

게임으로 배우는 파이썬

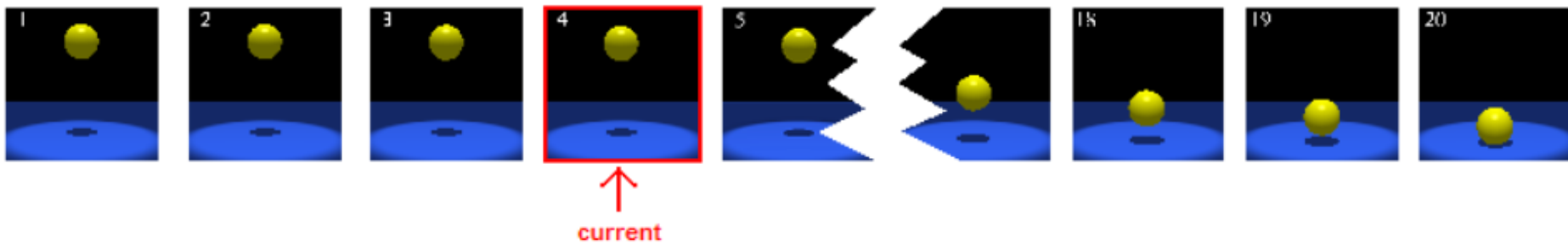




참고. 간단한 애니메이션

- 애니메이션이란 이미지가 빠르게 표시되는 이미지의 시퀀스
- 애니메이션을 얻기 위해선 다음과 같은 할 일이 있다.

1. 이미지를 리스트에 로드 (`pygame.image.load()` 사용)
2. 업데이트 때마다 현재 이미지를 다음 이미지로 변경
3. 카운터가 리스트 끝에 도달하면 처음으로 재설정
4. 매끄럽게 보이도록 충분히 빠르게 진행





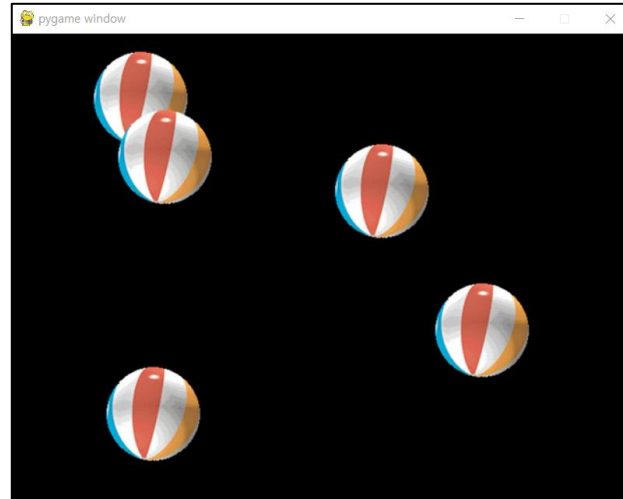
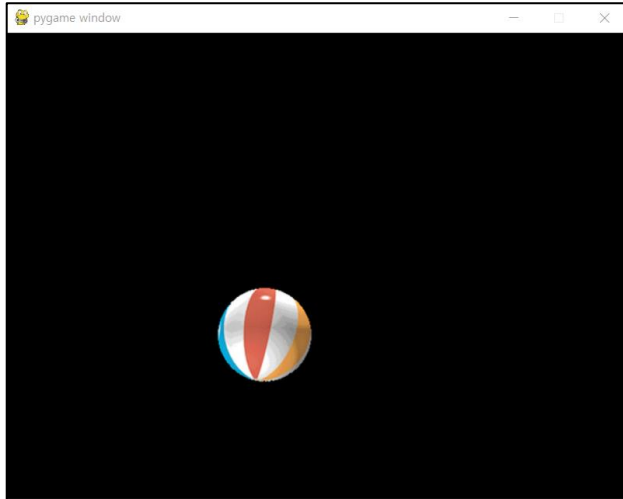
참고. 게임 만들기

➤ 프로젝트 만들기

1. 이야기 만들기
2. 등장 캐릭터 결정하기
3. 무대 결정하기
4. 캐릭터가 할 일 결정하기
5. 무대에서 할 일 결정하기
6. 프로젝트에 필요한 변수 결정하기

01. 튜기는 공 화면 보호기 만들기

- 윈도우 내에서 공을 튜기는 화면 보호기



- 공이 사방 벽을 부딪히며 움직이는 프로그램



01. 튜기는 공 화면 보호기 만들기

- 윈도우 내에서 공을 튜기는 화면 보호기
 - ① 윈도우의 크기를 지정합니다.
 - ② 볼 이미지를 불러온다. (ball.png)
 - ③ 볼과 관련된 모든 변수를 초기화하고 초기화한다.
 - ④ 공을 움직이고 윈도우 경계선과 충돌을 검사하며 공 방향과 속도를 바꾼다
 - ⑤ 공을 윈도우에 그린다



