⌨️

Typing Speed Test

@June 29, 2022

Documentation of Project:

In this video, I plan to create a speed typing test using python. I will work through “Pygame” for this project which is a **python programming language library for making multimedia applications for the interface**. I will also use system, random and time modules. Firstly, I will show how the test looks like during execution by running my code. I have already the code written for all the essential functions and will record a time-lapse of me writing the code for the run (main) function, and will explain t

he program in that order. I will end the video by once again running the program to show how the program works.

Outline for Presentation:

5 minute video

📜 Presentation

Code -

```
def run(self):
        self.reset_game()

        self.running = True
        while (self.running == True):
            clock = pygame.time.Clock()
            self.screen.fill((0, 0, 0), (50, 250, 650, 50))
            pygame.draw.rect(self.screen, (255, 0, 100), (70, 250, 650, 50), 3)
            # update the text of user input
            self.display(self.screen, self.input, 274, 26, (250, 250, 250))
            pygame.display.update()
            for event in pygame.event.get():
```

```
                if event.type == QUIT:
                    self.running = False
                    sys.exit()

                elif event.type == pygame.MOUSEBUTTONUP:
                    x, y = pygame.mouse.get_pos()
                    # position of input box
                    if (x >=70 and x <=650 and y>=250 and y<=300):
                        self.active = True
                        self.input = ''
                        self.start = time.time()
                        # position of reset box
                    if (x >=400 and x <=500 and y>=450 and self.end):
                        self.reset_game()
                        x, y = pygame.mouse.get_pos()


                elif event.type == pygame.KEYDOWN:
                    if self.active and not self.end:
                        if event.key == pygame.K_RETURN:
                            print(self.input)
                            self.results(self.screen)
                            print(self.output)
                            self.display(self.screen, self.output, 350, 28, (255, 70, 70))
                            self.end = True

                        elif event.key == pygame.K_BACKSPACE:
                            self.input = self.input [:-1]
                        else:
                            try:
                                self.input += event.unicode
                            except:
                                pass

            pygame.display.update()

        clock.tick(50)
```