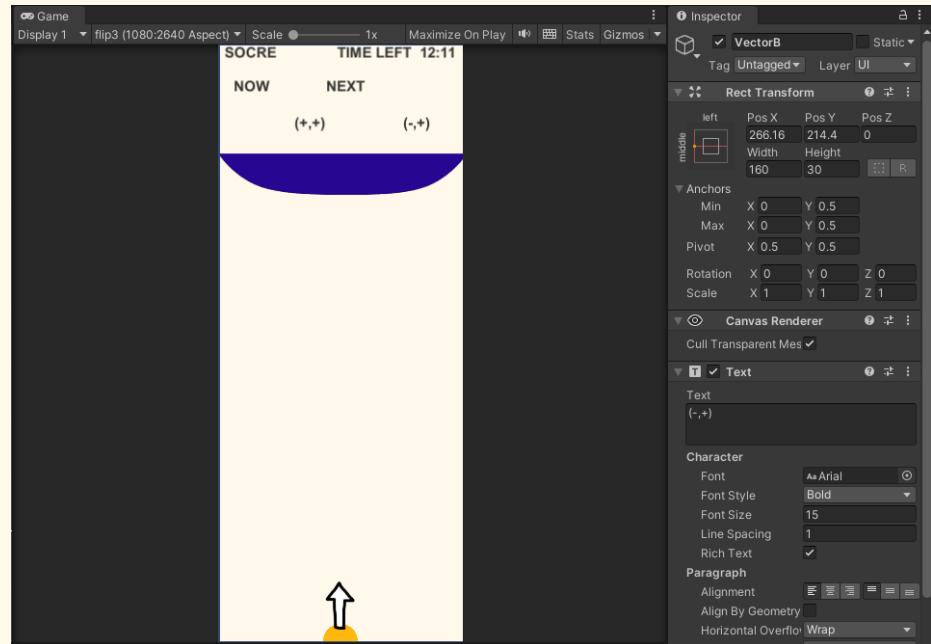


## 6. 개발 일지

### 8) 벡터 UI 배치

#### (2) Text

- Text에 우선 문자입력 → 이후 변수화를 해야 한다.
- Font Style을 Bold라고 지정, 굵게 처리
- Font Size를 15로 지정
- VectorB도 마찬가지로 만듦

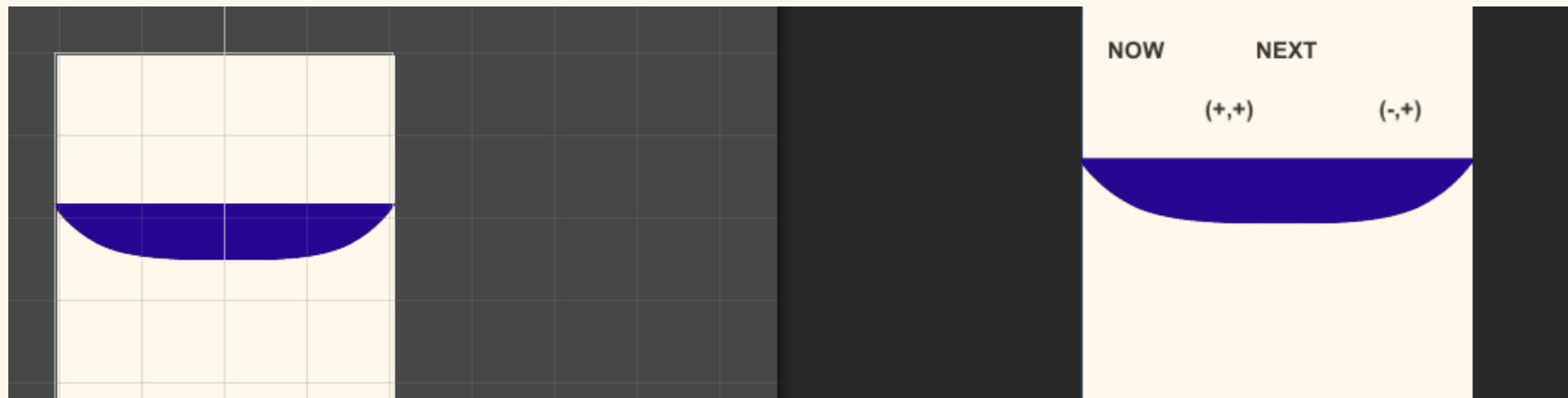


## 6. 개발 일지

### 7) 각종 Resource 배치

#### (1) HOLE 배치

- Resource 폴더에서 드래그 앤 드롭으로 게임 화면에 배치
- Order in Layer를 1로 만들어 맨 위로 올라오도록 조정

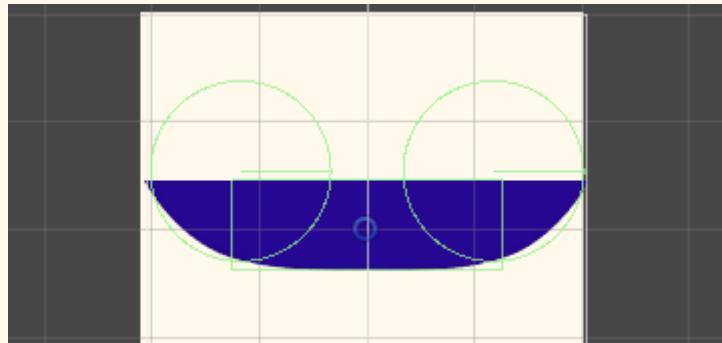


## 6. 개발 일지

### 7) 각종 Resource 배치

#### (1) HOLE 배치

- 충돌하고 서로 밀어내야 하므로, Box Collider 2D is Trigger 해제
- 중력 작용 피하기 위해 Rigidbody 2D에서 Body Type을 Kinematic으로 변경
- 사진 파일이 큰 관계로, Size를 조정하여 알맞게 조정
- Circle Collider 2D를 사용하여 양쪽 볼록한 구간에도 충돌하도록 설정



Rigidbody 2D

Body Type	Kinematic
Material	None (Physics M)
Simulated	<input checked="" type="checkbox"/>
Use Full Kinematic Collision	<input type="checkbox"/>
Collision Detection	Discrete
Sleeping Mode	Start Awake
Interpolate	None

Constraints

Info

Box Collider 2D

Edit Collider	
Material	None (Physics M)
Is Trigger	<input checked="" type="checkbox"/>
Used By Effector	<input type="checkbox"/>
Used By Composite	<input type="checkbox"/>
Auto Tiling	<input type="checkbox"/>
Offset	X 0 Y 0
Size	X 5 Y 1
Edge Radius	0

