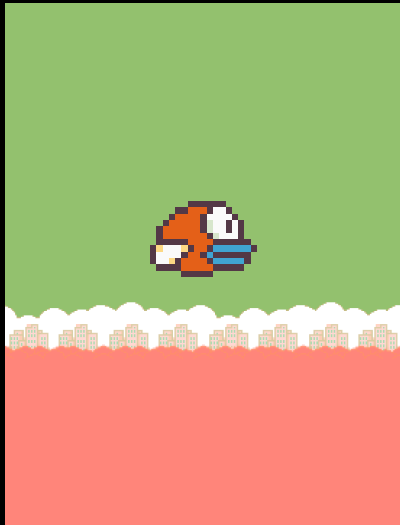
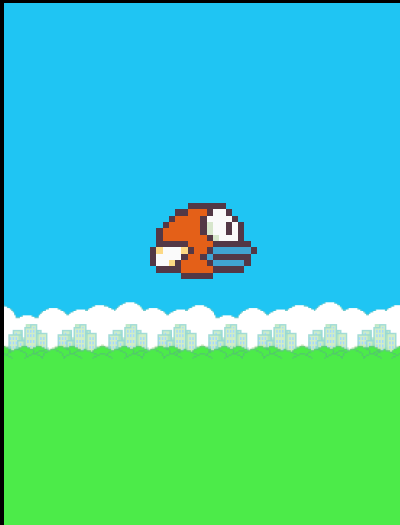


FloppyBird



20191172
함승우

CONCEPT



I'm going to implement flappy bird, an existing cell phone game.

- **Score**

As the game progresses, that is, as long as the bird survives, the score continues to rise. $\text{Score} = \text{Time}$

- **Difficulty**

- As time goes on, the game speed increases in order to increase the difficulty. This will speed up the rate at which obstacles are created

- As time goes on, I will increase the difficulty by changing the color of the obstacles and map to reduce the user's concentration. (Like dinosaur game in chrome)

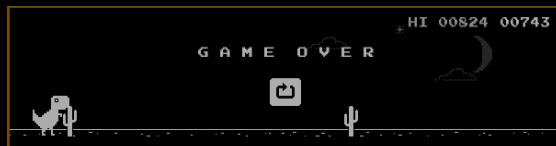
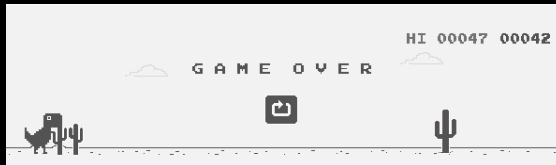


IMAGE & SOUND

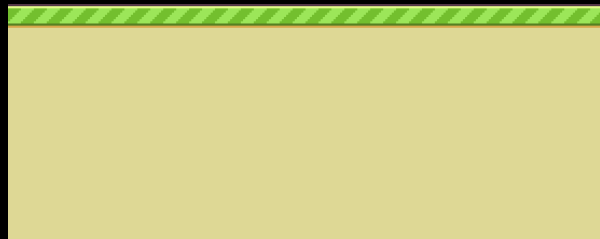
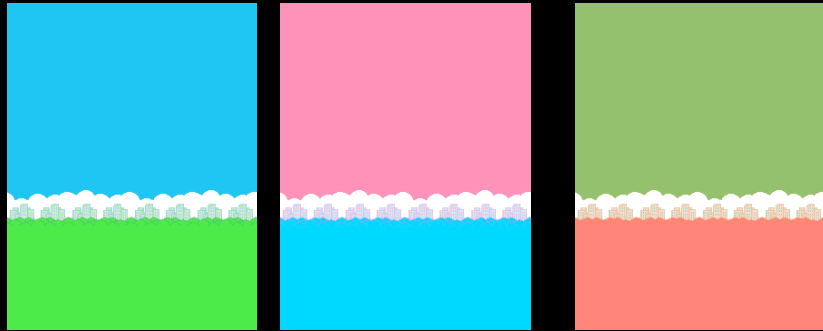
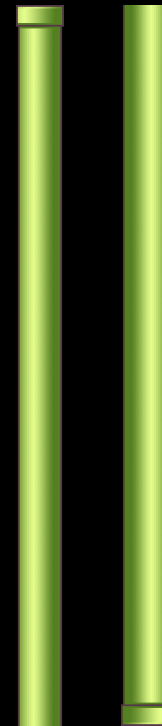


