

Algorithm and Programming

Lecturer: Jude Joseph Lamug Martinez, MCS

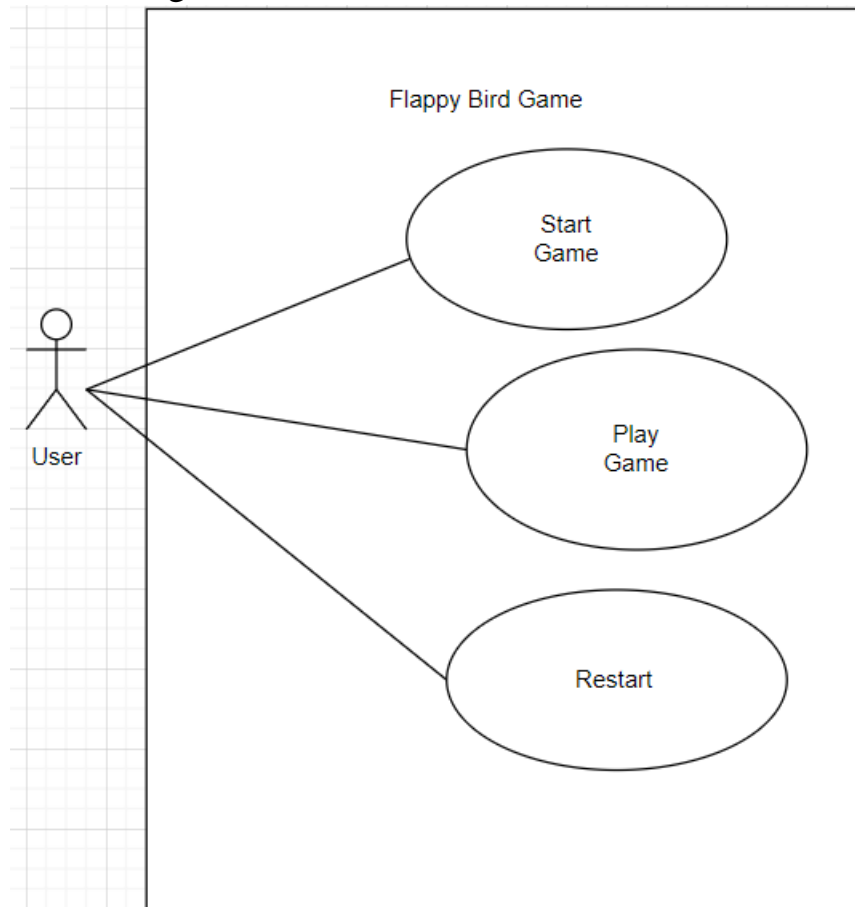
By: Louis Ruby Elsalim (2602185431)