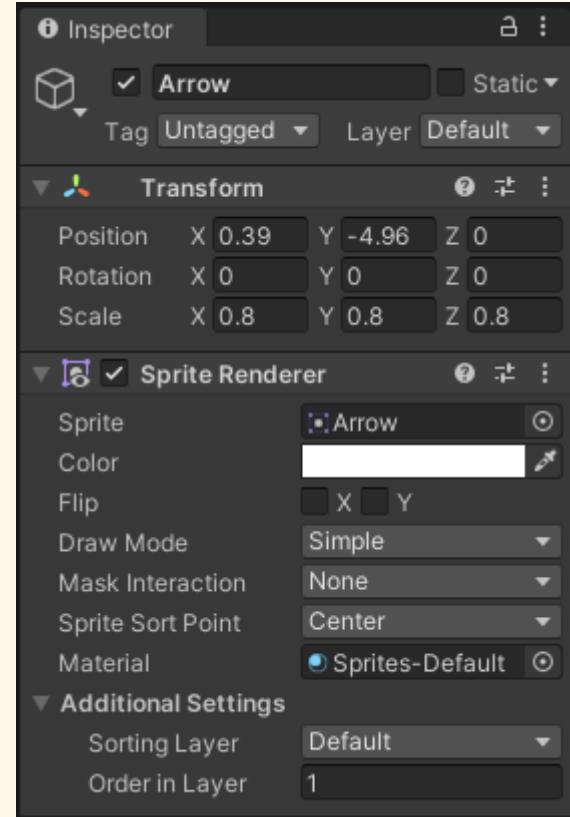
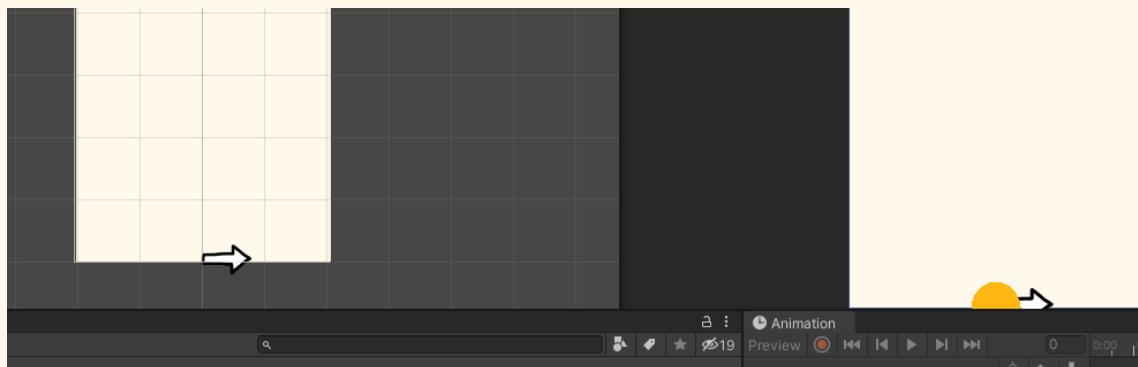
Info

## 6. 개발 일지

### 7) 각종 Resource 배치

#### (2) Arrow 배치

- Resource 폴더에서 드래그 앤 드롭으로 게임 화면에 배치
- Order in Layer를 1로 만들어 맨 위로 올라오도록 조정
- 버튼의 정중앙 하단을 기준으로 (0,0)이 되도록 한다.

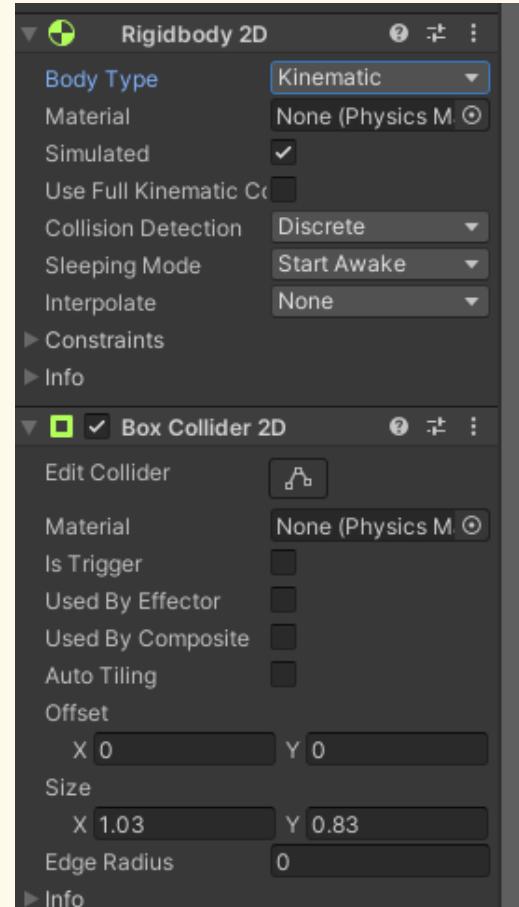


## 6. 개발 일지

### 7) 각종 Resource 배치

#### (2) Arrow 배치

- 충돌하고 서로 밀어내야 하므로, Box Collider 2D is Trigger 해제
- 중력 작용 피하기 위해 Rigidbody 2D에서  
Body Type를 Kinematic으로 변경



## 6. 개발 일지

### 8) Game Director(SCORE UI)

#### (1) 코드 작성

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; // UI 사용

public class GameDirector : MonoBehaviour
{
    GameObject socreint;
    GameObject HOLE;
    GameObject Arrow;
    public int socre = 0;

    // Start is called before the first frame update
    void Start()
    {
        // GameObject 찾기
        this.socreint = GameObject.Find("socreint");
        this.HOLE = GameObject.Find("HOLE");
        this.Arrow = GameObject.Find("Arrow");
    }