01. 튕기는 공 화면 보호기 만들기

단일 이미지, rect 객체로 윈도우에서 튕겨져 다니게 만들기

Filename : PygameOneBallBouceRects.py

```
import pygame
from pygame.locals import *
import sys
import random
```

```
BLACK = (0, 0, 0)
WINDOW_WIDTH = 640
WINDOW_HEIGHT = 480
FRAMES_PER_SECOND = 30
N_PIXELS_PER_FRAME = 3
```

파이게임 환경 초기화하기

```
pygame.init()
window = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
clock = pygame.time.Clock()
```

이미지, 소리 등 애셋 불러오기

```
ballImage = pygame.image.load('images/ball.png')
```

01. 튜기는 공 화면 보호기 만들기



```
# 변수 초기화하기
```

```
ballRect = ballImage.get_rect()
```

```
MAX_WIDTH = WINDOW_WIDTH - ballRect.width
```

```
MAX_HEIGHT = WINDOW_HEIGHT - ballRect.height
```

```
ballRect.left = random.randrange(MAX_WIDTH)
```

```
ballRect.top = random.randrange(MAX_HEIGHT)
```

```
xSpeed = N_PIXELS_PER_FRAME
```

```
ySpeed = N_PIXELS_PER_FRAME
```



01. 튜기는 공 화면 보호기 만들기

```
while True:
    for event in pygame.event.get():
        # Clicked the close button? Quit pygame and end the program
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # '프레임당' 처리해야 할 일 정의하기
    if (ballRect.left < 0) or (ballRect.right >= WINDOW_WIDTH):
        xSpeed = -xSpeed # reverse X direction

    if (ballRect.top < 0) or (ballRect.bottom >= WINDOW_HEIGHT):
        ySpeed = -ySpeed # reverse Y direction

    # 두 방향에 대한 속도로 공의 위치 갱신하기
    ballRect.left = ballRect.left + xSpeed
    ballRect.top = ballRect.top + ySpeed

    window.fill(BLACK) # 윈도우 내 내용물 지우기
    window.blit(ballImage, ballRect) # 윈도우 내 모든 내용물(요소) 그리기
    pygame.display.update() # 윈도우 갱신하기
    clock.tick(FRAMES_PER_SECOND) # 윈도우 갱신주기 정하기
```




01. 튜기는 공 화면 보호기 만들기

- 윈도우 내에서 공을 튜기는 화면 보호기

1. 볼을 만드는 클래스

- `__init__()` 메서드

공을 그릴 윈도우 자체, 윈도우 너비, 윈도우 높이, 이미지 로드

이미지를 불러오고 그려질 위치를 정하며 모든 인스턴스 변수를 초기화한다.

- `update()` 메서드

x, y 방향의 속도에 따라 프레임마다 위치를 바꾼다

경계선에 부딪혔는지 검사

- `draw()` 메서드

윈도우에 볼을 그린다.



01. 튜기는 공 화면 보호기 만들기

- 윈도우 내에서 공을 튜기는 화면 보호기
- 2. 볼 클래스를 활용한 메인 프로그램
 - Ball 객체를 생성한다.
 - 해당 객체의 `update()` 및 `draw()` 메서드를 프레임마다 호출한다.

```
import pygame
from pygame.locals import *
import random
```

```
# Ball class
class Ball():
```

```
    def __init__(self, window, windowWidth, windowHeight):
        self.window = window          # 나중에 그릴 수 있도록 윈도우 기억
        self.windowWidth = windowWidth
        self.windowHeight = windowHeight

        self.image = pygame.image.load('images/ball.png')
        # rect 는 [x, y, 너비, 높이]로 구성된다.
        ballRect = self.image.get_rect()
        self.width = ballRect.width
        self.height = ballRect.height
        self.maxWidth = windowWidth - self.width
        self.maxHeight = windowHeight - self.height

        self.x = random.randrange(0, self.maxWidth)
        self.y = random.randrange(0, self.maxHeight)

        speedsList = [-4, -3, -2, -1, 1, 2, 3, 4]
        self.xSpeed = random.choice(speedsList)
        self.ySpeed = random.choice(speedsList)
```



01. 튜기는 공 화면 보호기 만들기

```
def update(self):
    # Check for hitting a wall. If so, change that direction.
    if (self.x < 0) or (self.x >= self.maxWidth):
        self.xSpeed = -self.xSpeed

    if (self.y < 0) or (self.y >= self.maxHeight):
        self.ySpeed = -self.ySpeed

    # Update the Ball's x and y, using the speed in two directions
    self.x = self.x + self.xSpeed
    self.y = self.y + self.ySpeed

def draw(self):
    self.window.blit(self.image, (self.x, self.y))
```



