Final project
Destiny 2 themed 2D platformer

Abzal Orazbek

Design

Graphics

Sounds

Programming

```python
level1_data = {
    "background": "assets/images/background/europa_1.png",
    "music": "assets/sounds/music/beyond_light.mp3",
    "layout": [
        "WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW",
        "W                           W",
        "W                           W",
        "W                           W",
        "W       WWWWWWW      WWWWWWWW W",
        "W T               EE  T      W",
        "W WWW    WWWWWWWWWWWWWWW      W",
        "W        WWW                 W",
        "W           E               W",
        "W  WWWW                  T W W",
        "W  WWWWWWWWWWWWWW            W",
        "W             E             W",
        "W                     T     W",
        "W                           W",
        "WT                          W",
        "WWWWWW                      W",
        "W                           W",
        "W                           W",
        "W      E                    W",
        "W  T                        W",
        "WWWWWWWWWW           E     W W",
        "W                           W",
        "W                           W",
        "W        WWWW     WWWW       W",
        "W                           W",
        "W                           W",
        "W   T   T        T      T  W",
        "WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW",
    ],
    "tile_size": 30,
    "player_spawn": (1000, 700),
    "engram_count": 6,
```

FINAL

- assets
  - fonts
  - images
  - sounds
- docs
  - ~$AbzalOrazbek_Final_Pres...   U
  - ~$zalOrazbek_Final_Report....   U
  - AbzalOrazbek_Final_Presentation...
  - AbzalOrazbek_Final_Report.docx
- src
  - __pycache__
  - levels
  - game.py
  - level.py
  - main.py
  - player.py
  - ui.py
  - utils.py
- .gitattributes
- .gitignore
- README.md
- requirements.txt
- startGame.cmd
- startGame.sh
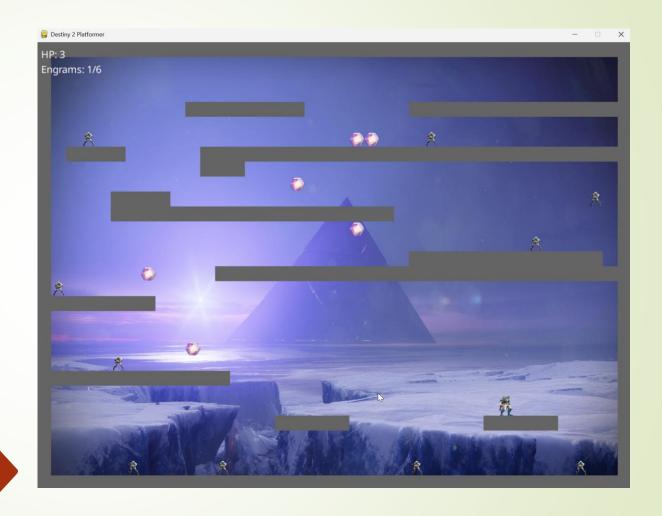
```python
for row_index, row in enumerate(layout):
    for col_index, cell in enumerate(row):
        x = col_index * tile_size
        y = row_index * tile_size
        if cell == 'W': # Wall
            tile = pygame.sprite.Sprite()
            tile.image = pygame.Surface((tile_size, tile_size))
            tile.image.fill((100, 100, 100)) # Gray color for walls
            tile.rect = tile.image.get_rect(topleft=(x, y))
            self.tile_group.add(tile)
        elif cell == 'E': # Engram
            engram = pygame.sprite.Sprite()
            engram.image = pygame.transform.scale(engram_image, (tile_size, tile_size)) # Scale engram image
            engram.rect = engram.image.get_rect(topleft=(x, y))
            self.engram_group.add(engram)
        elif cell == 'T': # Trap
            trap = pygame.sprite.Sprite()
            trap.image = pygame.transform.scale(trap_image, (tile_size, tile_size)) # Scale trap image
            trap.rect = trap.image.get_rect(topleft=(x, y))
            self.trap_group.add(trap)


class Player(pygame.sprite.Sprite):
    def __init__(self, spawn_position, image_path):
        super().__init__()
        self.image = pygame.image.load(image_path).convert_alpha()
        self.image = pygame.transform.scale(self.image, (30, 40))
        self.rect = self.image.get_rect(topleft=spawn_position)
        self.velocity = pygame.math.Vector2(0, 0)
        self.speed = 5
        self.gravity = 1
        self.jump_force = -30
        self.on_ground = False
        self.hp = 3 # Initial health points
```

Result

Future plans

- Better gravity
- More levels
- Random levels
- Codebase improvements

Thank you