

Piano Tile

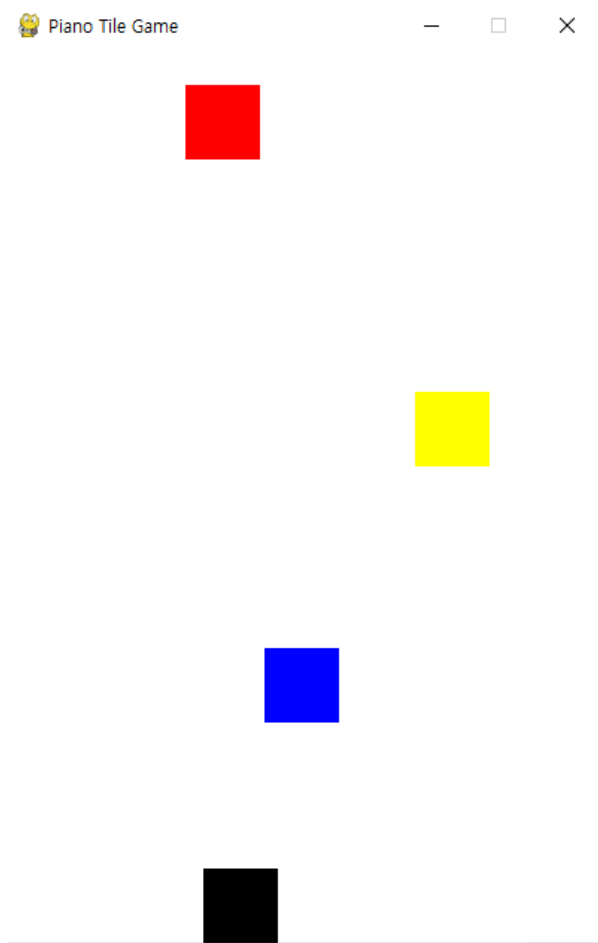
20201109 이가영

Flow of game

Music tile falls-> player move to collide with tile-> music is played

To make it sound like the player is singing, I want background music (MR) to be playing and when the player meets the tiles player can hear the "singing" track.

What I made so far



- Coding
 - player
 - tiles
 - collide

Player

```
# Player
class Player(pygame.sprite.Sprite):
    def __init__(self, screen_width):
        super().__init__()
        self.size = 50
        self.image = pygame.Surface((self.size, self.size))
        self.image.fill((0, 0, 0))
        self.rect = self.image.get_rect()
        self.rect.x = screen_width // 3
        self.rect.y = HEIGHT - self.size
        self.speed = 10
        self.is_colliding = False # New variable to track collision status
```

- Player
→ For now... a black rectangle

Tiles

```
# Tile
class Tile(pygame.sprite.Sprite):
    def __init__(self, screen_width, color):
        super().__init__()
        self.size = 50
        self.image = pygame.Surface((self.size, self.size))
        self.image.fill(color)
        self.rect = self.image.get_rect()
        self.rect.x = random.randint(screen_width // 6, screen_width * 5 // 6 - self.size)
        self.rect.y = random.randint(-self.size * 2, -self.size) # Adjusted initial y-coordinate
        self.speed = 5

    def update(self):
        self.rect.y += self.speed # Move the tile downward
```

- tile

→ Only spawn in the middle (divide screen to 6 columns and use only middle 4)

Main

```
def main():  
    pygame.mixer.music.set_volume(0.0) # Set background music volume to 0  
  
    # Create player  
    player = Player(WIDTH)  
  
    # Create sprite groups  
    all_sprites = pygame.sprite.Group()  
    tiles = pygame.sprite.Group()  
    all_sprites.add(player)  
  
    # Sequence of tiles (0 = Red, 1 = Green, 2 = Blue, 3 = Yellow)  
    tile_sequence = [0, 1, 2, 3, 0, 1, 2, 3] # Adjust the sequence as needed  
    current_tile_index = 0  
    tile_spawn_timer = 0  
    tile_spawn_delay = 20 # Adjust the delay between tile spawns
```