01. 튜기는 공 화면 보호기 만들기

pygame demo 6(a) - using the Ball class, bounce one ball

Filename : Main_BallBounce.py

```
import pygame
from pygame.locals import *
import sys
import random
from Ball import * # bring in the Ball class code
```

```
# 상수 정의하기
BLACK = (0, 0, 0)
WINDOW_WIDTH = 640
WINDOW_HEIGHT = 480
FRAMES_PER_SECOND = 30
```

```
# 파이게임 환경 초기화하기
pygame.init()
window = pygame.display.set_mode((WINDOW_WIDTH,
WINDOW_HEIGHT))
clock = pygame.time.Clock()
```



01. 튜기는 공 화면 보호기 만들기

```
# 변수 초기화하기
```

```
oBall = Ball(window, WINDOW_WIDTH, WINDOW_HEIGHT)
```

```
while True:
```

```
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
            pygame.quit()  
            sys.exit()
```

```
# 프레임당 처리해야 할 일 정의하기
```

```
oBall.update()          # 공 상태 갱신하기
```

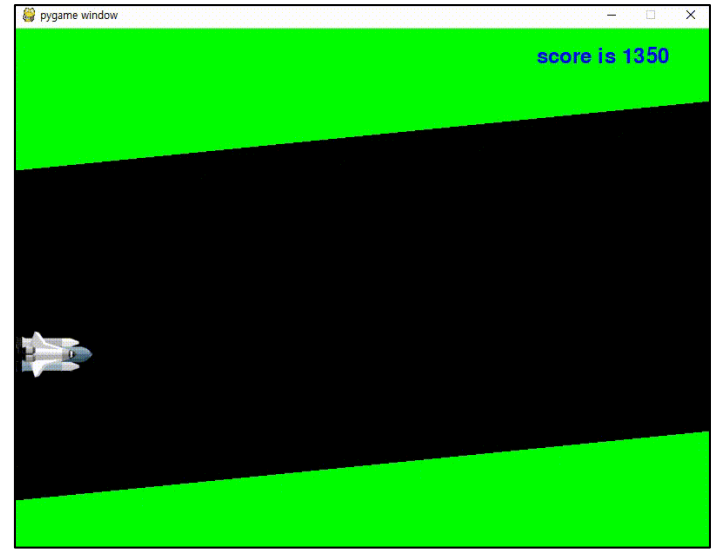
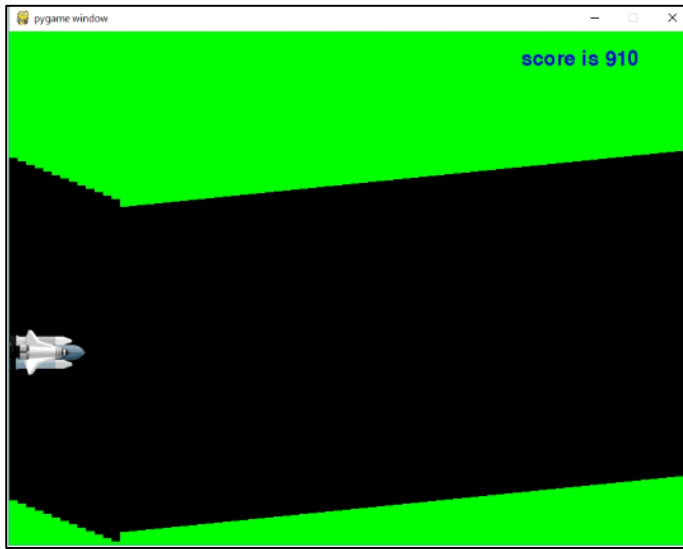
```
window.fill(BLACK)
```

```
oBall.draw()            # 공 그리기
```

```
pygame.display.update()
```

```
clock.tick(FRAMES_PER_SECOND)
```

