

**ALGORITHM AND PROGRAMMING
FINAL PROJECT**



WRITTEN

BY

**NAMA : AZZA NASIMA
CLASS : L1BC
MAJOR : COMPUTER SCIENCE
DATE : 15 JANUARY 2023**

**BINUS INTERNATIONAL UNIVERSITY
CENTRAL JAKARTA
2023**

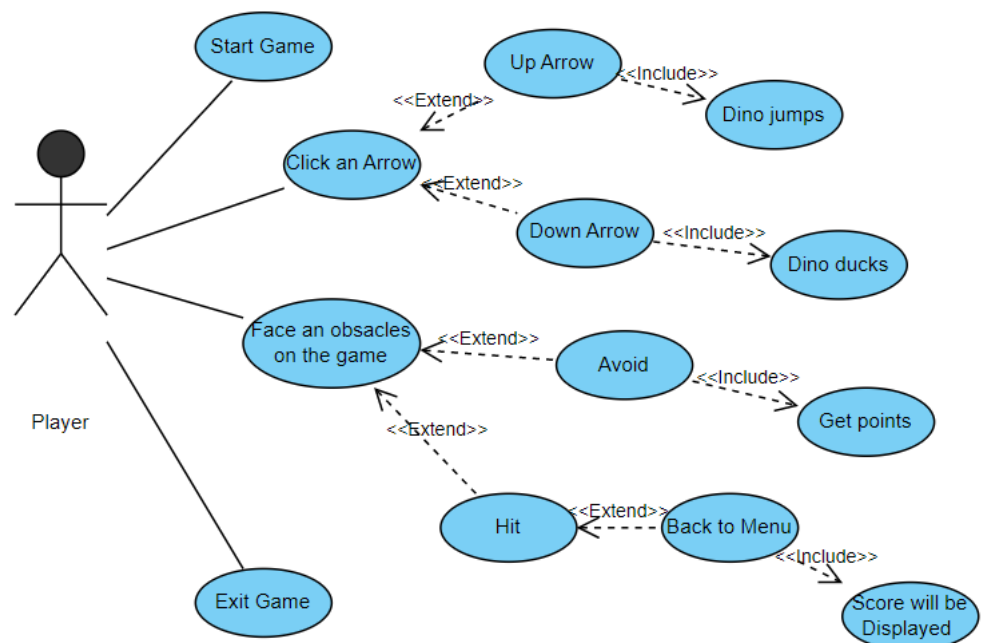
A. PROJECT DOCUMENTATION

1. Brief Description

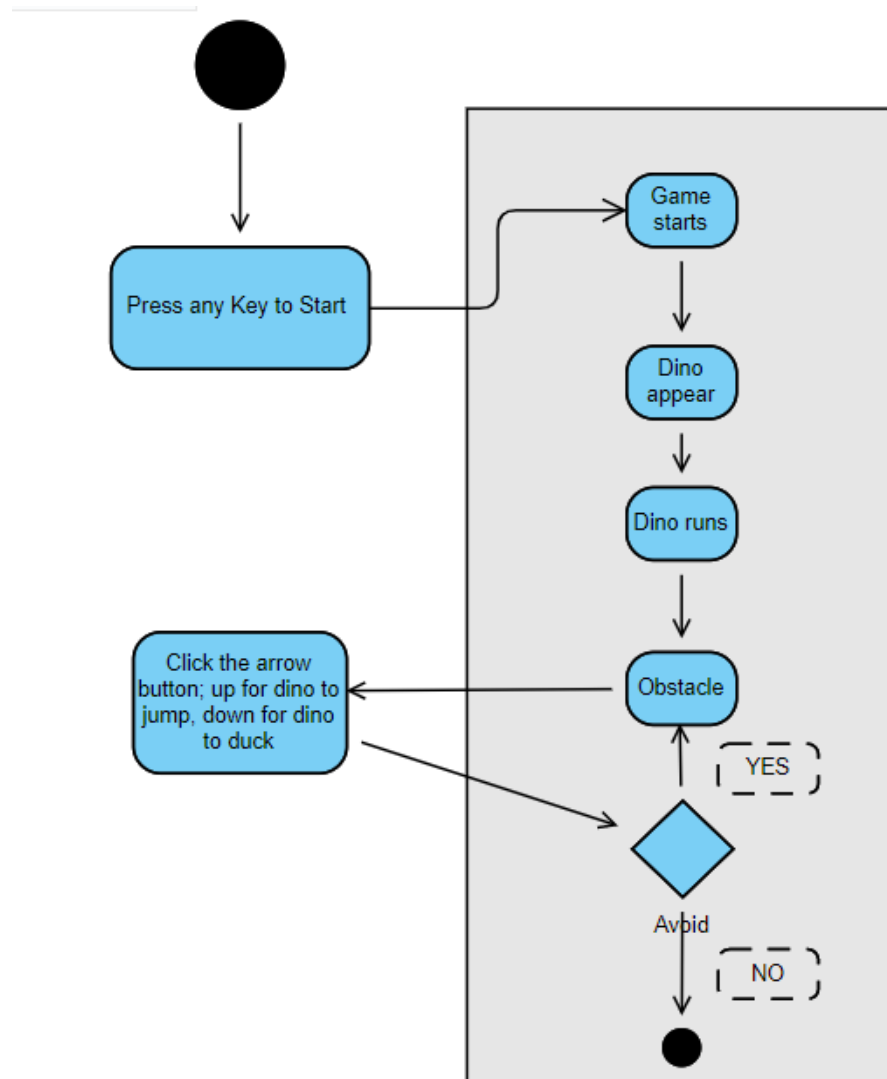
For the final project of the Algorithm and Programming class, I am recreating the T-rex Dinosaur Game, the hidden game from Chrome offline mode, which I used to play whenever there was no internet connection. You only need to press the up and down arrows on your computer to jump and duck, respectively, making the game relatively easy to understand. I obtained the Assets from Google, so my game would be similar to the original game.