→ Adjust the sequence
of falling tiles

Game loop

```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT] and player.rect.left > WIDTH // 6:
        player.rect.x -= player.speed
    if keys[pygame.K_RIGHT] and player.rect.right < WIDTH * 5 // 6:
        player.rect.x += player.speed

    # Update timer and spawn tiles based on the sequence
    tile_spawn_timer += 1
    if tile_spawn_timer >= tile_spawn_delay:
        color = [RED, GREEN, BLUE, YELLOW][tile_sequence[current_tile_index]]
        new_tile = Tile(WIDTH, color)
        tiles.add(new_tile)
        all_sprites.add(new_tile)
        current_tile_index = (current_tile_index + 1) % len(tile_sequence)
        tile_spawn_timer = 0

    # Update sprites
    all_sprites.update()
    tiles.update()
```

→ Player move only
between middle 4 column

→ Fix color of the tiles

Collide

```
# Update sprites
all_sprites.update()
tiles.update()

# Remove tiles that are out of the screen
tiles = pygame.sprite.Group([tile for tile in tiles if tile.rect.top < HEIGHT])

# Check for collisions with the player
collisions = pygame.sprite.spritecollide(player, tiles, True) #kill block when collide

for tile in tiles:
    if pygame.sprite.collide_rect(player, tile):
        player.is_colliding = True
        pygame.mixer.music.set_volume(1.0) # Set background music volume to 1

    else:
        player.is_colliding = False
        pygame.mixer.music.set_volume(0.0)

# Draw everything
screen.fill(WHITE)
all_sprites.draw(screen)
tiles.draw(screen)

pygame.display.flip()
clock.tick(FPS)
```

→ When tile meet player,
tile is killed

Problem

```
def main():  
    pygame.mixer.music.set_volume(0.0) # Set background music volume to 0
```

```
# Initialize Pygame mixer  
pygame.mixer.init()  
  
# Load sound files  
#hit_sound = pygame.mixer.Sound('hit_sound.wav')  
pygame.mixer.music.load('voice.mp3') #background music
```

```
# Check for collisions with the player  
collisions = pygame.sprite.spritecollide(player, tiles, True) #kill block when collide  
  
for tile in tiles:  
    if pygame.sprite.collide_rect(player, tile):  
        player.is_colliding = True  
        pygame.mixer.music.set_volume(1.0) # Set background music volume to 1  
  
    else:  
        player.is_colliding = False  
        pygame.mixer.music.set_volume(0.0)
```

When player hit tile,
sound played. If not hit
tile, sound not played.

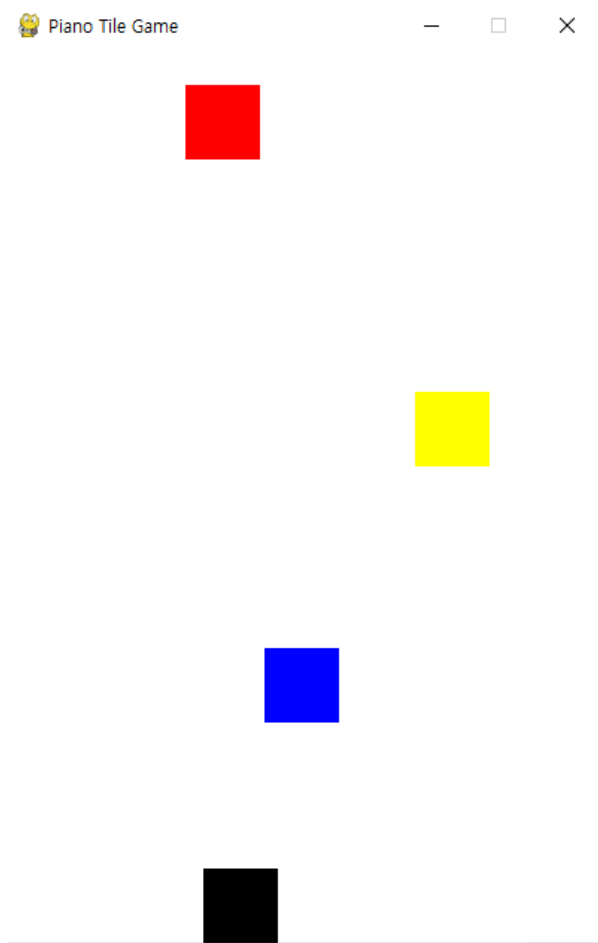
BUT I want the tiles to
play the sound track
when meet player and
go back to mute when
not meet.

I want background music
to be playing from start,
and tiles to act as a
button that turns on the
“singing” track when
meet player

Changes

- Originally wanted to use Phantom of the opera. But cannot find "singing" track and background music track that matches. So I'm trying to use different music.

What I have to do



- Coding
 - make sound when player hit tiles
- Design
 - Background
 - Player
 - notes