A. Project Documentation

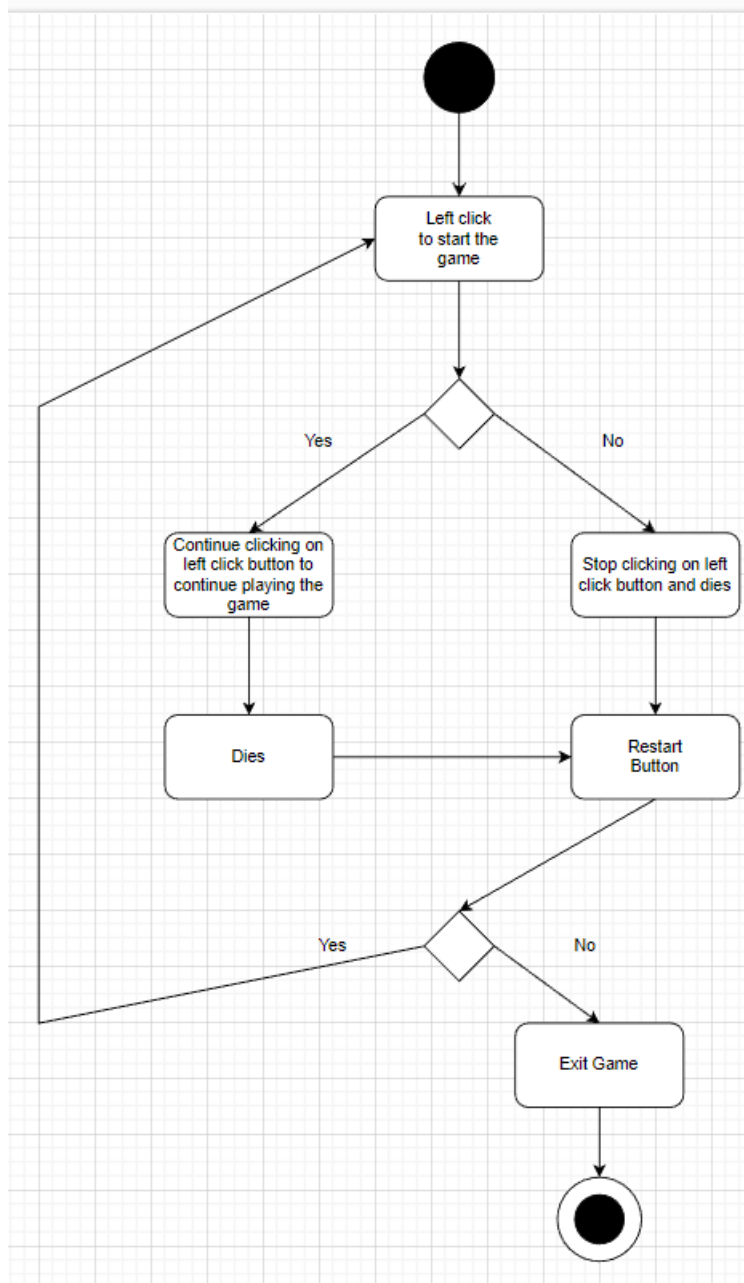
1. Brief Description

- Flappy bird is a popular game back in 2013, it went viral when players starts to turn it into a meme, where players are getting frustrated over the game. Even with a very simple concept, it was once a top 10 game in the world. Sadly, they take down the game back in 2014. Basically, the concept of the game is that you have to make the bird pass through the pipes in order to continue playing the game. The bird ascends each time you clicked on the screen or a mouse. Without clicking the screen or the mouse, the effects of gravity are applied to the bird. Each time the bird passes one pipe, you subsequently earn one point. The game ends once the bird collides with the pipe. You have to restart the game and lose all your hard-earned points.

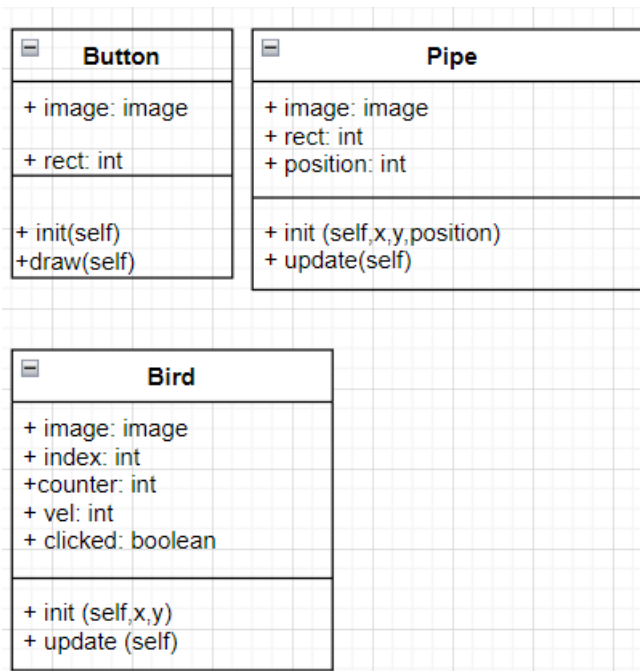
2. Use-Case Diagram



3. Activity Diagram



4. Class Diagram



5. Modules

- Pygame: set of python modules set for writing video games.
- * from pygame.locals: copies all names in pygame.locals into your current namespace.
- Random: built in module used for randomize number.