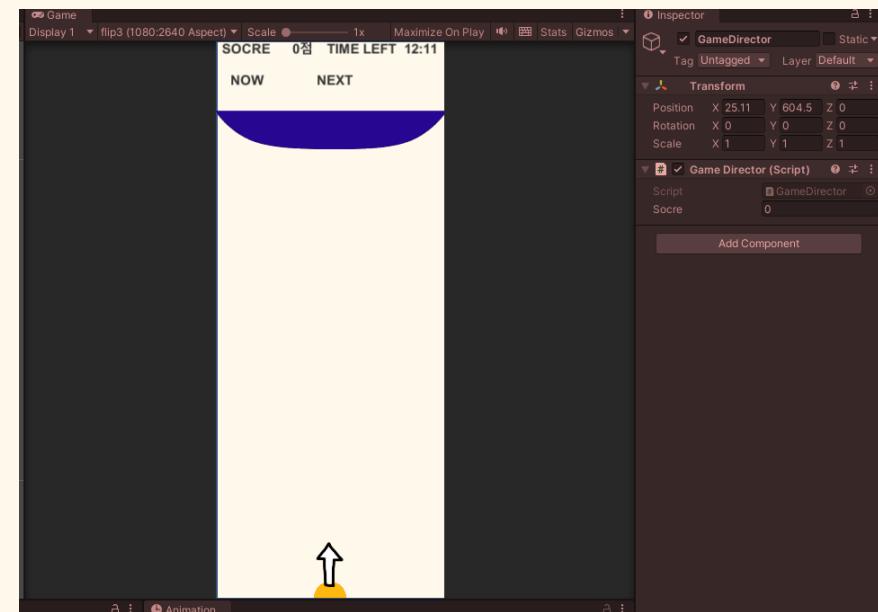
    // Update is called once per frame
    void Update()
    {
        this.socreint.GetComponent<Text>().text = socre.ToString("F0")+"점"; // score를 나타냄
    }
}
```

## 6. 개발 일지

### 8) Game Director(SCORE UI)

#### (2) 스크립트 적용

- 빈 오브젝트 생성
- 오브젝트 이름을 GameDirector라고 한다.
- Game Director 스크립트를 드래그 앤 드롭하여 적용한다.
- 게임 시작 시 SCORE 옆에 0점으로 표기됨을 확인했다.



## 6. 개발 일지

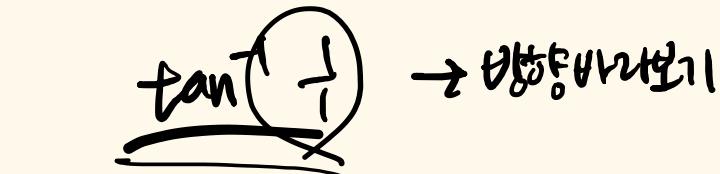
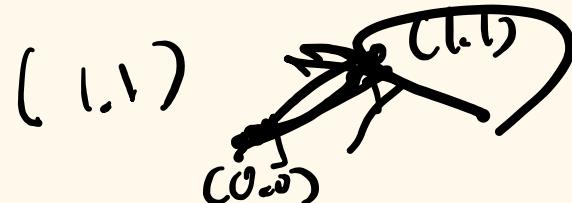
### 8) Game Director(VectorA UI)

(1) 코드 작성 (추가한 부분만 작성\_풀 코드는 깃허브에서 확인 가능)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; // UI 사용

public class GameDirector : MonoBehaviour
{
    GameObject VectorA;
    GameObject VectorB;
    public int ingredient_a_x, ingredient_a_y, ingredient_b_x, ingredient_b_y;
    // Start is called before the first frame update
    void Start()
    {
        // GameObject 찾기
        this.VectorA = GameObject.Find("VectorA");
        this.VectorB = GameObject.Find("VectorB");
    }

    // Update is called once per frame
    void Update()
    {
        this.VectorA.GetComponent<Text>().text = "(" + ingredient_a_x.ToString("F0") + "," + ingredient_a_y.ToString("F0") + ")"; // VectorA 성분을 나타냄
        this.VectorB.GetComponent<Text>().text = "(" + ingredient_b_x.ToString("F0") + "," + ingredient_b_y.ToString("F0") + ")"; // VectorB 성분을 나타냄
    }
}
```



(1,1) - (0,0)  
End origin End Pos

Vector2.up \* force  
bytge X fo.

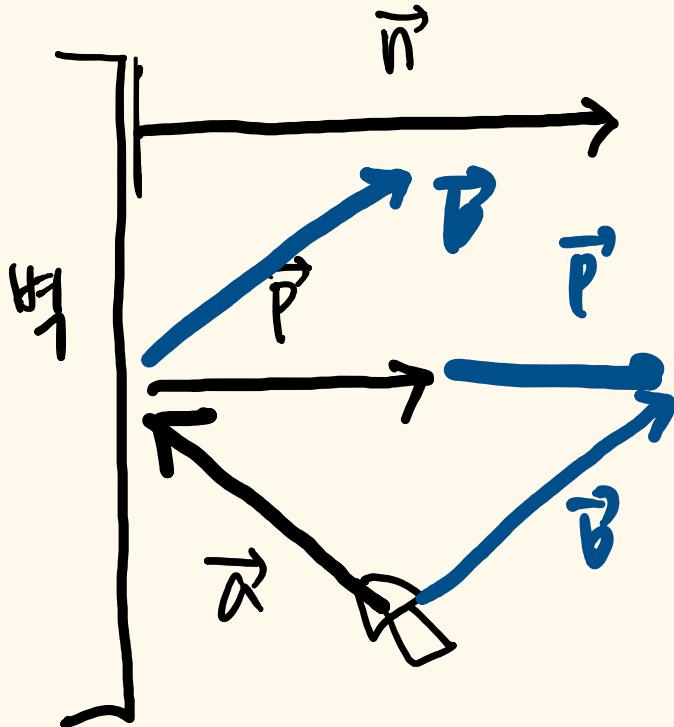
성분을 나타냄

## 6. 개발 일지

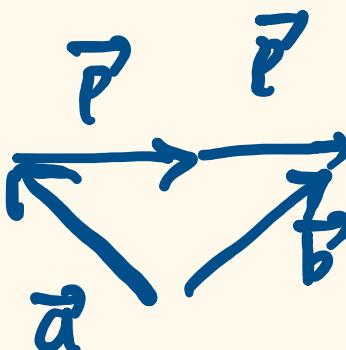
### 8) Game Director(VectorA UI)

(1) 코드 작성 (추가한 부분만 작성\_풀 코드는 깃허브에서 확인 가능)

தீர்வு முறை பல்கலை



$$\vec{P} = \text{Proj}_{\vec{n}} \vec{a} = \frac{|\vec{a} \cdot \vec{n}|}{|\vec{n}|} \cdot \frac{\vec{n}}{|\vec{n}|}$$



$$\vec{b} = \vec{a} + 2\vec{p}$$

## 6. 개발 일지

### 9) ArrowController

#### (1) 코드 작성

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Unity 스크립트 | 참조 대상
public class ArrowController : MonoBehaviour
{
    GameObject FireButton;
    GameObject HOLE;
    Rigidbody2D arrow_rigidbody2d;