Image: <https://gist.github.com/allenluce/2002be29f52c5848352a9cf8488472be>

Sound: <https://www.mewpot.com/>

Game Over
Press 'R' to Restart

FlappyBird
Press Space to Start



CODE #1

```
1  import pygame
2  from sys import exit
3  import random
4
5  pygame.init()
6  clock = pygame.time.Clock()
7
8  # Window setting
9  WINDOW_HEIGHT = 720
10 WINDOW_WIDTH = 551
11
12 center_W = WINDOW_WIDTH // 2
13 center_H = WINDOW_HEIGHT // 2
14
15
16 screen = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
17
18
19 #색
20 WHITE = (255, 255, 255)
21 BLACK = (0, 0, 0)
```

```
25 #font
26 font = pygame.font.SysFont('arial', 30)
27
28
29 # Images
30 bird_images = [pygame.image.load("assets/bird_1.png"),
31                pygame.image.load("assets/bird_2.png"),
32                pygame.image.load("assets/bird_3.png")]
33
34 ground_image = pygame.image.load("assets/ground.png")
35 top_pipe_image = pygame.image.load("assets/pipe_1.png")
36 bottom_pipe_image = pygame.image.load("assets/pipe_2.png")
37 game_over_image = pygame.image.load("assets/game_over.png")
38 start_image = pygame.image.load("assets/start.png")
39
40 skyline_image_1 = pygame.image.load("assets/background_1.png")
41 skyline_image_2 = pygame.image.load("assets/background_2.png")
42 skyline_image_3 = pygame.image.load("assets/background_3.png")
43
44 #Sound
45 pygame.mixer.music.load('assets/bgm.mp3')
46 pygame.mixer.music.play(-1)
47 jump_sound = pygame.mixer.Sound('assets/jump.mp3')
48 game_over_sound = pygame.mixer.Sound('assets/ending.mp3')
```

CODE #2 (GROUND)

```
57 class Ground(pygame.sprite.Sprite):
58     def __init__(self, x, y):
59         pygame.sprite.Sprite.__init__(self)
60         self.image = ground_image
61         self.rect = self.image.get_rect()
62         self.rect.x, self.rect.y = x, y
63
64     def update(self):
65         #ground moves (game speed)
66         self.rect.x -= game_speed
67         if self.rect.x <= -WINDOW_WIDTH:
68             self.kill()
69
```

For making Ground in Flappy bird

(Line 60 ~ 62): Using sprite because of different properties

(Line 64~68): The x position of the ground image moves to the left according to game_speed.

If ground is completely outside the window, use the kill function

kill()

remove the Sprite from all Groups

kill() -> None

The Sprite is removed from all the Groups that contain it. This won't change anything about the state of the Sprite. It is possible to continue to use the Sprite after this method has been called, including adding it to Groups.

[Search examples for pygame.sprite.Sprite.kill](#)

CODE #3 (BIRD)

```
71 class Bird(pygame.sprite.Sprite):
72     def __init__(self):
73         pygame.sprite.Sprite.__init__(self)
74         self.image = bird_images[0]
75         self.rect = self.image.get_rect()
76         self.rect.center = bird_pos
77         self.image_index = 0
78         self.mov = 0
79         self.alive = True
80
81     def update(self, u_input):
82         # bird's moving
83         if self.alive:
84             self.image_index += 1
85             self.image = bird_images[self.image_index % 3]
86
87         #gravity, bird is moving
88         self.mov += 0.5
89
90         #gravity acts until it touches the ground
91         if self.rect.y < 570:
92             self.rect.y += (self.mov)
93
94         if self.mov > 5:
95             self.mov = 5
96
97         #bird's jump when SPACE is pressed
98         if u_input[pygame.K_SPACE] and self.alive:
99             jump_sound.play()
100             if self.rect.y > 0:
101                 self.mov = -3
```

For making Bird in Flappy bird
3 bird images, these can be used to implement the bird movement

if the bird is alive iterate through the index of the list to make bird move.

Make the bird fall until it touches the ground image.

The fall of the bird does not increase gradually but falls steadily when it reaches a certain value.