6. Essential Algorithms

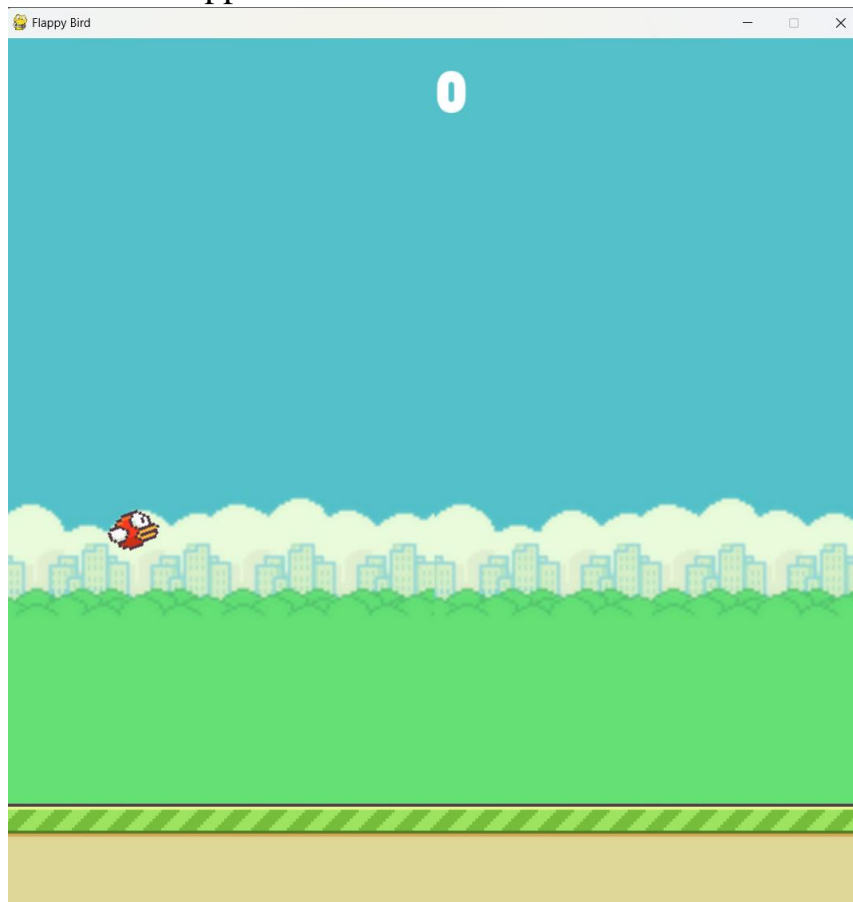
- With the help of some classes and functions, this are the algorithms to run the game.

```

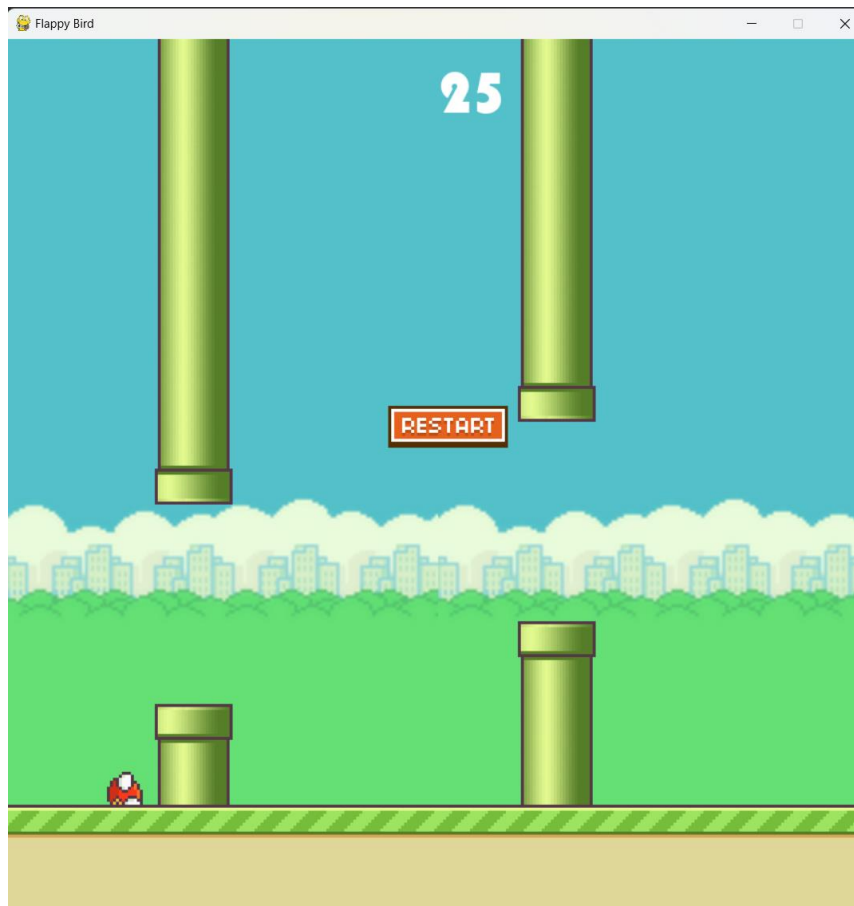
Go Run Terminal Help
flappy_bird-main
flappy_bird_tut1.py flappy.py x flappy_bird_tut5.py flappy_bird_tut6.py
flappy_bird-main > flappy.py > ...
170
171 #To get the game starting
172 run = True
173 while run:
174
175     #to make everything runs as we want, and not as fast as possible
176     clock.tick(fps)
177
178     #draw background
179     screen.blit(bg, (0,0))
180
181     pipe_group.draw(screen)
182     bird_group.draw(screen)
183     bird_group.update()
184
185     #draw and scroll the ground
186     screen.blit(ground_img, (ground_scroll, 768))
187
188     #check the score
189     if len(pipe_group) > 0:
190         if bird_group.sprites()[0].rect.left > pipe_group.sprites()[0].rect.left\
191             and bird_group.sprites()[0].rect.right < pipe_group.sprites()[0].rect.right\
192             and pass_pipe == False:
193             pass_pipe = True
194         if pass_pipe == True:
195             if bird_group.sprites()[0].rect.left > pipe_group.sprites()[0].rect.right:
196                 score += 1
197                 pass_pipe = False
198     draw_text(str(score), font, white, int(screen_width / 2), 20)
199
200
201     #look for collision
202     if pygame.sprite.groupcollide(bird_group, pipe_group, False, False) or flappy.rect.top < 0:
203         game_over = True
204     #once the bird has hit the ground it's game over and no longer flying
205     if flappy.rect.bottom >= 768:
206         game_over = True
207         flying = False
  