02. Cave



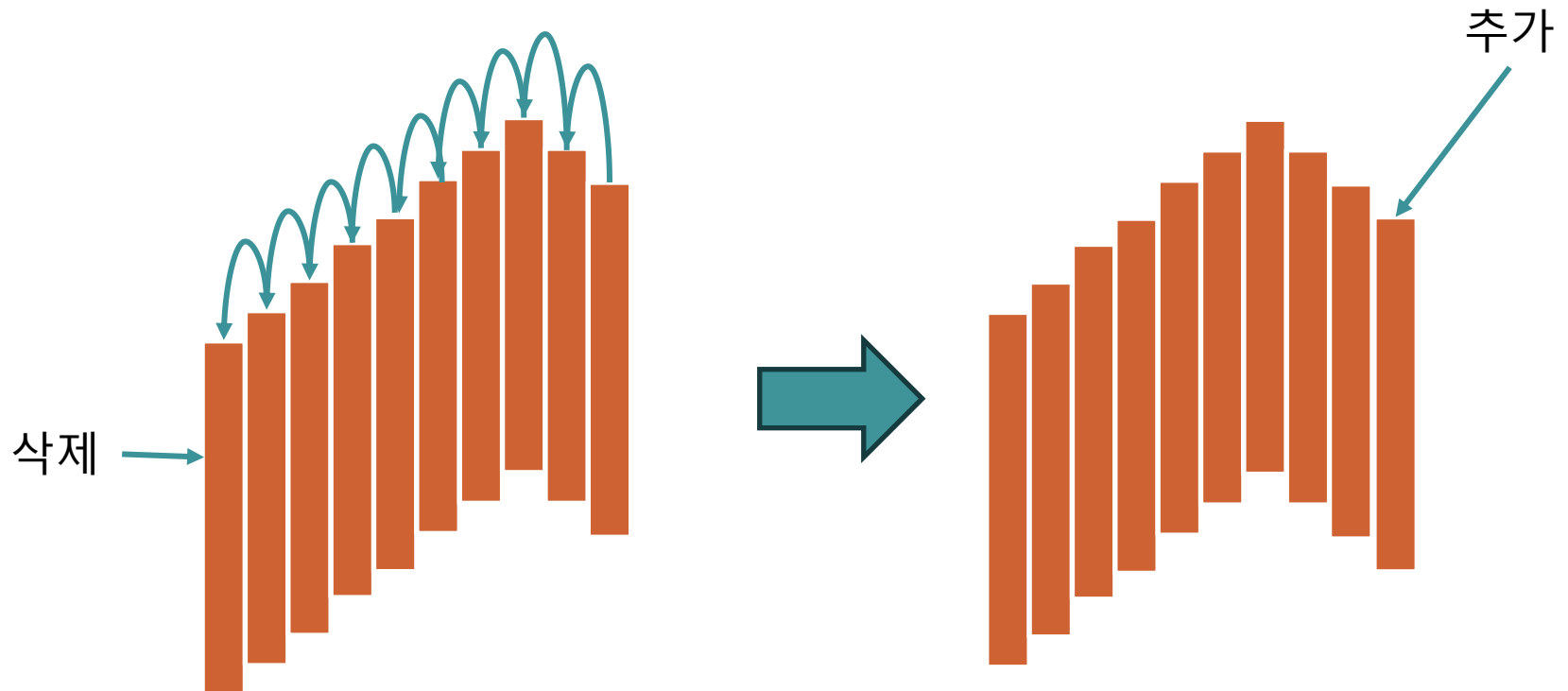
- 가로 스크롤게임.
스페이스 키를 누르면 위 방향으로 가속도가 붙는다. 동굴은 점차 좁아진다.
벽에 부딪치면 게임 종료



02. Cave

➤ 동굴 표현

사각형을 왼쪽 방향으로 움직이고, 맨 앞의 사각형을 오른쪽 끝에 추가



03. 마인 스윕퍼



- 폭탄 게임

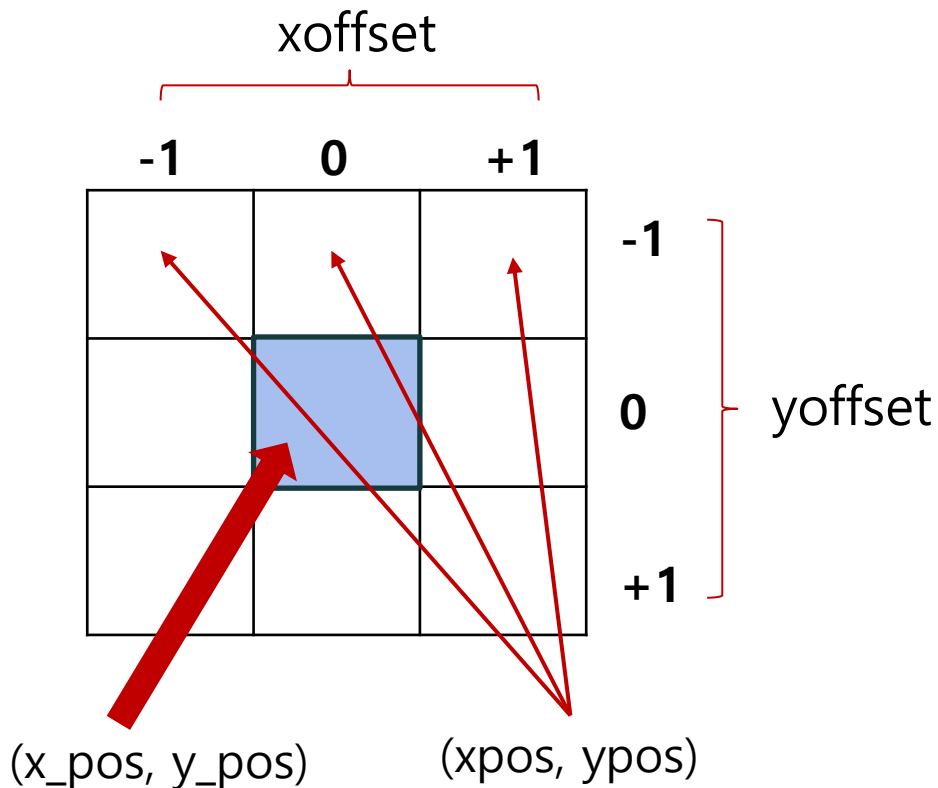
숨겨져있는 폭탄을 피하면서 전체 타일을 뒤집는 게임.

