

# WHO IS NEXT



Bo Jackson  
Ohtani Shohei

# IDEA

We call an all-rounder, or [이도류],  
someone who is good at different positions.



*But sometimes...*

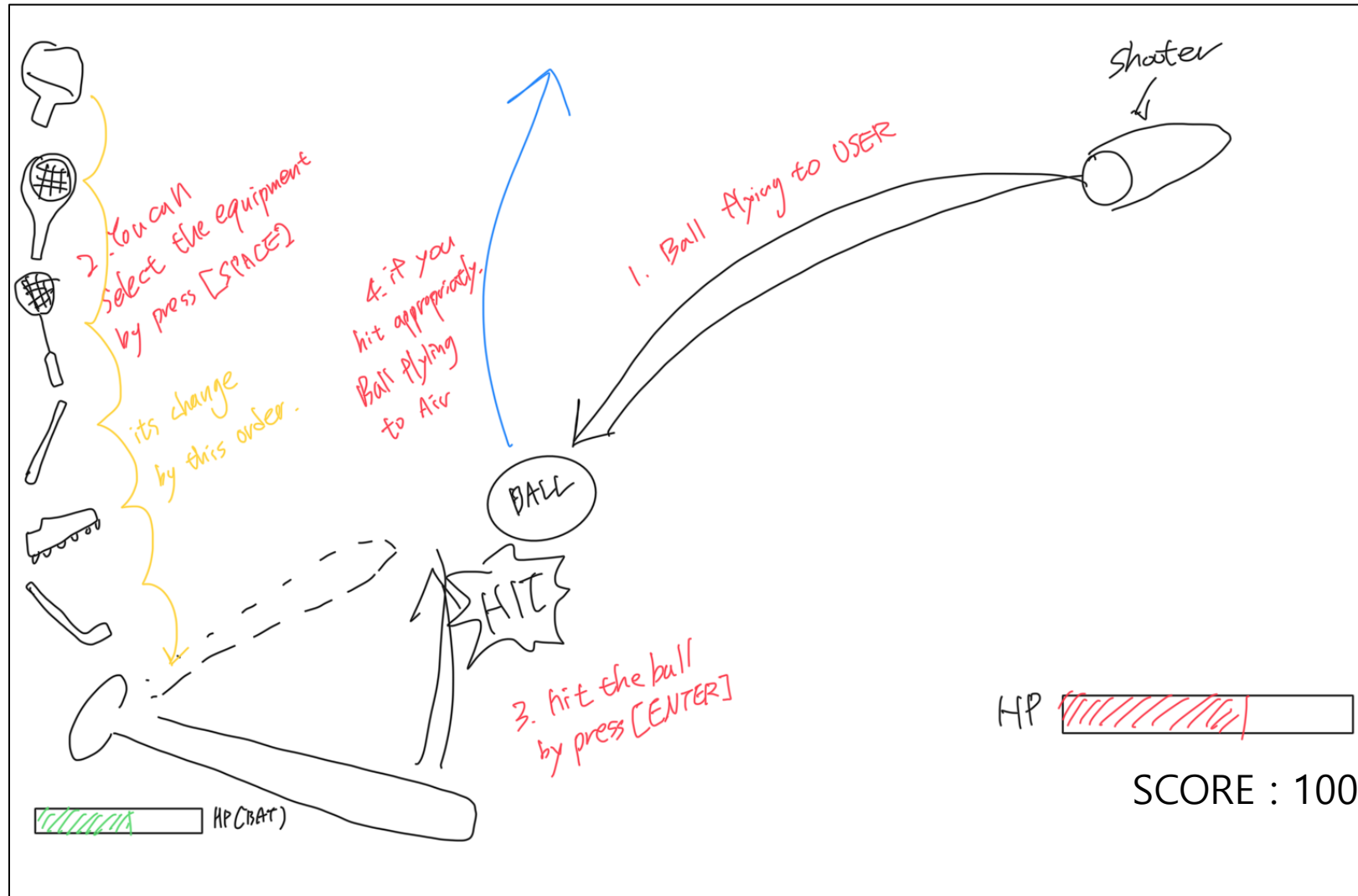
there are people who are good at two sports.