    // Start is called before the first frame update
    // Unity 메시지 | 참조 대상
    void Start()
    {
        this.FireButton = GameObject.Find("FireButton");
        this.HOLE = GameObject.Find("HOLE");

        this.arrow_rigidbody2d = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    // Unity 메시지 | 참조 대상
    void Update()
    {
        // 방향 전환
        float angle = Mathf.Atan2(GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_y, GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_x) * Mathf.Rad2Deg; // transform.rotation = Quaternion.Euler(0,0,angle); // angle만큼 객체 회전
    }
}
```

## 6. 개발 일지

### 9) ArrowController

#### (1) 코드 작성

```
if(Input.GetButtonDown("FireButton"))
{
    // 방향 전환
    float angle = Mathf.Atan2(GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_y,
GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_x) * Mathf.Rad2Deg; // 각도 구하기
    transform.rotation = Quaternion.Euler(0, 0, angle); // angle만큼 객체 회전

    // 성분에 비례하여 화살표 발사
    float force =
    Mathf.Sqrt((GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_y)^2+( GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_x)^2);
    this.arrow_rigidbody2d.AddForce(transform.forward * force);
}
```

```
참조 대체
public void OnClickFireButton()
{
    // 방향 전환
    float angle = Mathf.Atan2(GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_y, GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_x)
    transform.rotation = Quaternion.Euler(0, 0, angle); // angle만큼 객체 회전