숫자는 주위에 있는 폭탄 수 를 나타낸다.

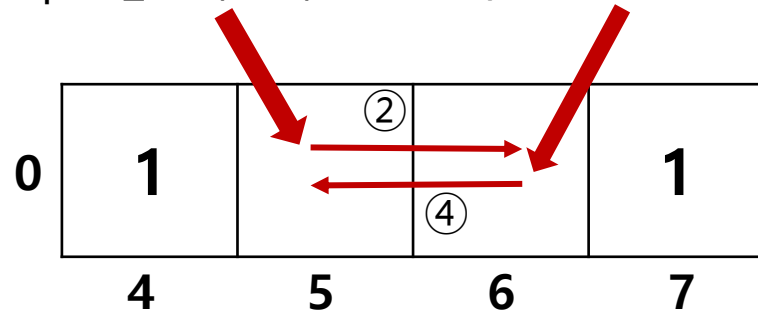
03. 마인 스위퍼

➤ 폭탄 수 세기

num_of_bomb(field, x_pos, y_pos) 함수



① open_tile(5, 0) ③ open_tile(6, 0)



04. Snake

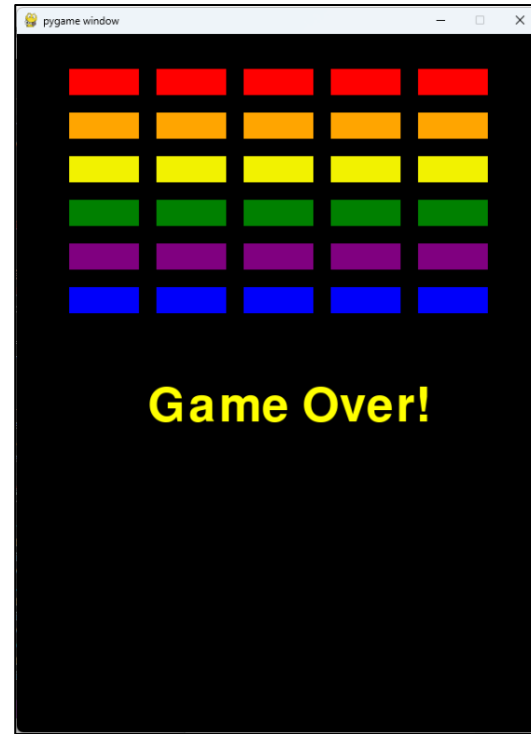
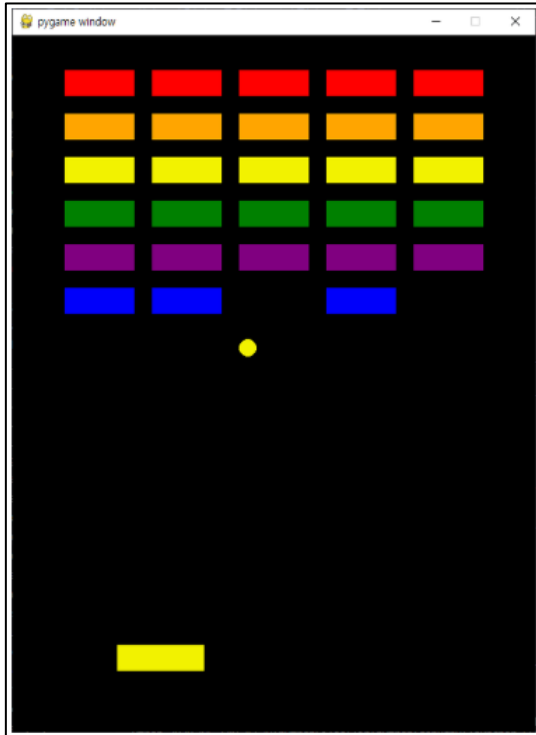


- 뱀 게임

방향키를 움직여 뱀을 움직인다. 먹이를 먹으면 뱀의 꼬리가 하나 늘어난다.
뱀이 벽에 부딪히거나 꼬리와 부딪히면 게임 종료.