When the user presses the space bar, the bird jumps

CODE #4 (PIPE)

```
106 class Pipe(pygame.sprite.Sprite):
107     def __init__(self, x, y, image):
108         pygame.sprite.Sprite.__init__(self)
109         self.image = image
110         self.rect = self.image.get_rect()
111         self.rect.x = x
112         self.rect.y = y
113
114     def update(self):
115         # Move Pipe
116         self.rect.x -= game_speed
117         if self.rect.x <= -WINDOW_WIDTH:
118             self.kill()
119
120     global score
121
122     #counting score
123     if bird_pos[0] > self.rect.topleft[0] and bird_pos[0] < self.rect.topright[0]:
124         score += 1/50
```

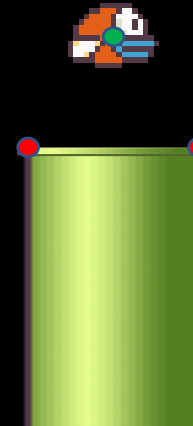
```
x,y
top, left, bottom, right
topleft, bottomleft, topright, bottomright
midtop, midleft, midbottom, midright
center, centerx, centery
size, width, height
w,h
```

For making Obstacle(pipes) in Flappy bird

Make the pipe move to the left as with the ground.

I set the score to increase by 1 when the bird pass through the pipes
lefttop and righttop.

topleft, topright return two integers of tuple.



CODE #5-1 (GAMING)

```
129 def gaming():
130     global score
131
132     # initialize bird, pipes, ground Sprite.
133
134     bird = pygame.sprite.GroupSingle()
135     bird.add(Bird())
136
137     pipe_spawn_time = 0
138     pipes = pygame.sprite.Group()
139
140     #First ground in game
141     x_pos_ground, y_pos_ground = 0, 600
142     ground = pygame.sprite.Group()
143
144     ground.add(Ground(x_pos_ground, y_pos_ground))
```

Set up a sprite group to move bird, pipes, and ground at once

Make first ground in game.

The difference between GroupSingle() and Group() is Group() can hold multiples sprites while GroupSingle() can hold just one multiple sprite.

`pygame.sprite.GroupSingle()`

Group container that holds a single sprite.
`GroupSingle(sprite=None) -> GroupSingle`

The GroupSingle container only holds a single Sprite. When a new Sprite is added, the old one is removed.

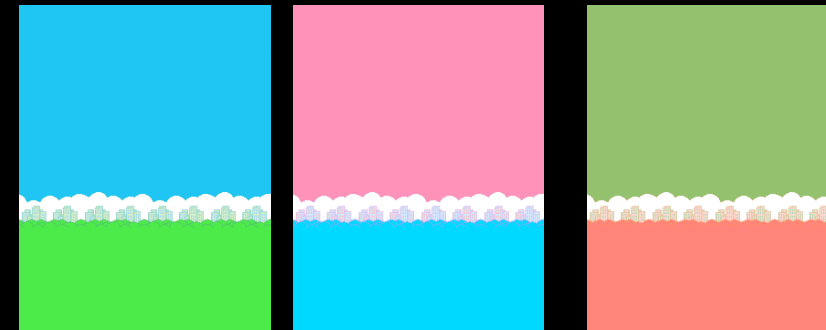
There is a special property, `GroupSingle.sprite`, that accesses the Sprite that this Group contains. It can be None when the Group is empty. The property can also be assigned to add a Sprite into the GroupSingle container.

[Search examples for pygame.sprite.GroupSingle](#)

CODE #5-2 (GAMING)

```
147     run = True
148     while run:
149         for event in pygame.event.get():
150             if event.type == pygame.QUIT:
151                 pygame.quit()
152                 exit()
153
154         screen.fill((0, 0, 0))
155
156         # Making map change
157         score_copy = score
158         if score_copy / 30 > 1:
159             score_copy = score - 30 * round(score_copy / 30)
160
161         if 0 <= score_copy < 10:
162             screen.blit(skyline_image_1, (0, 0))
163         if 10 <= score_copy < 20:
164             screen.blit(skyline_image_2, (0, 0))
165         if 20 <= score_copy < 30:
166             screen.blit(skyline_image_3, (0, 0))
167
```

Based on the score set in the pipe class, the sections were divided onto 0~10 points, 11~20 points, and 21~30 points, so that different background images were used for each section.



CODE #5-3 (GAMING)

```
177     u_input = pygame.key.get_pressed()
178
179     pipes.draw(screen)
180
181     #Score interface
182     score_text = font.render(f'SCORE: {round(score)}', True, WHITE)
183     screen.blit(score_text, (20, 20))
184
185     ground.draw(screen)
186     bird.draw(screen)
187
188     if bird.sprite.alive:
189         pipes.update()
190         ground.update()
191         bird.update(u_input)
```

Above we draw the elements of the game and display the score on the screen.

