Hindi Vidya Prachar Samiti’s

RAMNIRANJAN JHUNJHUNWALA COLLEGE

Ghatkopar (W), Mumbai-400 086



PIXEL PLATFORMER

(a Python based Platformer game)

Mini Project

First Year Bachelor Of Science-Information Technology

|  |  |
| --- | --- |
| NAME | ROLL |
| RAJBHAR SUDESH DINESH | 066 |
| SHINDE OMKAR BABURAO | 080 |

Under the Guidance of

-Punam Sindhu

Index

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no** |
| 1. | Introduction. | 3 |
| 2. | Language of the game. | 3 |
| 3. | Pygame module | 4 |
| 4. | The Game | 5 |
| 5. | The website | 13 |
| 6. | conclusion | 19 |

**Introduction :-**

**Platform games** (often simplified as **platformer** or **jump 'n' run**) is a video game genre and subgenre of action games in which the core objective is to move the player character between points in a rendered environment.

Other acrobatics such as climbing, swinging from objects such as vines or grappling hooks, jumping off walls, air dashing, gliding through the air, being shot from cannons, or bouncing from springboards or trampolines. These mechanics, even in the context of other genres, are commonly called *platforming.*

The most common unifying element of games of this genre is the ability to jump.

**Language of the game (back-end) :-**

This game is developed using python.

There are various frameworks which help in developing games like Pygame.

In this game, we have used Pygame framework. We have a character/pixel in the game, and he must cross all the obstacles in the way using the arrow keys.

The player has the control of the character, but the obstacles cannot be controlled by the player. We have used arrow keys to move the character back and forth and jump when necessary.

**Pygame Module:-**

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

There is basic formula from classical mechanics to make an object jump.

**F = 1/2 \* m \* v^2**

Where F is the force up/down, m is the mass of the object and v is the velocity. The velocity goes down over time because when the object jumps the velocity will not increase more in this simulation.

When object reaches the ground, the jump ends. If is jump variable is True or False it indicates object is jumping or not. If is jump is True, object position will be updated according to the above formula.



**The Game:-**



The “Pixel Platformer “game is an entry level game directed towards the kids.

The main objective of this game is to surpass all the obstacles coming in the way and reach to the top. If the player misses with any platform, the game gets over and player must start again.

The player can surpass the obstacles by jumping over it with the arrow key controls.

We have made this game very fun to play. This is a very easy game , and people of any age group can play this game, but more emphasized towards the age group of 5-13.

**MAIN ASPECTS OF THE GAME:-**

1.Game Environment.

2.Collision Detection.

3.Player movement.