    // 성분에 비례하여 화살표 발사
    float force = Mathf.Sqrt((GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_y) ^ 2 + (GameObject.Find("GameDirector").GetComponent<GameDirector>().ingredient_a_x) ^ 2);
    this.arrow_rigidbody2d.AddForce(transform.forward * force);
}
```

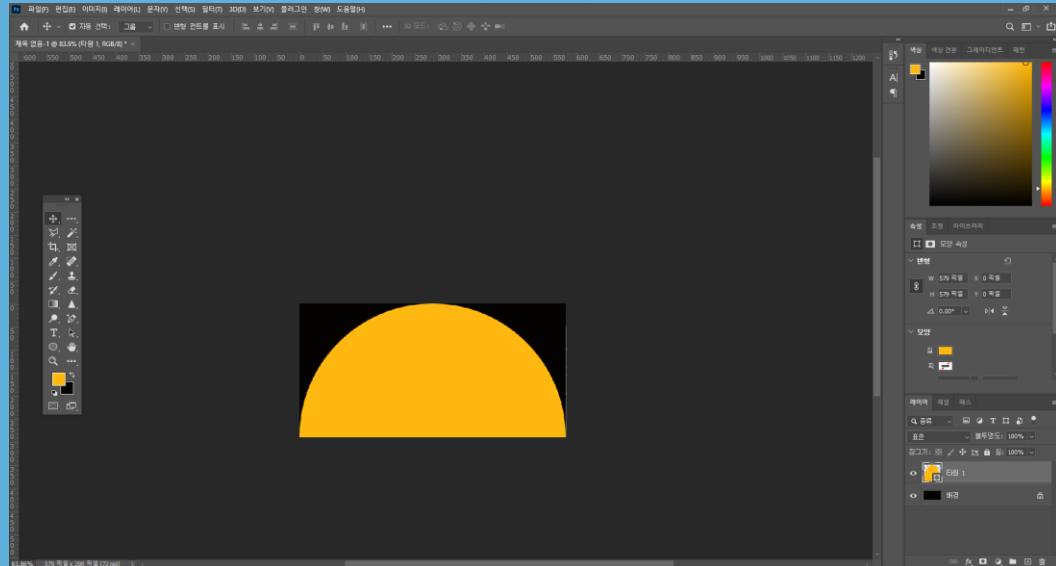
수정 이유: 버튼이 눌렸을 때만 작동하게 하기 위해 bool형을 사용할 예정이고, FireButton을 기준 Image에서 UI → button으로 변경하기 위해

# 7. 아트 일지

## 1) 발사 버튼 제작

### (1) 반원 제작

- Shift 키를 계속 눌러 타원이 아닌 원으로 만든다.
- 주황색 계열로 칠하고, 획은 무색으로 한다.
- 자르기를 사용하여 전체 화면 자체를 잘라낸다.



## 7. 아트 일지

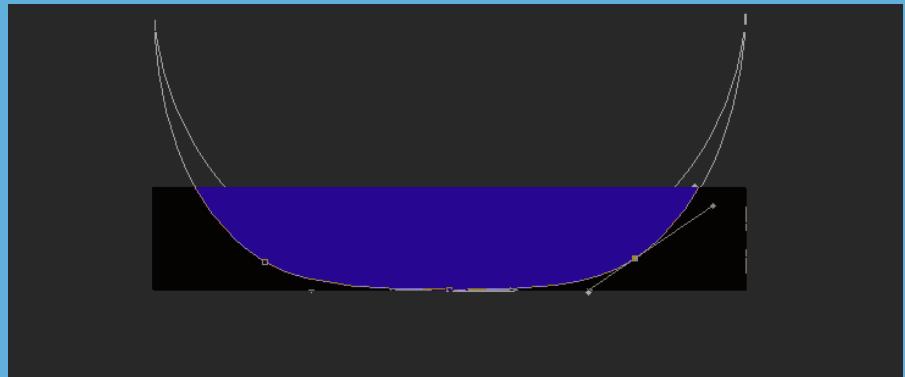
### 2) HOLE 제작

#### (1) 활꼴 제작

- Shift 키를 계속 눌러 타원이 아닌 원으로 만든다.
- 남색 계열로 칠하고, 획은 무색으로 한다.
- 자르기를 사용하여 전체 화면 자체를 잘라낸다.

#### (2) 패스에 기준점을 추가하여 편집

- 패스에 기준점을 추가하여 모서리 둥근 사각형 느낌으로 편집한다.



# 기획 및 개발 일정\_맨 마지막 페이지로 갈 것

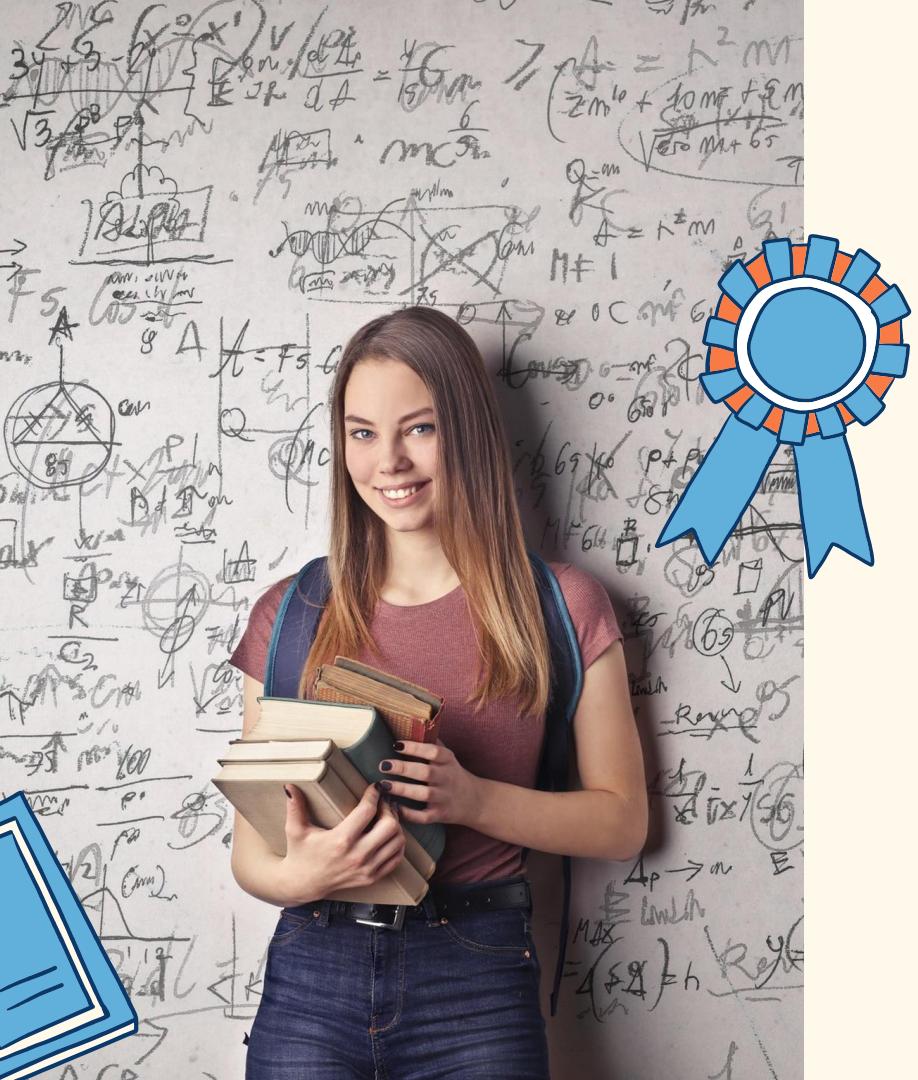
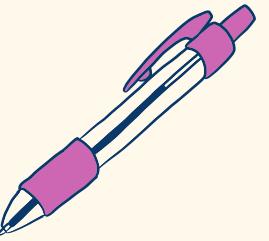
	Week 1_기획							Week 2_개발						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
기획 및 개발 PPT 작성	기획 시작					기획 마감								