And keep it updated while the game is running(while the bird is alive).

The update of the bird, that is, the jump, is updated with keyboard input.

CODE #5-4 (GAMING)

```
195 #Game over
196 colli_pipes = pygame.sprite.spritecollide(bird.sprites()[0], pipes, False)
197 colli_ground = pygame.sprite.spritecollide(bird.sprites()[0], ground, False)
198
199 if colli_pipes or colli_ground:
200     #bird die and game stop
201     bird.sprite.alive = False
202
203     #game over interface
204     screen.blit(game_over_image, (center_W - game_over_image.get_width() // 2,
205                                     center_H - game_over_image.get_height() // 2))
206     game_over_sound.play()
207     if u_input[pygame.K_r]:
208         #restart interface
209         score = 0
210         break
211
212 # making pipes
213 if pipe_spawn_time <= 0 and bird.sprite.alive:
214     x_top, x_bottom = 700, 700
215     y_top = random.randint(-800, -480)
216     y_bottom = y_top + random.randint(80, 130) + bottom_pipe_image.get_height()
217     pipes.add(Pipe(x_top, y_top, top_pipe_image))
218     pipes.add(Pipe(x_bottom, y_bottom, bottom_pipe_image))
219     pipe_spawn_time = random.randint(60, 180)
220 pipe_spawn_time -= 1
```

By using `spritecollide`, the case where a bird collides with a pipe or ground is identified.

The game loses(the bird dies, `Alive = False`) accordingly, and the game over interface appears

Then press `r` to restart it.

After figuring out the size of the top image and bottom image of the pipes, pipes in random positions were spawned according to the random `pipe_spawn_time`.

`pipe_spawn_time` indicates the interval between pipes.

`pygame.sprite.spritecollide()`

Find sprites in a group that intersect another sprite.

`spritecollide(sprite, group, dokill, collided = None) -> Sprite_list`

Return a list containing all Sprites in a Group that intersect with another Sprite. Intersection is determined by comparing the `Sprite.rect` attribute of each Sprite.

The `dokill` argument is a bool. If set to `True`, all Sprites that collide will be removed from the Group.

The `collided` argument is a callback function used to calculate if two sprites are colliding. it should take two sprites as values, and return a bool value indicating if they are colliding. If `collided` is not passed, all sprites must have a "rect" value, which is a rectangle of the sprite area, which will be used to calculate the collision.

collided callables:

```
collide_rect, collide_rect_ratio, collide_circle,
collide_circle_ratio, collide_mask
```

Example:

```
# See if the Sprite block has collided with anything in the Group block_list
# The True flag will remove the sprite in block_list
blocks_hit_list = pygame.sprite.spritecollide(player, block_list, True)

# Check the list of colliding sprites, and add one to the score for each one
for block in blocks_hit_list:
    score += 1
```