Button	In game
Any Key	Start the game
Up Arrow	Jump
Down Arrow	Duck

2. Use-case diagram

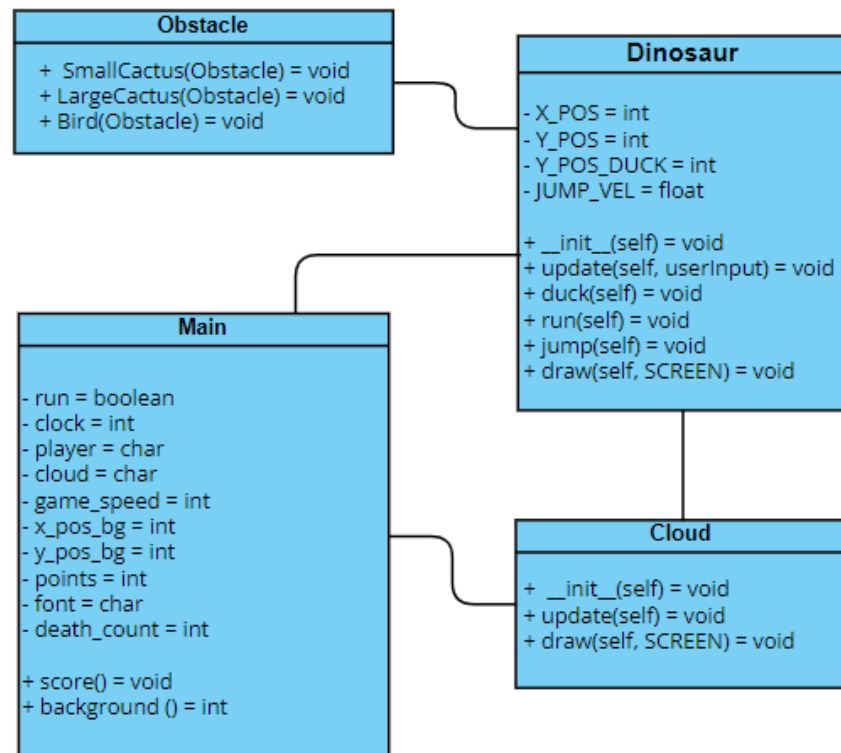


3. Activity Diagram



- The grey area shown above is for the activity diagram in the game, while outside the box is the activity diagram for the player

4. Class Diagram - the name of the class, multiplicities, and relationships



5. Modules (list down and short description)

- Import pygame: to import all the available pygame modules into the pygame package, and the pygame init() is used to initialize all imported pygame modules.
- Import os: provides the facility to establish the interaction between the user and the operating system
- Import random: to generate random numbers, a random module can also use it to perform random actions such as generating random numbers and printing random values for a list or string.
- Import sys: provides the name of the existing python modules which have been imported