4.Jump mechanics

5.Gravity mechanics

6.Scrolling of screen

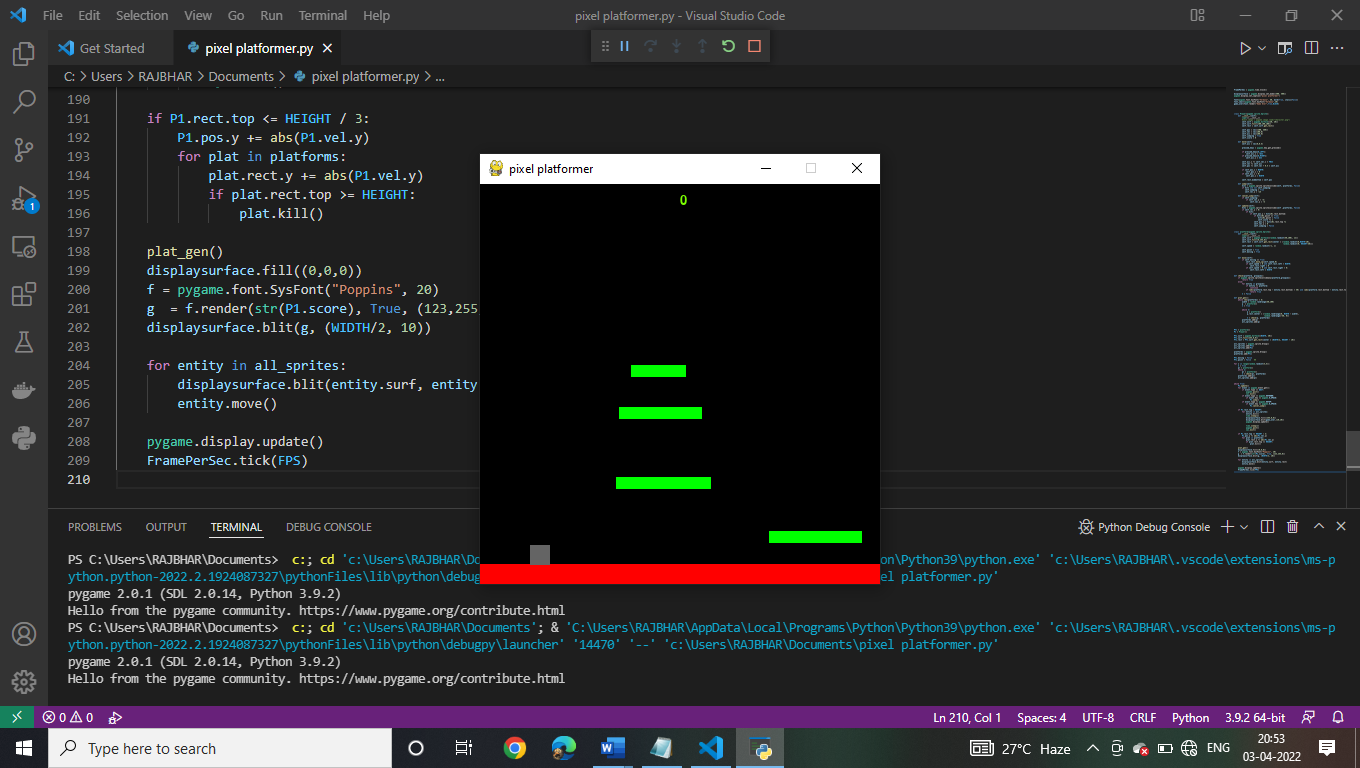
7.score counter

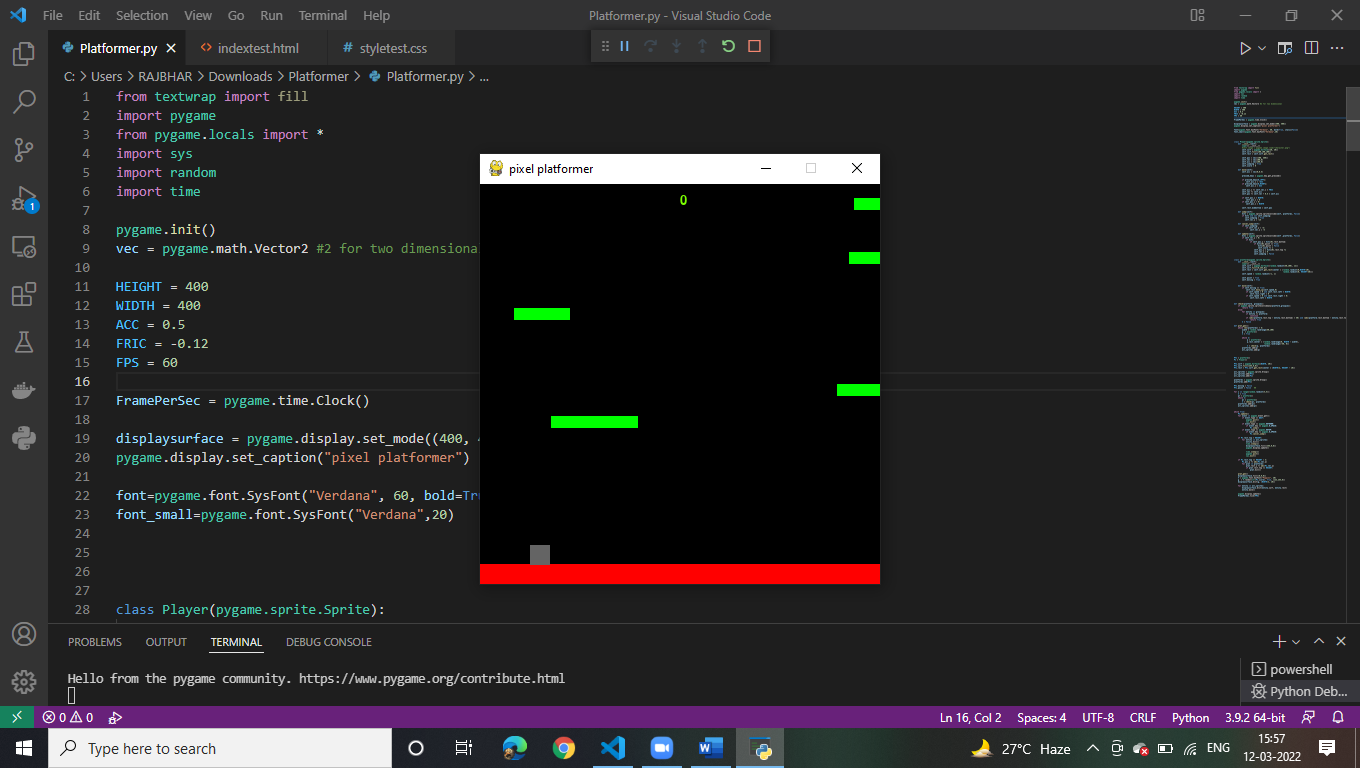
8.Game over

9.Random platform movement

Input Code:

|  |
| --- |
| from textwrap import fill  import pygame  from pygame.locals import \*  import sys  import random  import time    pygame.init()  vec = pygame.math.Vector2 #2 for two dimensional    HEIGHT = 400  WIDTH = 400  ACC = 0.5  FRIC = -0.12  FPS = 60  BLACK=(0,0,0)    FramePerSec = pygame.time.Clock()    displaysurface = pygame.display.set\_mode((400, 400))  pygame.display.set\_caption("pixel platformer")  font=pygame.font.SysFont("Verdana", 60, bold=True, italic=False)  font\_small=pygame.font.SysFont("Verdana",20)  game\_over=font.render("Game Over",True,BLACK)    class Player(pygame.sprite.Sprite):  def \_\_init\_\_(self):  super().\_\_init\_\_()  #self.image = pygame.image.load("character.png")  self.surf = pygame.Surface((20, 20))  self.surf.fill((100,100,100))  self.rect = self.surf.get\_rect()    self.pos = vec((200, 360))  self.vel = vec(100,0)  self.acc = vec(100,0)  self.jumping = True  self.score = 0    def move(self):  self.acc = vec(0,0.5)    pressed\_keys = pygame.key.get\_pressed()    if pressed\_keys[K\_LEFT]:  self.acc.x = -ACC  if pressed\_keys[K\_RIGHT]:  self.acc.x = ACC    self.acc.x += self.vel.x \* FRIC  self.vel += self.acc  self.pos += self.vel + 0.5 \* self.acc    if self.pos.x > WIDTH:  self.pos.x = 0  if self.pos.x < 0:  self.pos.x = WIDTH    self.rect.midbottom = self.pos    def jump(self):  hits = pygame.sprite.spritecollide(self, platforms, False)  if hits and not self.jumping:  self.jumping = True  self.vel.y = -15    def cancel\_jump(self):  if self.jumping:  if self.vel.y < -3:  self.vel.y = -3    def update(self):  hits = pygame.sprite.spritecollide(self ,platforms, False)  if self.vel.y > 0:  if hits:  if self.pos.y < hits[0].rect.bottom:  if hits[0].point == True:  hits[0].point = False  self.score += 1  self.pos.y = hits[0].rect.top +1  self.vel.y = 0  self.jumping = False      class platform(pygame.sprite.Sprite):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.surf = pygame.Surface((random.randint(50,100), 12))  self.surf.fill((0,255,0))  self.rect = self.surf.get\_rect(center = (random.randint(0,WIDTH-10),  random.randint(0, HEIGHT-30)))  self.speed = random.randint(-1, 1)    self.point = True  self.moving = True      def move(self):  if self.moving == True:  self.rect.move\_ip(self.speed,0)  if self.speed > 0 and self.rect.left > WIDTH:  self.rect.right = 0  if self.speed < 0 and self.rect.right < 0:  self.rect.left = WIDTH      def check(platform, groupies):  if pygame.sprite.spritecollideany(platform,groupies):  return True  else:  for entity in groupies:  if entity == platform:  continue  if (abs(platform.rect.top - entity.rect.bottom) < 40) and (abs(platform.rect.bottom - entity.rect.top) < 40):  return True  C = False    def plat\_gen():  while len(platforms) < 6:  width = random.randrange(50,100)  p = platform()  C = True    while C:  p = platform()  p.rect.center = (random.randrange(0, WIDTH - width),  random.randrange(-50, 0))  C = check(p, platforms)  platforms.add(p)  all\_sprites.add(p)        PT1 = platform()  P1 = Player()    PT1.surf = pygame.Surface((WIDTH, 20))  PT1.surf.fill((255,0,0))  PT1.rect = PT1.surf.get\_rect(center = (WIDTH/2, HEIGHT - 10))    all\_sprites = pygame.sprite.Group()  all\_sprites.add(PT1)  all\_sprites.add(P1)    platforms = pygame.sprite.Group()  platforms.add(PT1)  PT1.moving = False  PT1.point = False ##    for x in range(random.randint(4,5)):  C = True  pl = platform()  while C:  pl = platform()  C = check(pl, platforms)  platforms.add(pl)  all\_sprites.add(pl)      while True:  P1.update()  for event in pygame.event.get():  if event.type == QUIT:  pygame.quit()  sys.exit()  if event.type == pygame.KEYDOWN:  if event.key == pygame.K\_SPACE:  P1.jump()  if event.type == pygame.KEYUP:  if event.key == pygame.K\_SPACE:  P1.cancel\_jump()  if P1.rect.top > HEIGHT:  for entity in all\_sprites:  entity.kill()  time.sleep(1)  displaysurface.fill((255,0,0))  displaysurface.blit(game\_over,(10,10))  pygame.display.update()    time.sleep(1)  pygame.quit()  sys.exit()    if P1.rect.top <= HEIGHT / 3:  P1.pos.y += abs(P1.vel.y)  for plat in platforms:  plat.rect.y += abs(P1.vel.y)  if plat.rect.top >= HEIGHT:  plat.kill()    plat\_gen()  displaysurface.fill((0,0,0))  f = pygame.font.SysFont("Poppins", 20)  g = f.render(str(P1.score), True, (123,255,0))  displaysurface.blit(g, (WIDTH/2, 10))    for entity in all\_sprites:  displaysurface.blit(entity.surf, entity.rect)  entity.move()    pygame.display.update()  FramePerSec.tick(FPS) |





**THE GAME WEBSITE:-**

**Pixel platformer website.**

The website that we have developed is with correspondence to the game and its knowledge. It is developed using HTML and Python

**HTML:**

The hypertext markup language is the standard markup language for documents designed too be displayed in a web browser. It can be assisted by technologies such as cascading Style Sheets.

**CSS:**

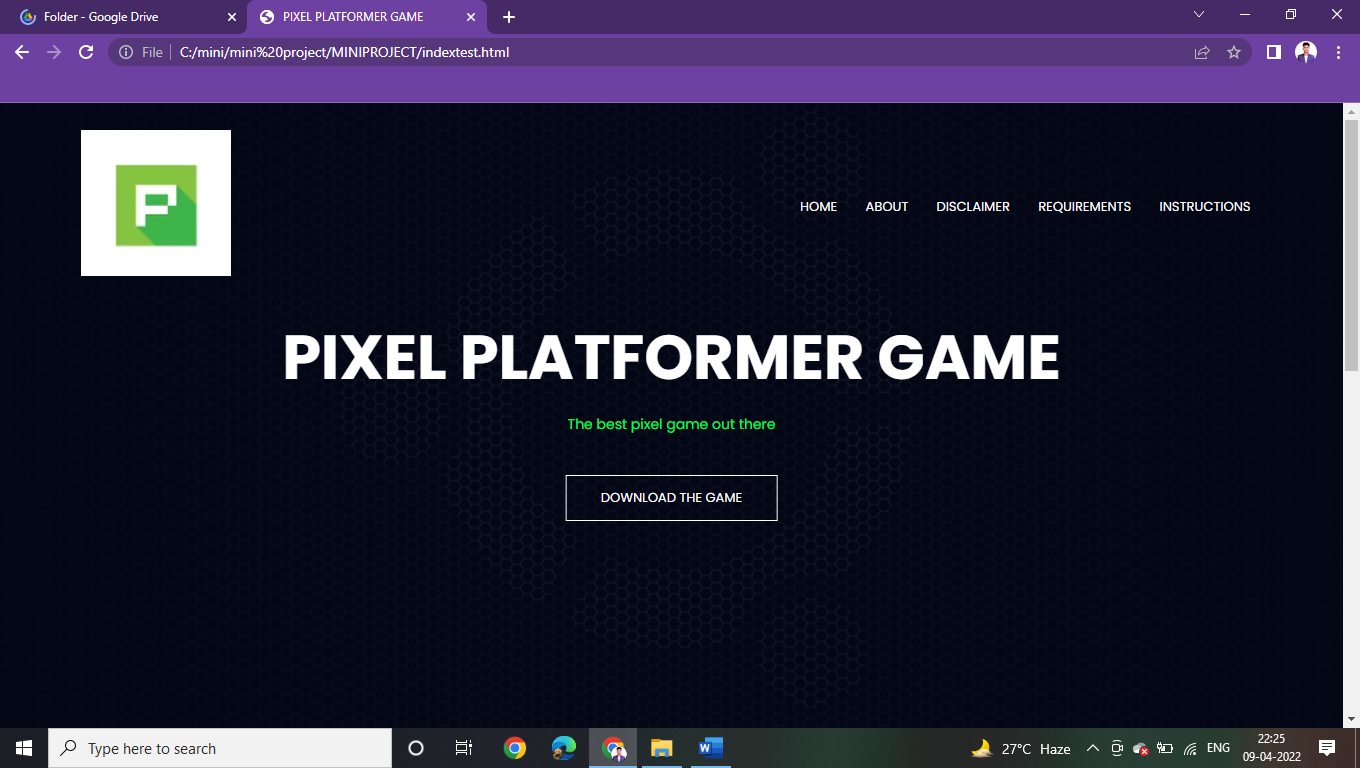
Cascading Style sheets is a style sheet language used for describing the presentation of a document written in a markup such as HTML. CSS is a cornerstone technology of the world wide web, alongside HTML and JavaScript.