게임 설정 및 시스템 기획	게임 개요	시스템 기획	PPT start		순서도 작성									
메인 구현							메인 틀 완성							
챕터 전체 메소드														
investment 메소드 구현														
sale 메소드 구현														
valuechange part														
버그 발생 유무 테스트														회계 정리



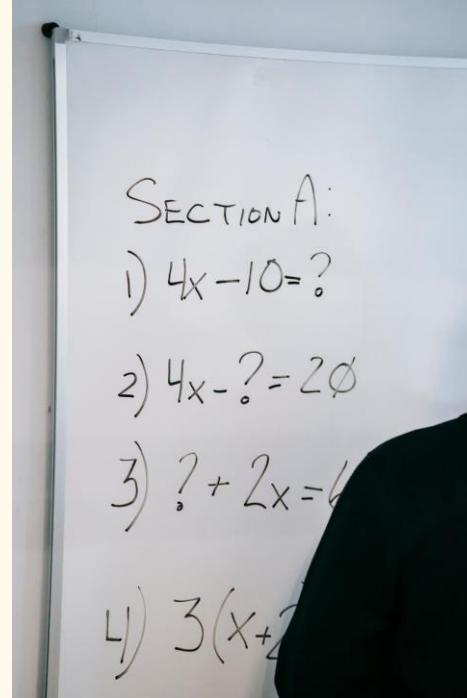
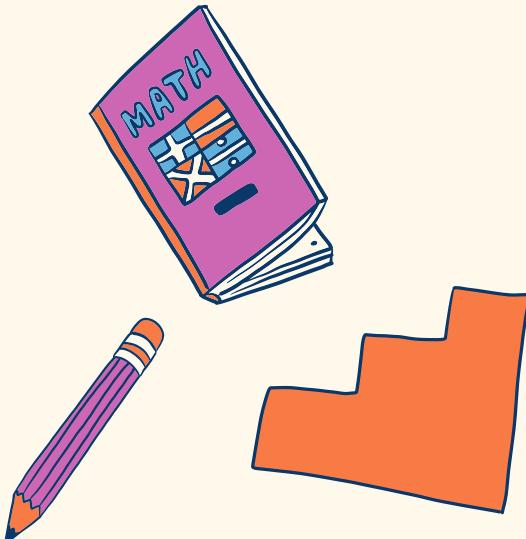
Hello there!

# I'm Rain

I'll be sharing with you my beautiful ideas. Follow me at **@reallygreatsite** to learn more.



# Write Your Topic or Idea



Add a main point

Briefly elaborate on what you want to discuss.



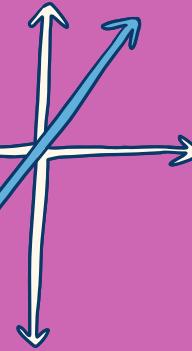
Add a main point

Briefly elaborate on what you want to discuss.

# Via Heading Here

$$a^2 + b^2 = c^2$$

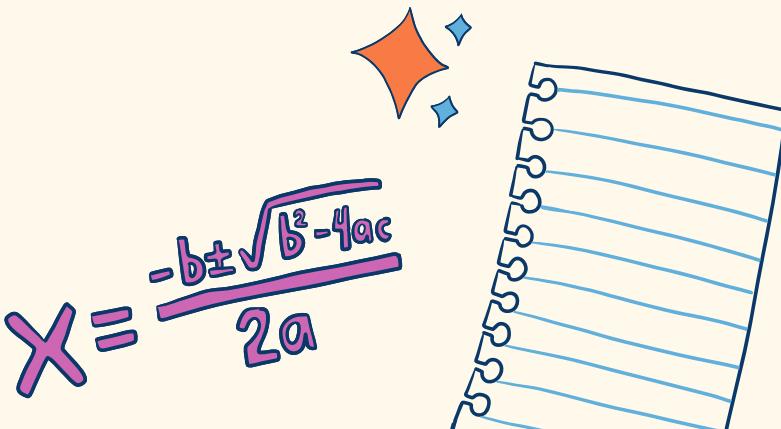




Write a caption  
for the photos.

# Add a Team Members Page

Elaborate on what you want to discuss.



Name

Title or Position



Name

Title or Position



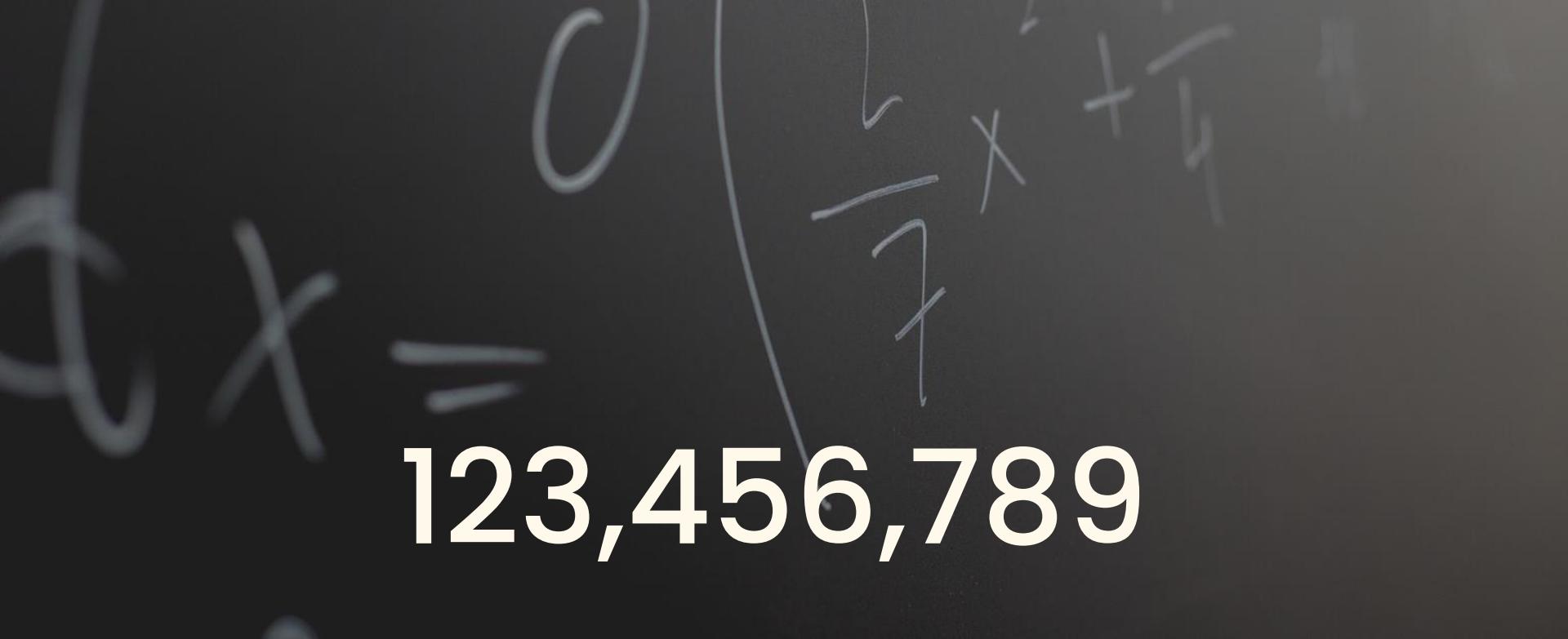
Name

Title or Position



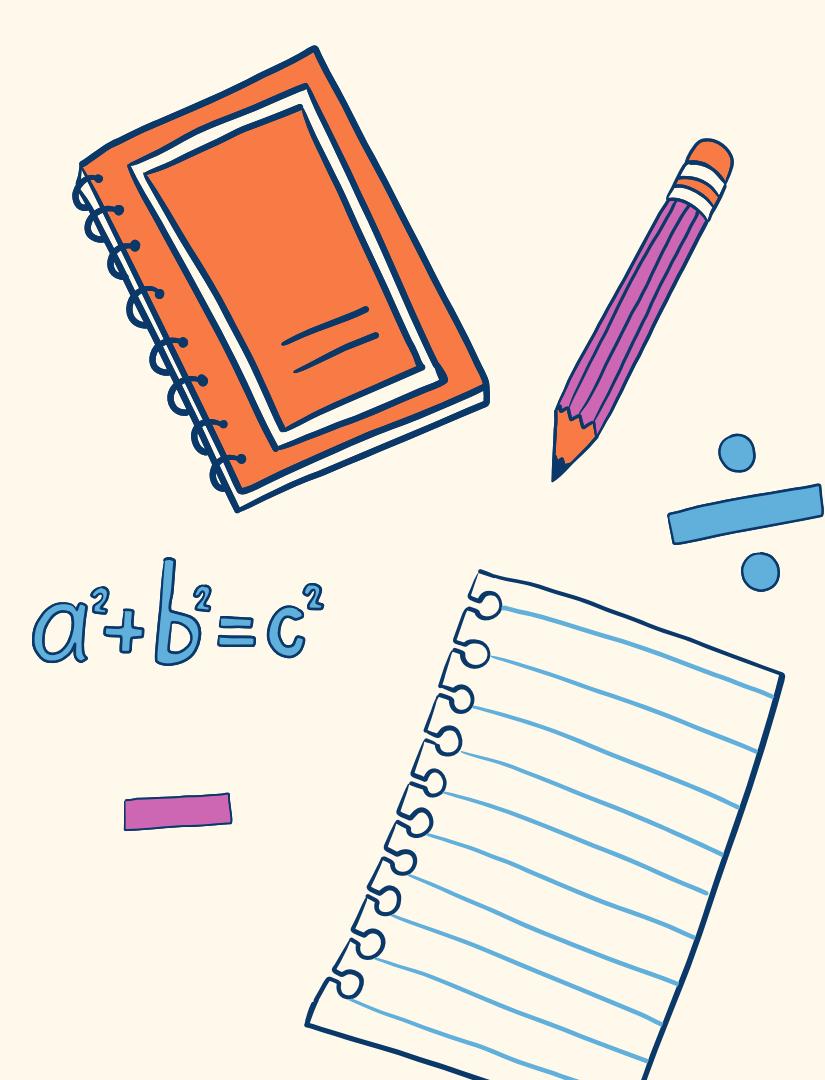
Name

Title or Position



123,456,789

Elaborate on the featured statistic.