05. 블록깨기

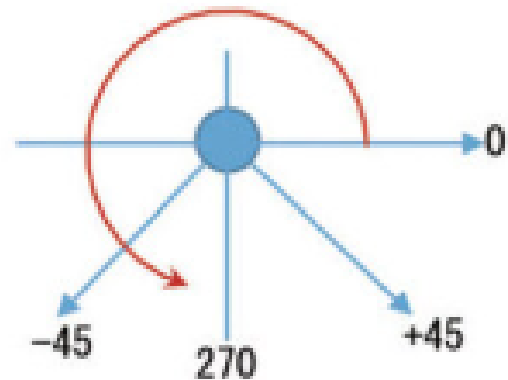
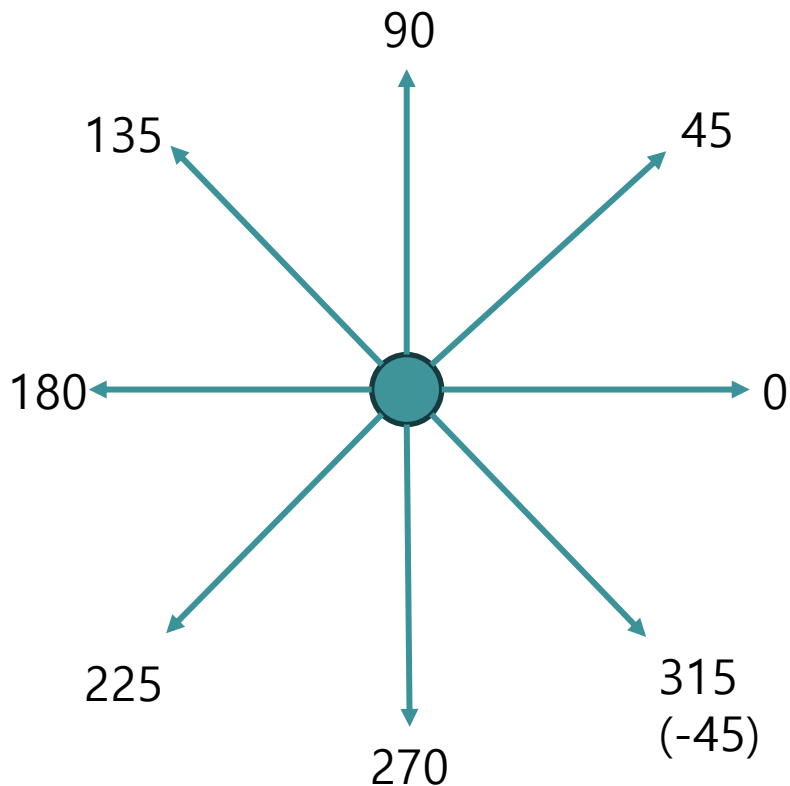


- 블록깨기

방향키를 움직여 패들을 좌우로 움직인다. 공이 패들에 부딪히면 방향을 바꾸고 패들 아래로 떨어지면 게임 종료.