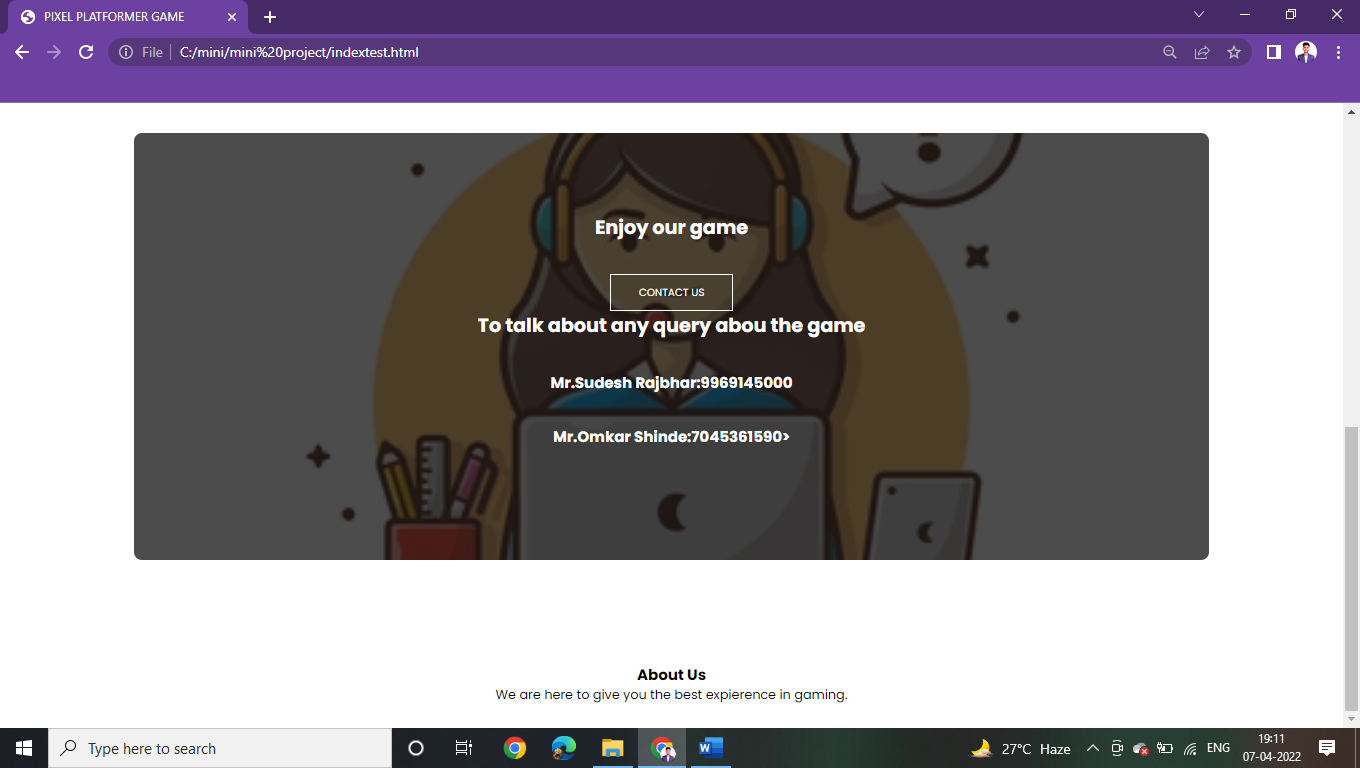
The following website contains ::

1. Home page
2. Abouts of the game.
3. Disclaimer of the game
4. Requirements of the game
5. Instructions required to play the game