# Thank you!

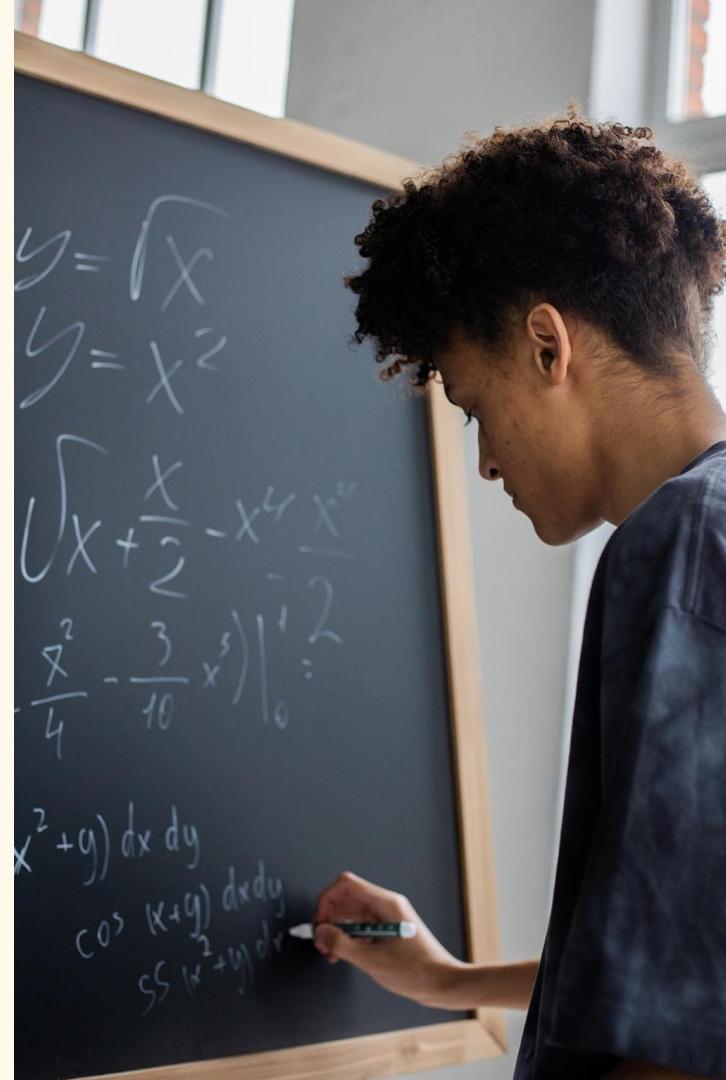
Write a closing statement or  
call-to-action here.

# Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Pexels](#)

**Happy designing!**



# Add a Timeline Page



## Add a main point

Briefly elaborate on what you want to discuss.



## Add a main point

Briefly elaborate on what you want to discuss.



## Add a main point

Briefly elaborate on what you want to discuss.



## Add a main point

Briefly elaborate on what you want to discuss.

# S

## Strengths

What are you doing well?  
What sets you apart?  
What are your good qualities?

# W

## Weaknesses

Where do you need to improve?  
Are resources adequate?  
What do others do better than  
you?

# O

## Opportunities

What are your goals? Are  
demands shifting? How can it be  
improved?

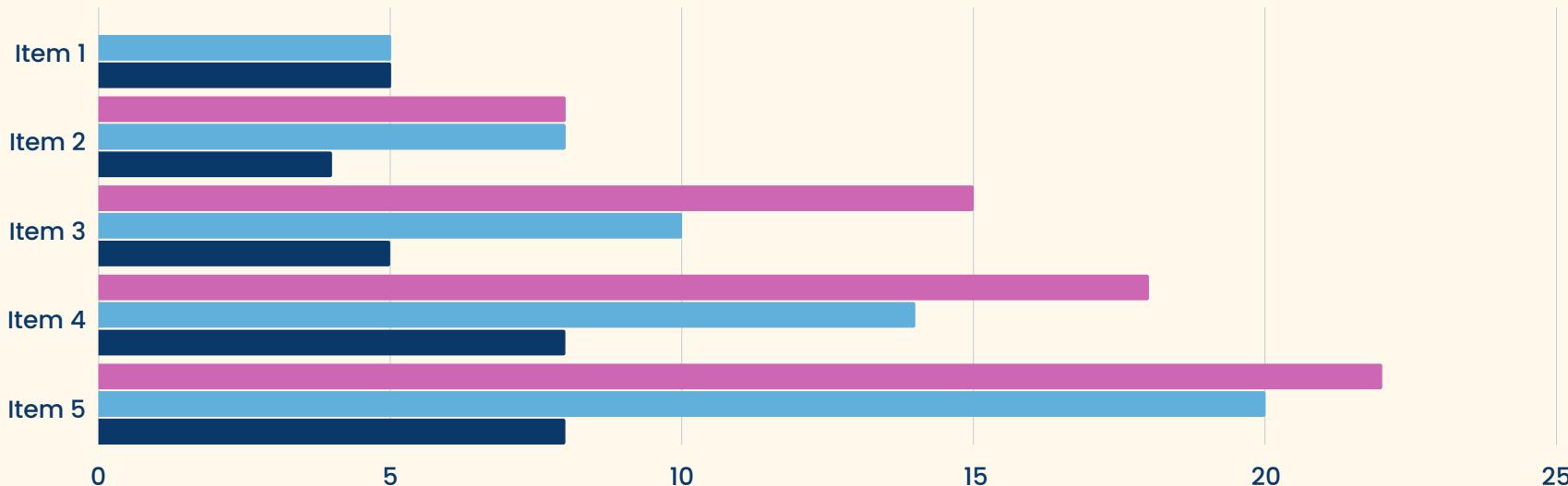
# T

## Threats

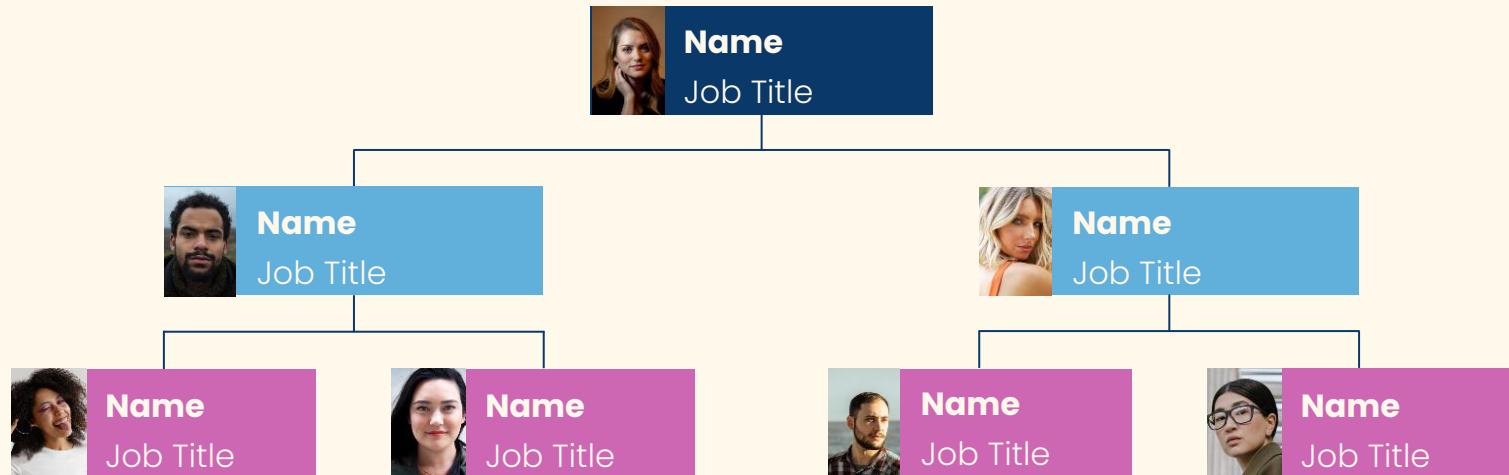
What are the blockers you're  
facing? What are factors outside  
of your control?