6. Essential algorithms

The two pictures I attached below are the essential algorithms on my code since I am making a t-rex dinosaur game; without the dinosaur class, my game wouldn't exist. Moreover, the obstacle class is also necessary to make the game more exciting. Otherwise, the game will only consist of a dinosaur running.

```
class Dinosaur:
    X_POS = 80 # x position of the dinosaur on the screen
    Y_POS = 310 # y position of the dinosaur on the screen
    Y_POS_DUCK = 340 # when ducking, the dinosaur need to be shown further down the screen
    JUMP_VEL = 8.5 # velocity of the dinosaur as soon as it jumps

    def __init__(self): # to initialize the dinosaur whenever an object of this class is created
        # innit method is used to include all the images of the dinosaur
        self.duck_img = DUCKING
        self.run_img = RUNNING
        self.jump_img = JUMPING

        self.dino_duck = False
        self.dino_run = True # the dinosaur neither ducking nor jumping, it's simply running on the road
        self.dino_jump = False

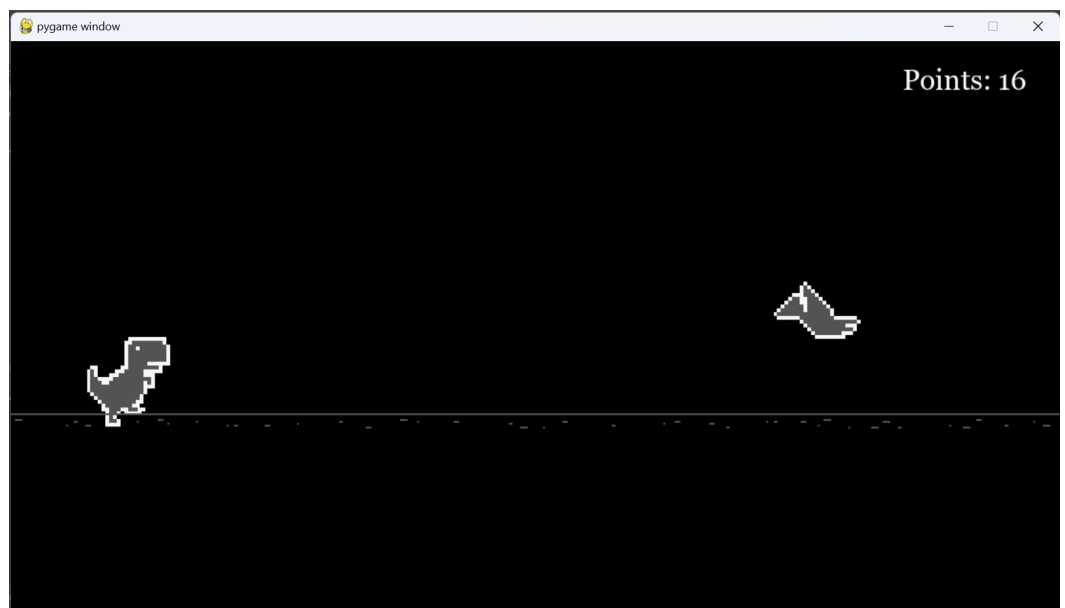
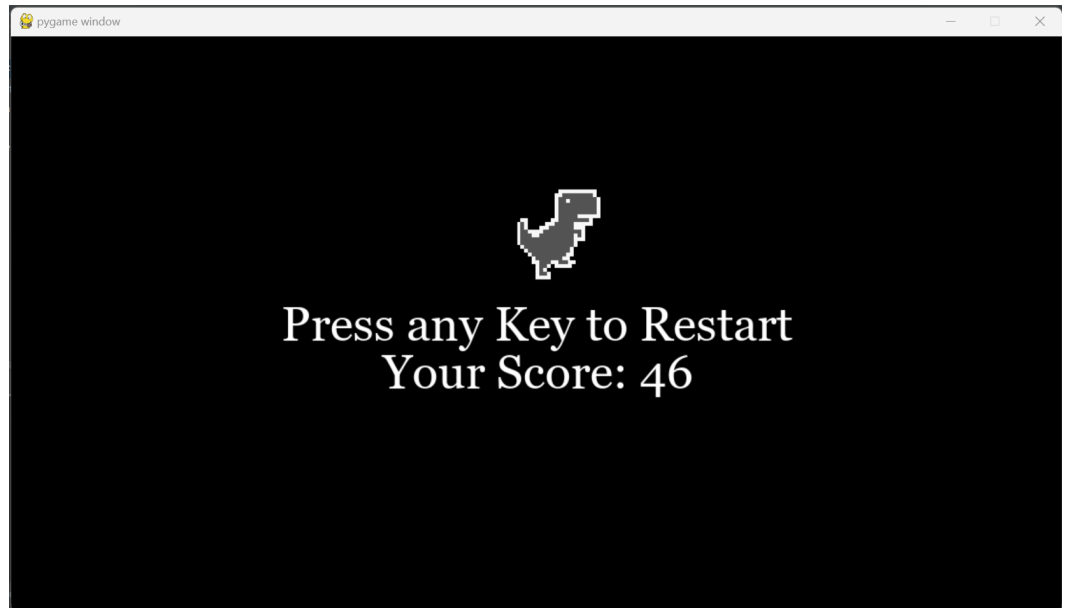
        self.step_index = 0 # to animate the dinosaur
        self.jump_vel = self.JUMP_VEL
        self.image = self.run_img[0] # initialize the first image
        # to get the rectangle of the dinosaur image:
        self.dino_rect = self.image.get_rect() # which is going to be the hit boxes for the dinosaur later on
        self.dino_rect.x = self.X_POS # the x coordinate for the rectangle
        self.dino_rect.y = self.Y_POS # the y coordinate for the rectangle
```