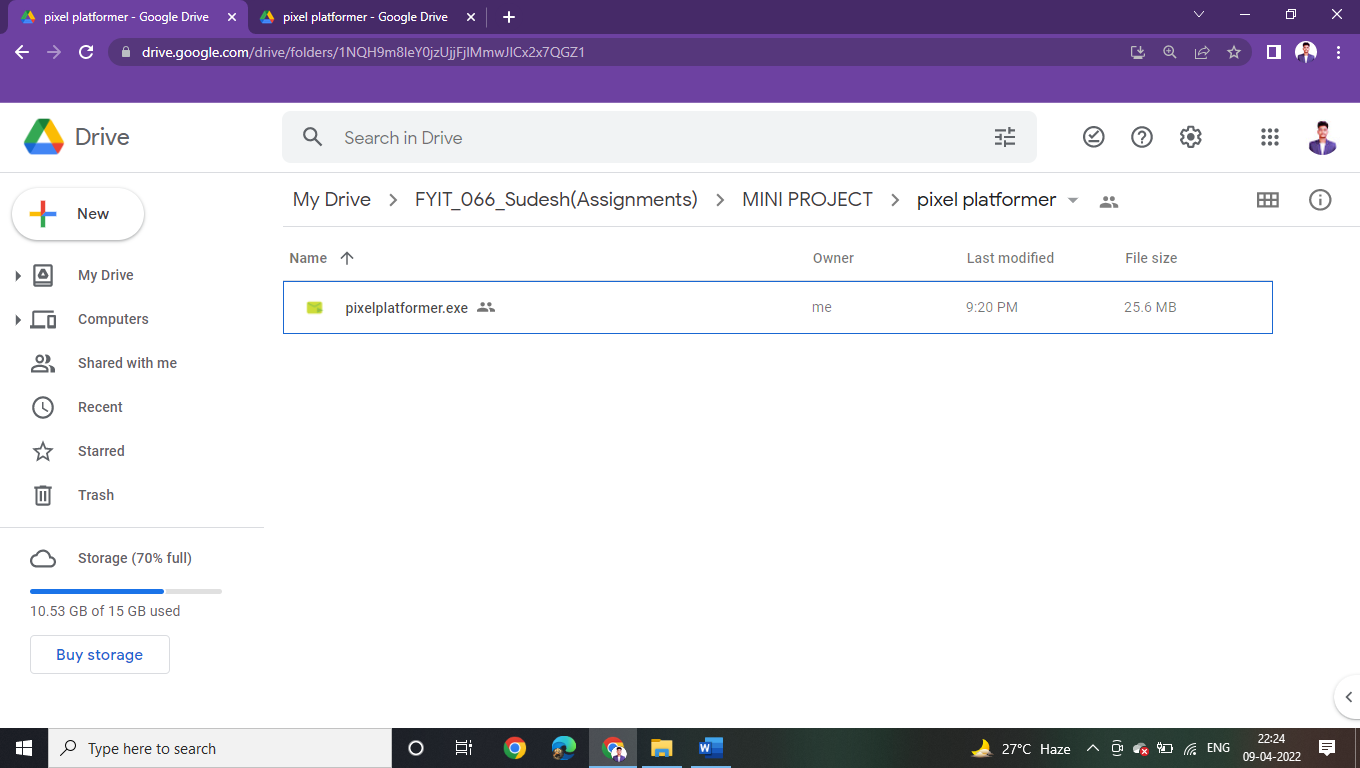
Each one of the sheets is linked with other four, i.e., you can access any other site from one particular site.