05. 블록깨기

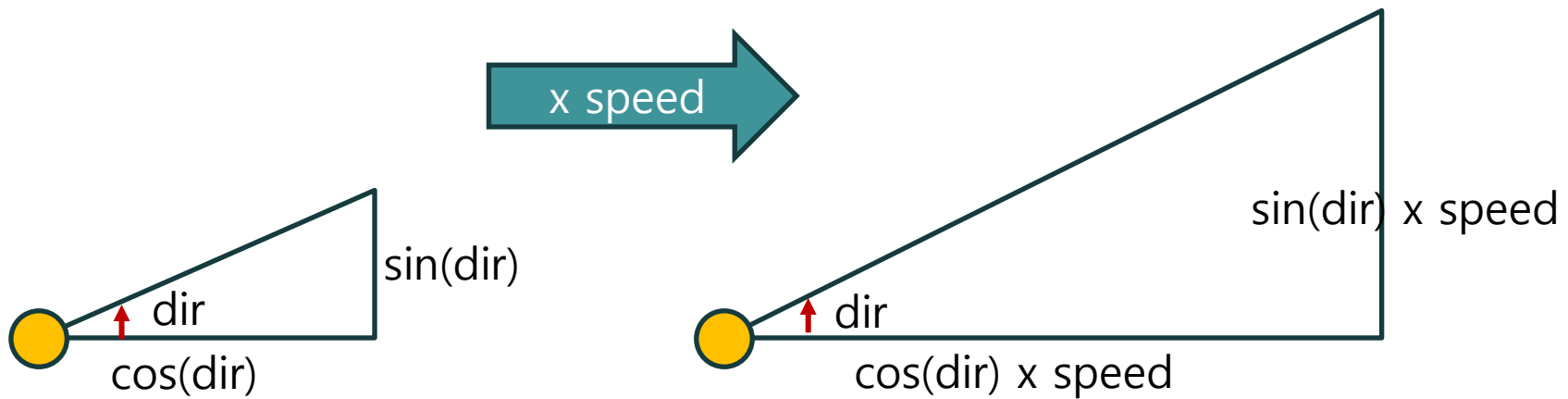
➤ 객체의 움직이는 방향



볼 시작 방향 지정

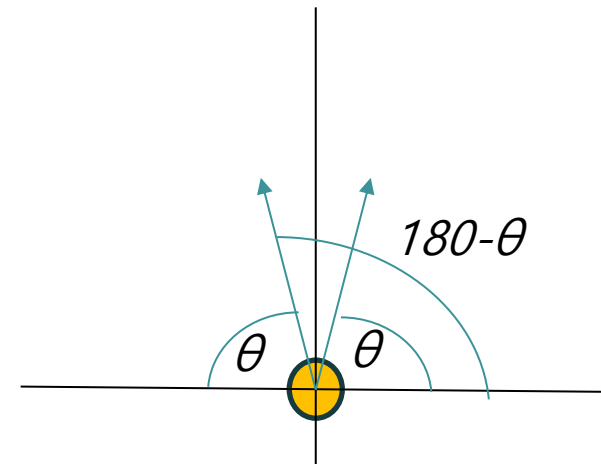
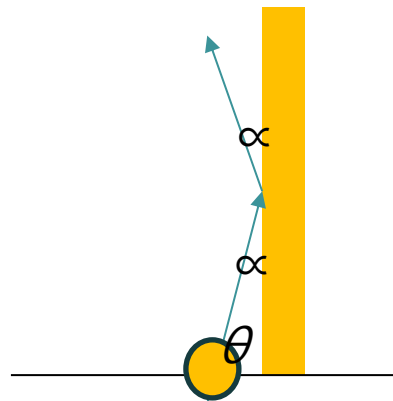
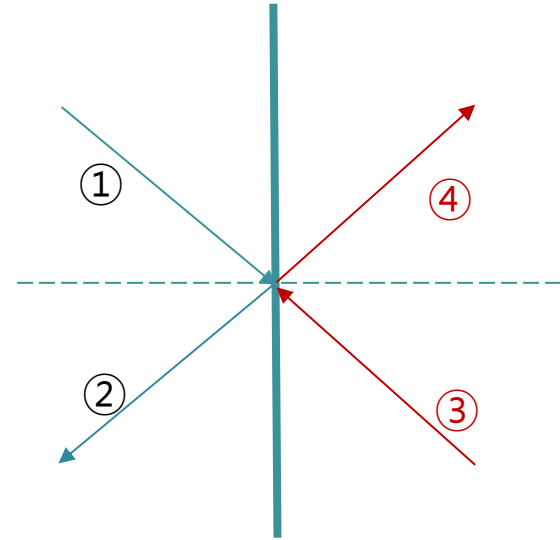
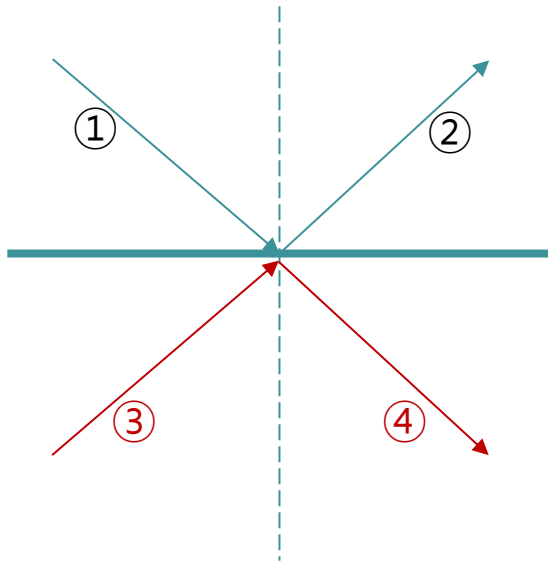
05. 블록깨기