# Write Your Topic or Idea

Elaborate on what you want to discuss.



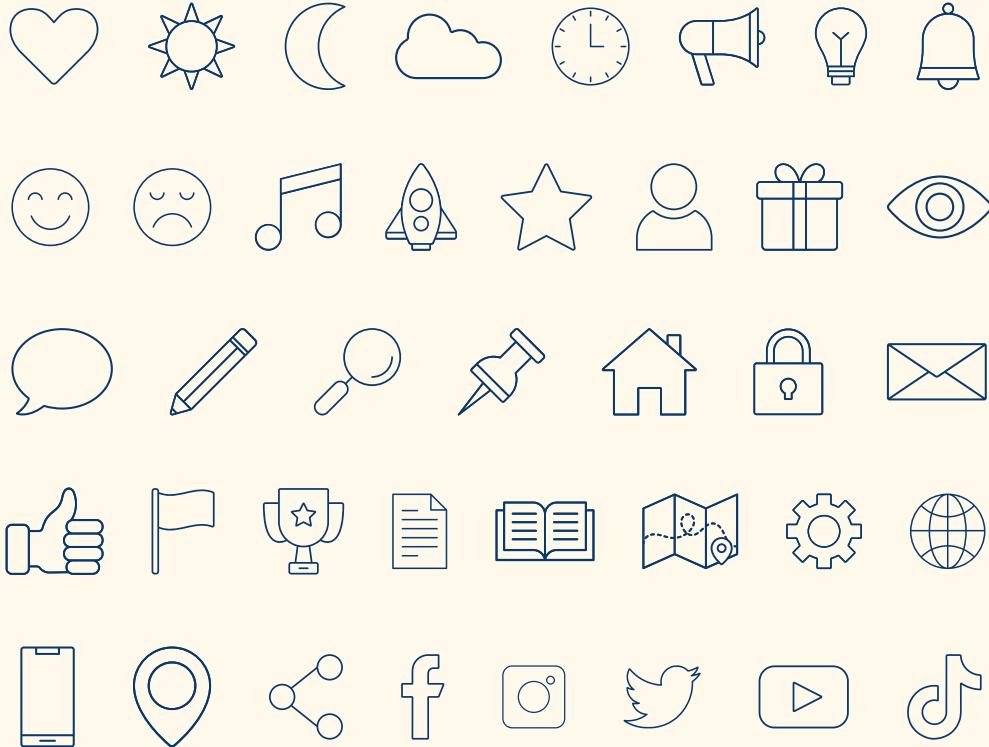
# Add a Chart Page



# Icons Resource Page

Use these design resources in your Presentation. Happy designing!

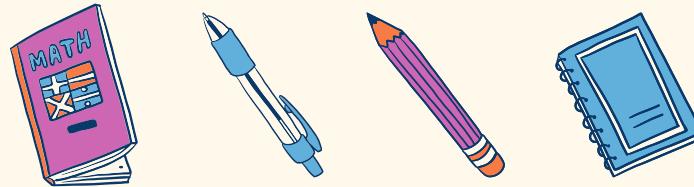
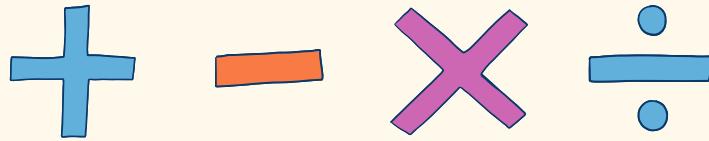
Don't forget to delete this page before presenting.



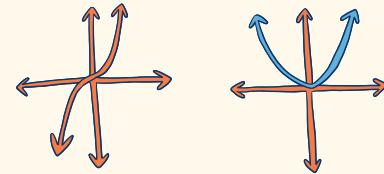
# Elements Resource Page

Use these design resources in your Presentation. Happy designing!

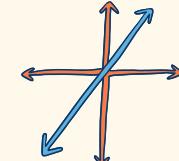
Don't forget to delete this page before presenting.



$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



$$a^2 + b^2 = c^2$$



# Elements Resource Page

Use these design resources in your Presentation. Happy designing!

Don't forget to delete this page before presenting.





## Free templates for all your presentation needs



For PowerPoint and  
Google Slides



100% free for personal  
or commercial use



Ready to use,  
professional and  
customizable



Blow your audience  
away with attractive  
visuals