**THE HOME PAGE:**

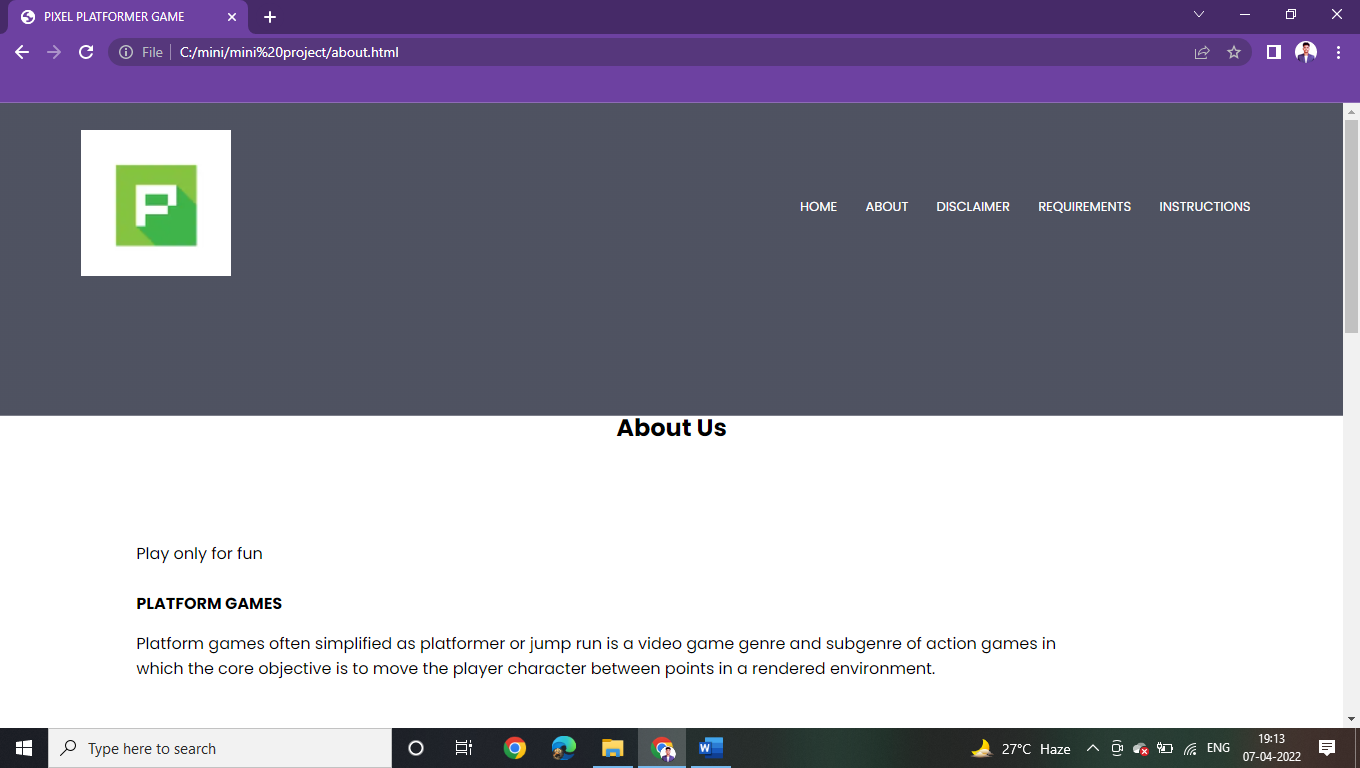


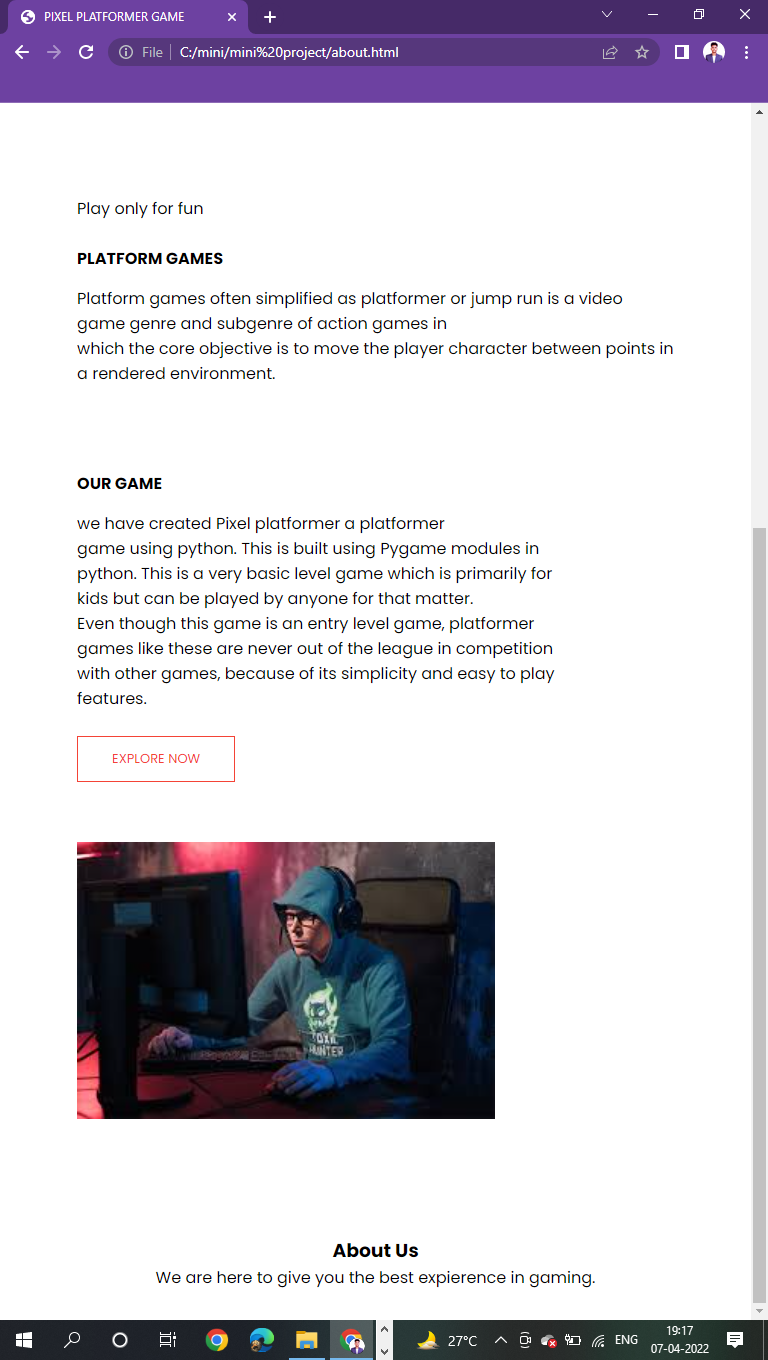


DOWNLOAD THE GAME Redirects you to the Drive Link of the file, from where you can download your file :-