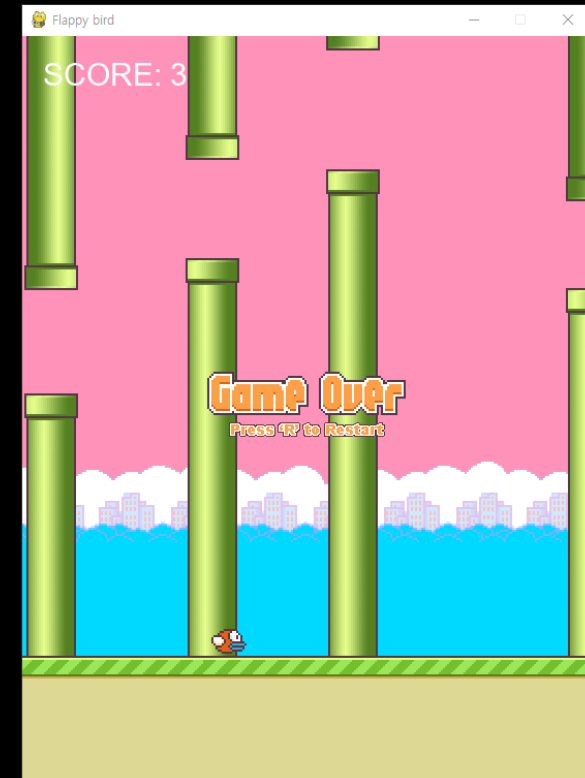
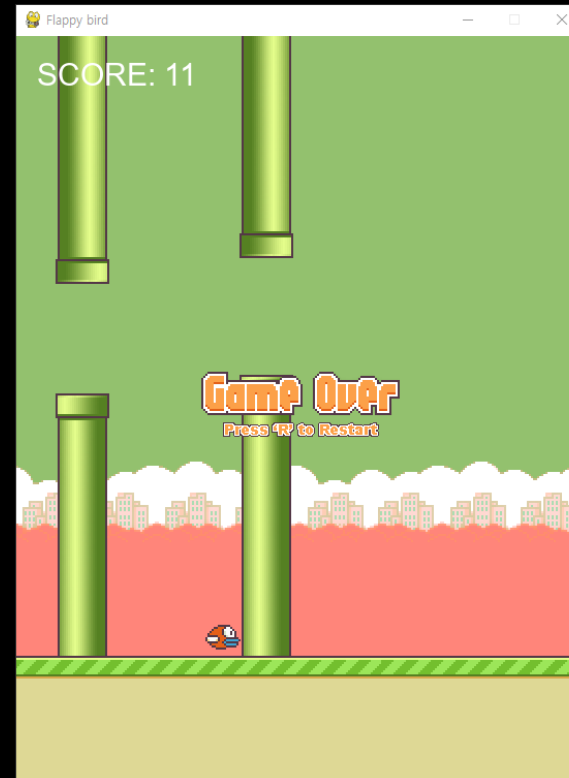
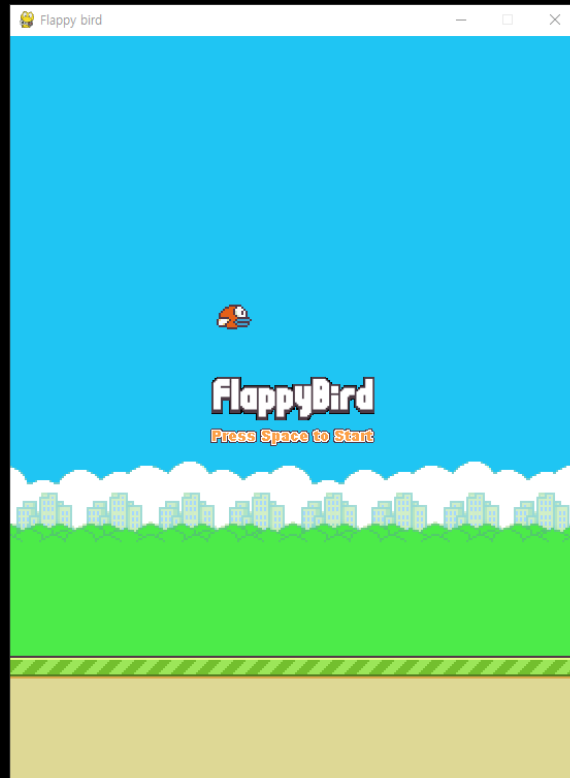
CODE #6

Create Game Start Interface

```
226 while game_over:
227     for event in pygame.event.get():
228         if event.type == pygame.QUIT:
229             pygame.quit()
230             exit()
231
232     # Game start interface
233     screen.fill((0, 0, 0))
234     screen.blit(skyline_image_1, (0, 0))
235     screen.blit(ground_image, Ground(0, 600))
236     screen.blit(bird_images[0], (200, 260))
237     screen.blit(start_image, (center_W - start_image.get_width() // 2,
238                               center_H - start_image.get_height() // 2))
239
240     # keyboard input
241     u_input = pygame.key.get_pressed()
242
243     if u_input[pygame.K_SPACE]:
244         gaming()
245
246
247     pygame.display.update()
248
```

RESULT

https://github.com/Seung20/Final_project_fl



CONCLUSION



1. I tried to adjust the difficulty of the game by increasing the game speed, but the bug appeared that slowed down the whole game speed, so I adjusted the difficulty by reducing the spacing of the pipes randomly.
2. it is unfortunate that the position of the gap between the top pipe and the bottom pipe is randomly set, creating impassable obstacle.
3. Also, inspired by the chrome dinosaur game, we tried to reduce the user's concentration by changing the color of the map or background, but the effect was insignificant, so I changed the concept to reach a different map according to the score.