➤ 볼 움직이기

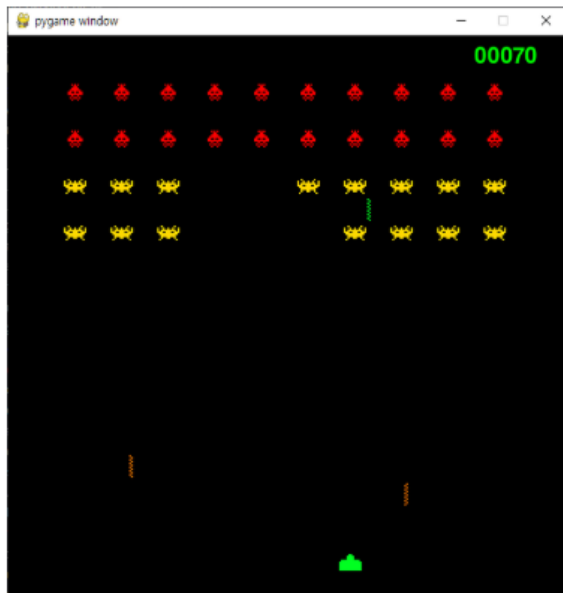


05. 블록깨기

➤ 볼 튕기기



06. 슈팅

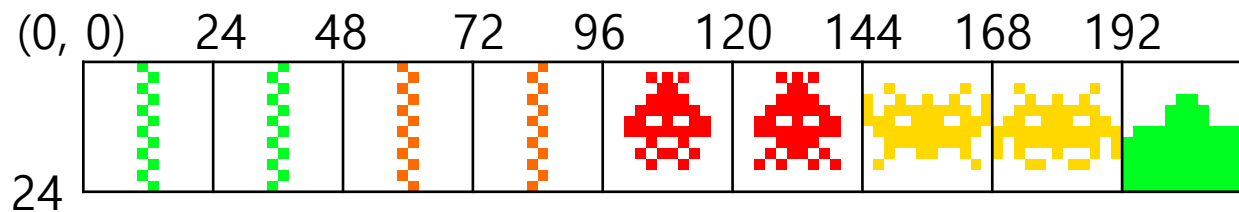
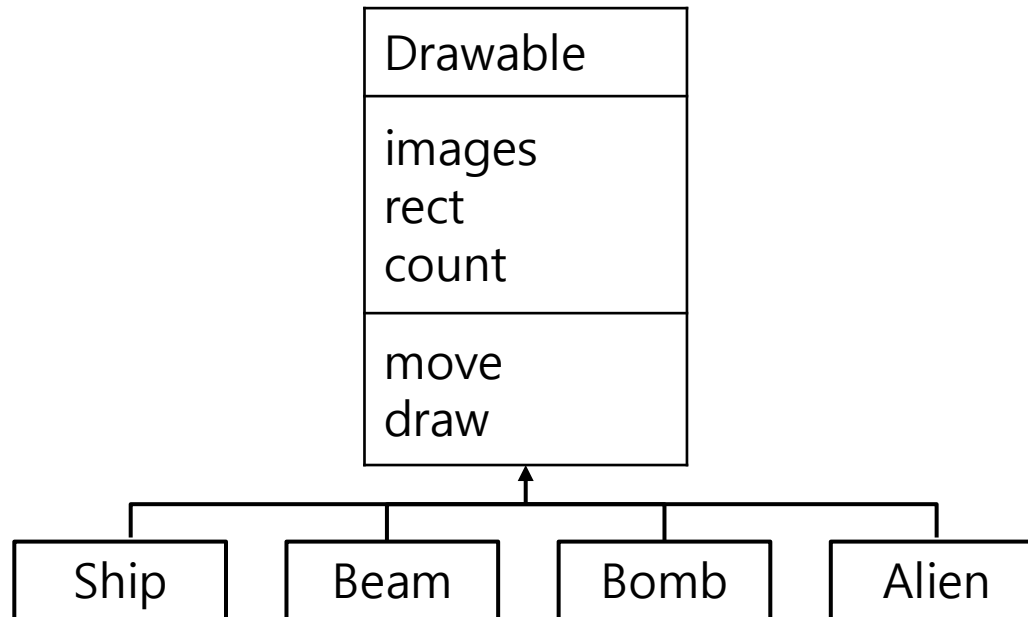


- 슈팅게임

좌우 화살표 키를 움직여 로켓을 움직인다. 스페이스 키를 누르면 미사일을 발사한다. 외계인은 벽에 닿으면 조금씩 아래로 움직인다.



06. 슈팅



06. 슈팅

- image file : strip.png