This game is about creating such MONSTER

# Concept

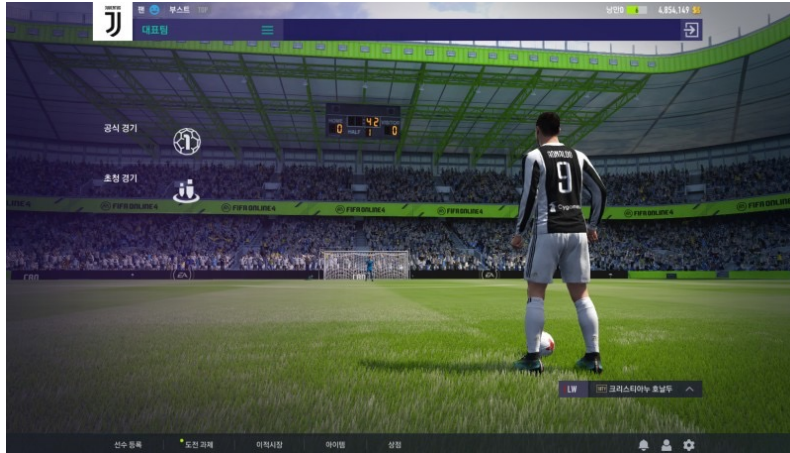
- various balls are constantly flying.
- The user has a variety of equipment(bat, racket, etc.) and can equip them one by one.
- When a ball is thrown at you, you must hit it with the appropriate equipment.
- If you hit the ball with the wrong equipment, you lose the equipment and lose HP.
- When all HP is lost, the game ends and a record is kept.

# Reference

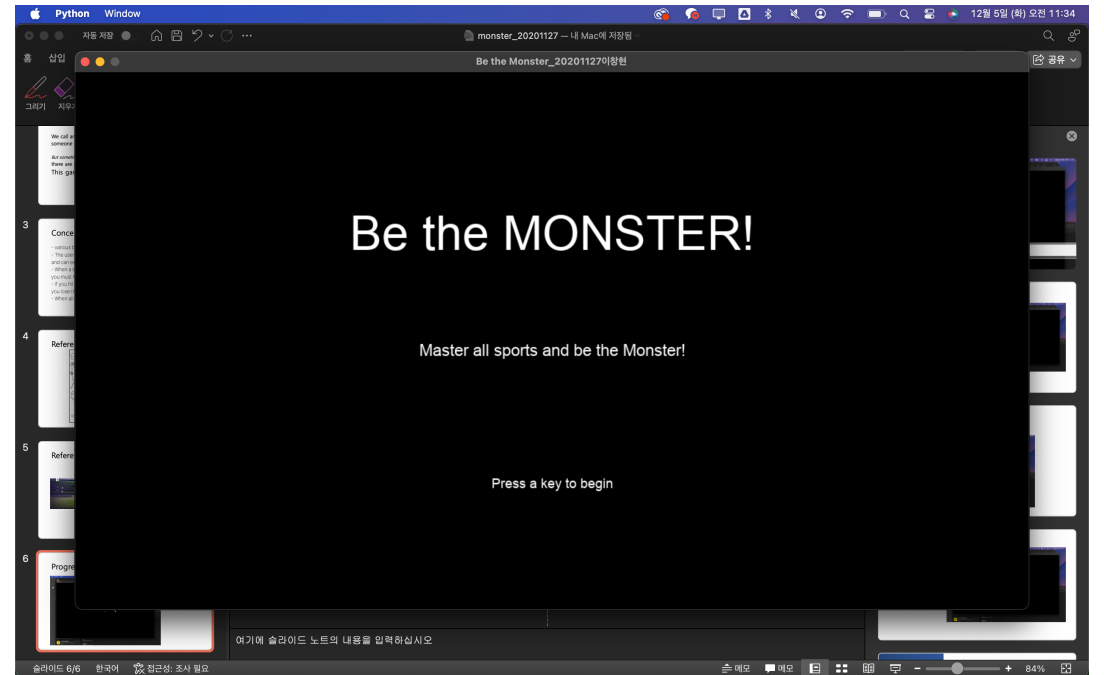
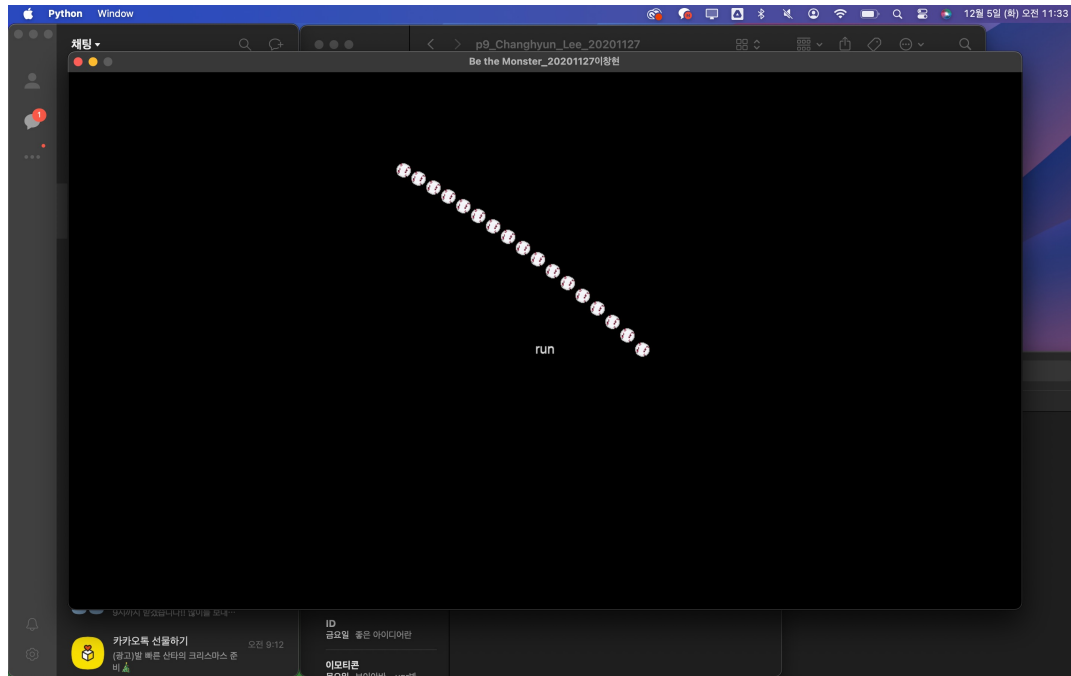