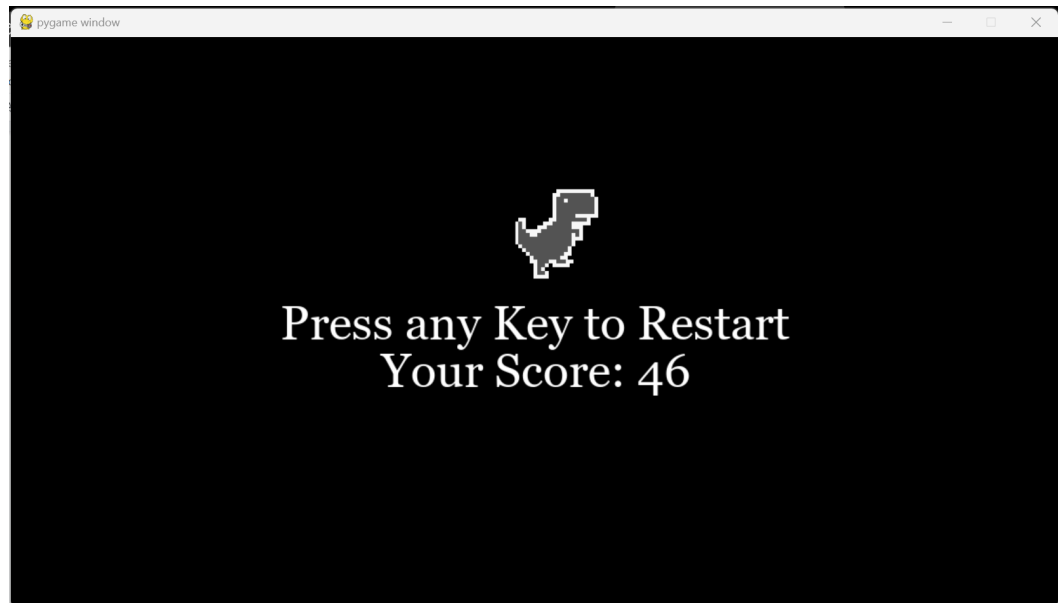
```
class Obstacle:
    def __init__(self, image, type):
        self.image = image # image of the obstacles
        self.type = type # the type of the obstacles in integer value
        self.rect = self.image[self.type].get_rect() # rect coordinates of the img
        self.rect.x = SCREEN_WIDTH

    def update(self):
        self.rect.x -= game_speed
        if self.rect.x < -self.rect.width: # moves the obstacles off the screen on the left hand side
            obstacles.pop()

    def draw(self, SCREEN):
        SCREEN.blit(self.image[self.type], self.rect) # blit the image onto the screen
```

7. Screenshots of your application





8. Lessons learned/Reflection

Although this game is still considered simple to create in Python, at the end of the day, I learned how to create a game through this experience. I gave a lot of thought to whatever game I should make for my final project before committing to it. Before developing one, there was some trial and error, as well as opening numerous Google tabs and YouTube videos as sources of inspiration. I rushed the process, which was one thing I regret the most. If I had started working on this final project a few weeks earlier, I might have been able to create something more complex.

Due to my limited programming knowledge, I had trouble with some of the codes while creating this game. Additionally, making a game is more difficult than it first appears. Sometimes we may overlook a minor detail, causing the software to crash or fail to load. But since the game I made is still seen as being simple, I didn't encounter many difficulties. Despite my struggles, I quickly came up with the answer. Last but not least, if given a chance, I would like to make the game I created more like the original game. For example, I might place a high score button next to the score or game-over symbols. Nonetheless, I am happy my game functions well.

B. PROGRAM/SOURCE CODE

https://github.com/azzansm/AP_FE.git

C. 3-5 MINUTE VIDEO DEMO

<https://youtu.be/yxxJHDHsuDQ>