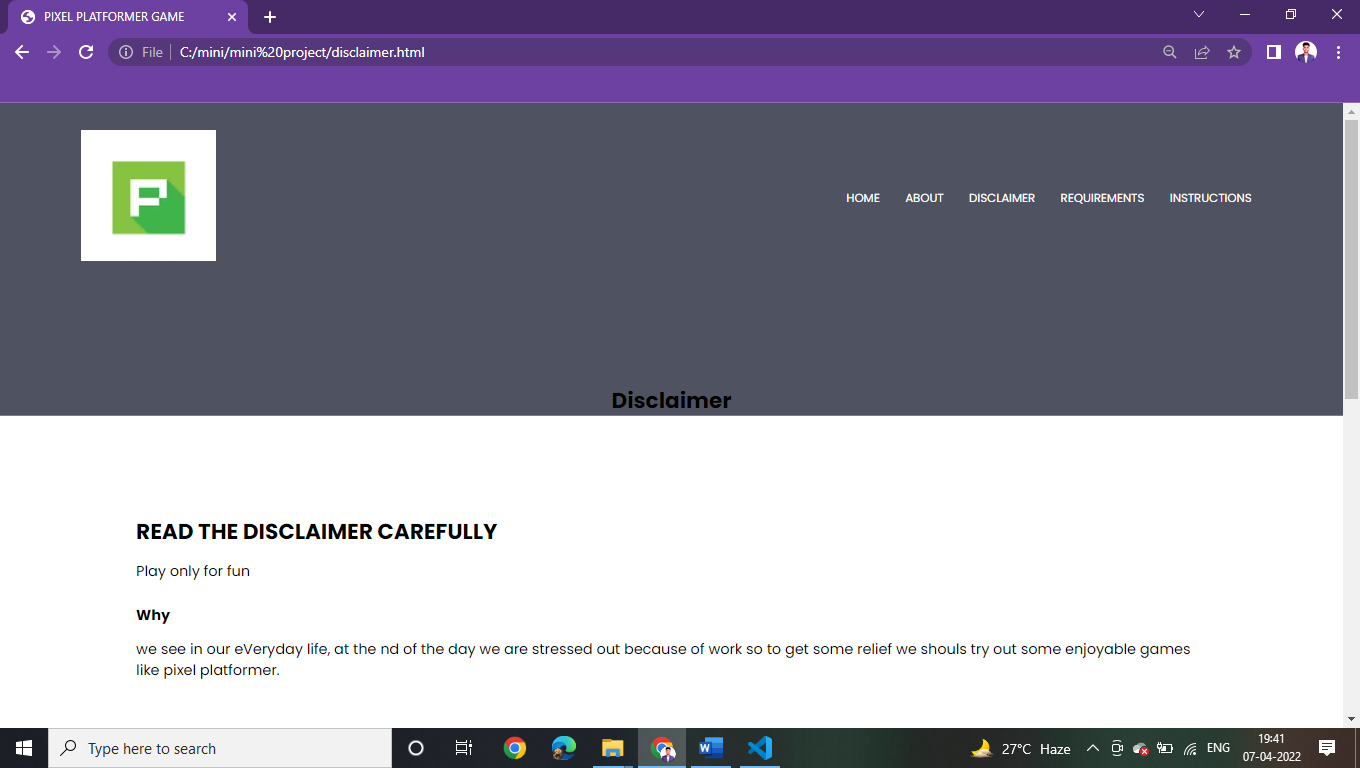


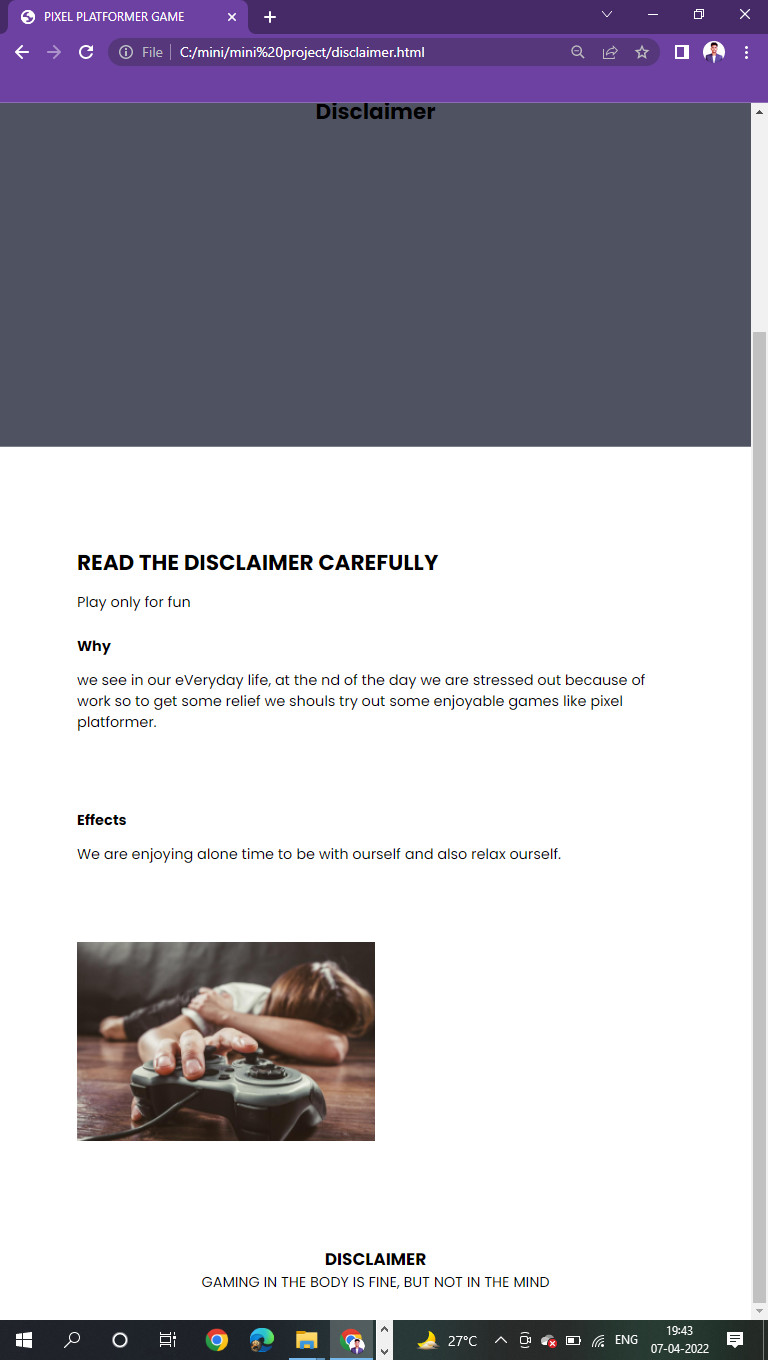
THE ABOUT PAGE:-



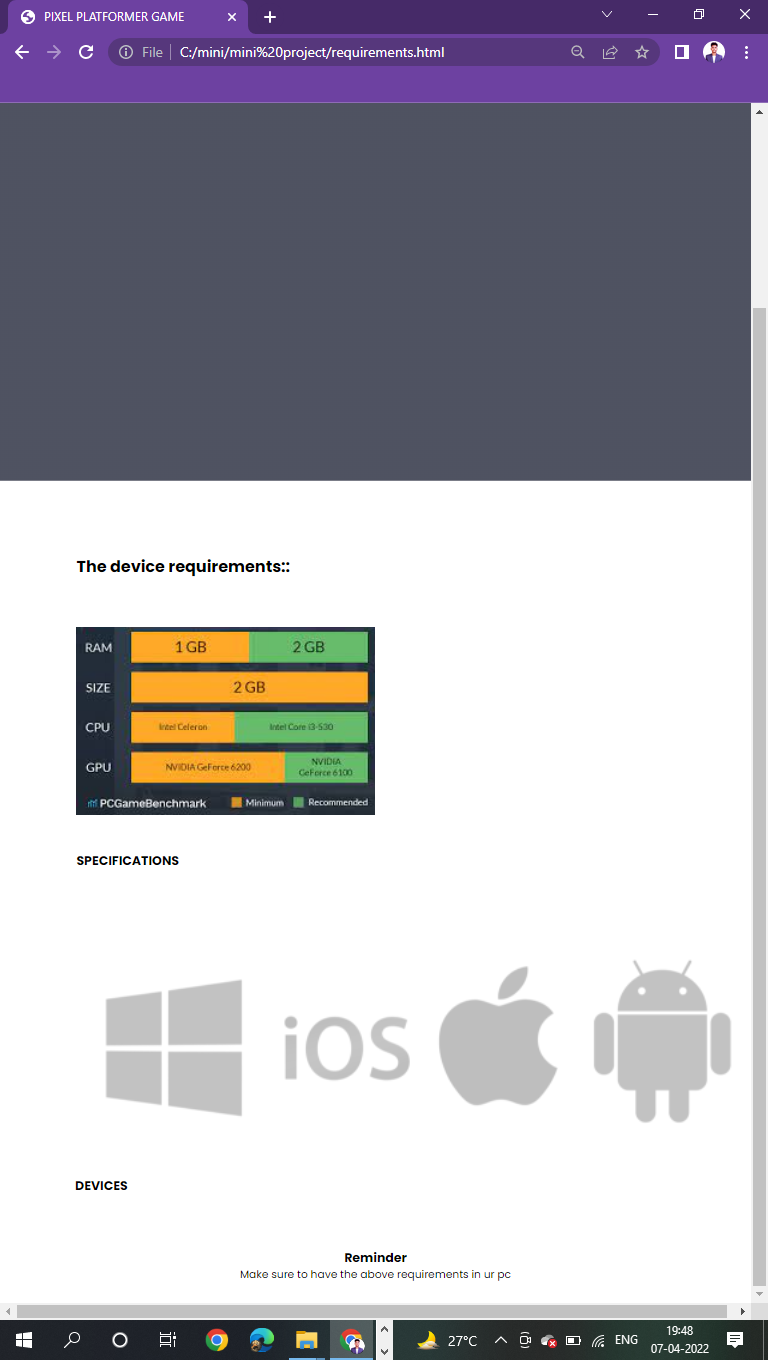


THE DISCLAIMER PAGE:-

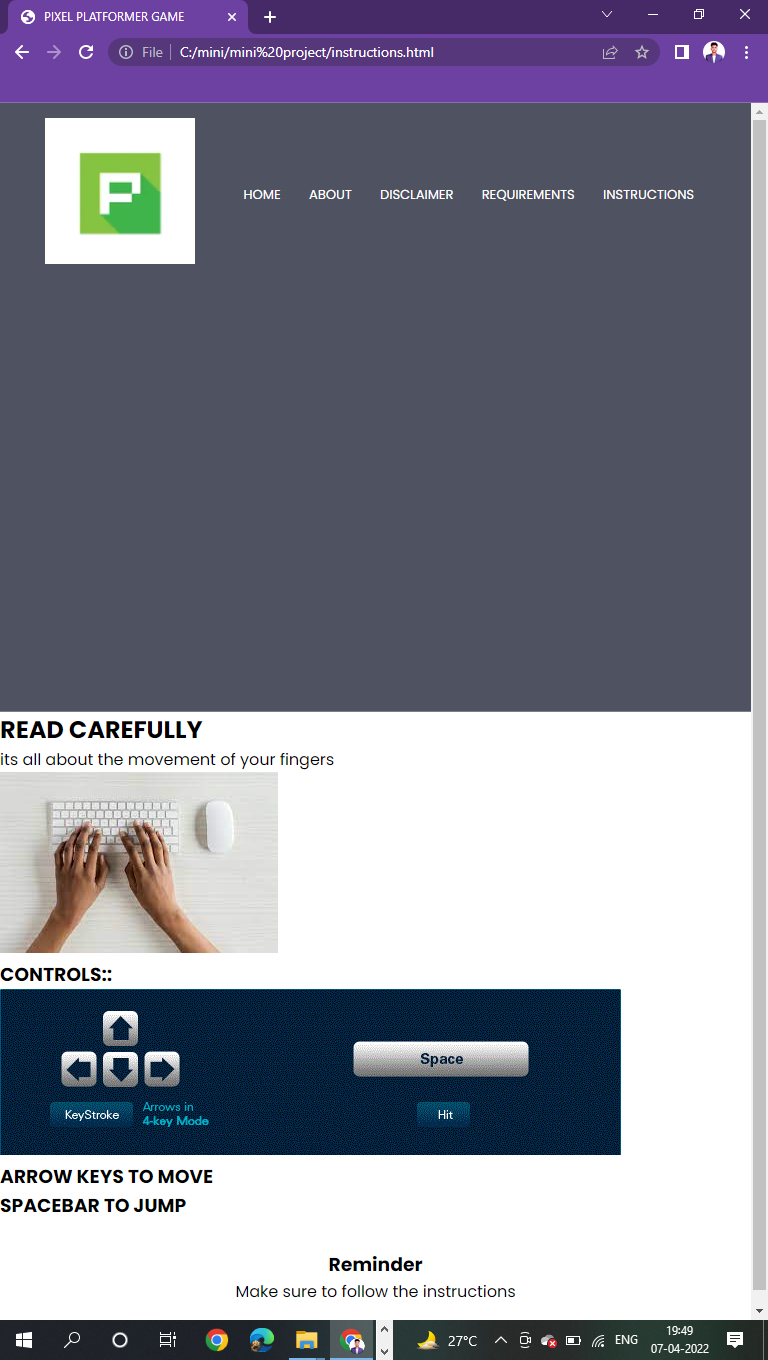




THE REQUIREMENTS PAGE:-



THE INSTRUCTIONS PAGE:-



PURPOSE OF THE WEBSITE: TO Provide the information about the game and also to link the game with the website.

**Conclusion:-**

In conclusion, we have created “Pixel platformer” a platformer game using python. This is built using Pygame modules in python. This is a very basic level game which is primarily for kids but can be played by anyone for that matter.

Even though this game is an entry level game, platformer games like these are never out of the league in competition with other games, because of its simplicity and easy to play features.

**REFERENCE LINKS:-**

<https://coderslegacy.com/python/python-pygame-tutorial/>

<https://youtu.be/Ongc4EVqRjo>