# Reference



# Current progress



# code

```
84 class Ball(pg.sprite.Sprite):
85     def __init__(self):
86
87         # 랜덤한 이미지 설정
88         self.image = random.choice(ball_images)
89         self.rect = self.image.get_rect()
90
91         # 초기 위치 설정
92         self.rect.center = (random.randint(50, WIDTH - 50), 50)
93
94         # 초기 속도 및 가속도 설정
95         self.speed_x = random.choice([-5, 5])
96         self.speed_y = random.uniform(-5, 5)
97         self.gravity = 0.5
98
99         # 1초 간격으로 생성을 위한 타이머 설정
100        self.appearance_timer = pg.time.get_ticks()
101        self.appearance_interval = 1000 # 1 second
102
```

```
104 def update(self):
105     now = pg.time.get_ticks()
106     if now - self.appearance_timer > self.appearance_interval:
107         self.appearance_timer = now
108         new_ball = Ball(self.all_balls, self.all_equipment)
109         new_ball.speed_x = random.choice([-5, 5])
110         new_ball.speed_y = random.uniform(-5, 5)
111         new_ball.rect.center = (random.randint(50, WIDTH-50), 50)
112         self.all_balls.add(new_ball)
113
114         self.rect.x += self.speed_x
115         self.rect.y += self.speed_y
116         self.speed_y += self.gravity
117
118         if self.rect.bottom > HEIGHT:
119             self.kill()
```

# code

```
121 class Circle(pg.sprite.Sprite):
122     def __init__(self, all_circles):
123         super().__init__(all_circles)
124
125         # 초기 위치 설정
126         self.rect = pg.Rect(1000,200, 20, 20)
127         self.rect.center = (1000,200)
128
129         # 초기 속도 및 가속도 설정
130         self.speed_x = 20
131         self.speed_y = 0
132         self.gravity = 0.5
133
134         # 랜덤한 이미지 설정
135         original_image = random.choice(ball_images)
136
137         # 이미지 크기 조정
138         scaled_width = 20 # 조절하고자 하는 폭
139         scaled_height = 20 # 조절하고자 하는 높이
140         self.image = pg.transform.scale(original_image, (scaled_width, scaled_height))
141         self.rect = self.image.get_rect()
```

```
143     def update(self):
144         # 속도와 가속도의 영향을 받아 이동
145         self.rect.x += self.speed_x
146         self.rect.y += self.speed_y
147
148         # 가속도 적용
149         self.speed_y += self.gravity
150
151         # 특정 조건에 도달하면 제거
152         if self.rect.y > HEIGHT:
153             print("Circle removed!")
154             self.kill()
```



# have to solve

- equipment class
- score bar(variable)
- collide logic

# have to prepare

- 10 various ball image
- 10 various equipment image
- rule explanation video