```

```
go Run Terminal Help ← → flappy_bird-main
flappy_bird-main > flappy.py x flappy_bird_tut5.py flappy_bird_tut6.py
flappy_bird-main > flappy.py > ...
210 if flying == True and game_over == False:
211     #generate new pipes
212     time_now = pygame.time.get_ticks()
213     if time_now - last_pipe > pipe_frequency:
214         pipe_height = random.randint(-100, 100)
215         btm_pipe = Pipe(screen_width, int(screen_height / 2) + pipe_height, -1)
216         top_pipe = Pipe(screen_width, int(screen_height / 2) + pipe_height, 1)
217         pipe_group.add(btm_pipe)
218         pipe_group.add(top_pipe)
219         last_pipe = time_now
220
221     pipe_group.update()
222
223     ground_scroll -= scroll_speed
224     #reset ground img
225     if abs(ground_scroll) > 35:
226         ground_scroll = 0
227
228
229 #check for game over and reset
230 if game_over == True:
231     if button.draw():
232         game_over = False
233         score = reset_game()
234
235
236 for event in pygame.event.get():
237     if event.type == pygame.QUIT:
238         run = False
239     if event.type == pygame.MOUSEBUTTONDOWN and flying == False and game_over == False:
240         flying = True
241
242 #update background
243 pygame.display.update()
244
245 pygame.quit()
246
```

7. Screenshots of application



- Before the game starts



- When the game ends

8. Lesson learned/Reflection

- Within the span of these few months doing this project, I have learnt a lot regarding pygame and python such as the built-in functions and a better grasp of the loops. I also realise that I have poor time management, I felt pressured around the deadline of the submission date since I still have so many works, I have to work on.