ENGINEERING PROJECT I

(Computer Engineering)

/2D Platform Game\

Leyla Abdullayeva

Information About The Game

- That game is a 2D platform game which playing with PC keyboards
- That game is running on Windows 10

What is the aim in our game:

- You need to collect the coins in game and get best score

Important Things:

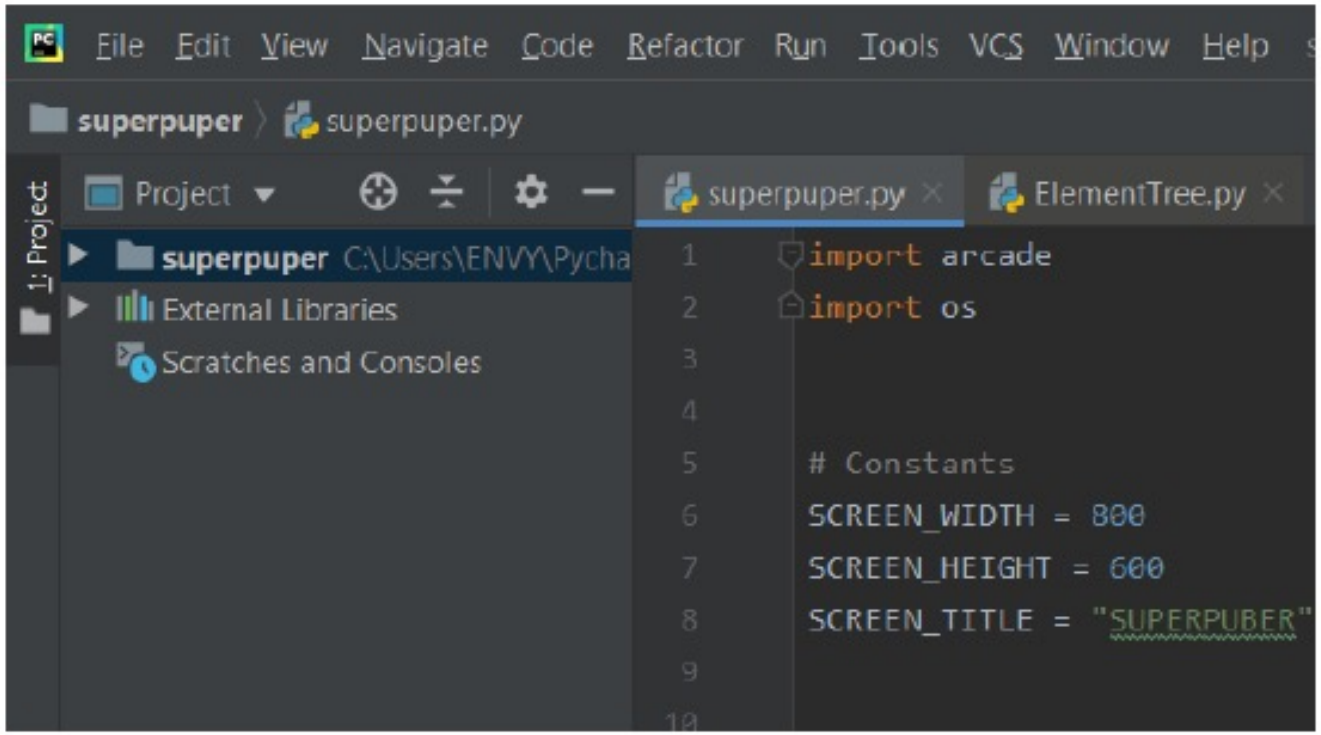
- In game, you have 3 health.
- There are some pits which you should not fall. If you fall, you will be die and lose your 1 health.

What development tools and software languages did they use for the project?

- We used **Python** 3.8 programming language.
- **Python** is an outstanding language for people learning to program.
- And also we used **PyCharm** Compiler.
- **PyCharm** is an integrated development environment (IDE) used in computer programming, specifically for the Python language.

What we used in our project ?

- Arcade .
- OS (operating system).



What The difficulties we faced ?

- How to create a map for the level .
- How to create an animations for the player .
- How to create more than one level .
- How to check if the player touch something like (lava) .

```python
        platforms_layer_name = 'Platforms'
        coins_layer_name = 'Coins'
        moving_platforms_layer_name = 'Moving Platforms'
        dont_touch_layer_name = "Don't Touch"

        # Map name
        map_name = f":resources:tmx_maps/ws500_{level}.tmx"

        # Read in the tiled map
        my_map = arcade.tilemap.read_tmx(map_name)

        # Calculate the right edge of the my_map in pixels
        self.end_of_map = my_map.map_size.width * GRID_PIXEL_SIZE

        # PLATFORMS
        self.wall_list = arcade.tilemap.process_layer(my_map, platforms_layer_name, TILE_SCALING)

        # Coins
        self.coin_list = arcade.tilemap.process_layer(my_map, coins_layer_name, TILE_SCALING)

        # Moving platforms(if we have)
        moving_platforms_list = arcade.tilemap.process_layer(my_map, moving_platforms_layer_name, TILE_SCALING)
        for sprite in moving_platforms_list:
            self.wall_list.append(sprite)

        # Background Objects
        self.background_list = arcade.tilemap.process_layer(my_map, "Background", TILE_SCALING)
        self.ladder_list = arcade.tilemap.process_layer(my_map, "Ladders", TILE_SCALING)

        # Don't Touch Layer
        self.dont_touch_list = arcade.tilemap.process_layer(my_map, dont_touch_layer_name, TILE_SCALING)
```

1. Opening a game window.

In order to open a game window, you must've downloaded arcade library beforehand.

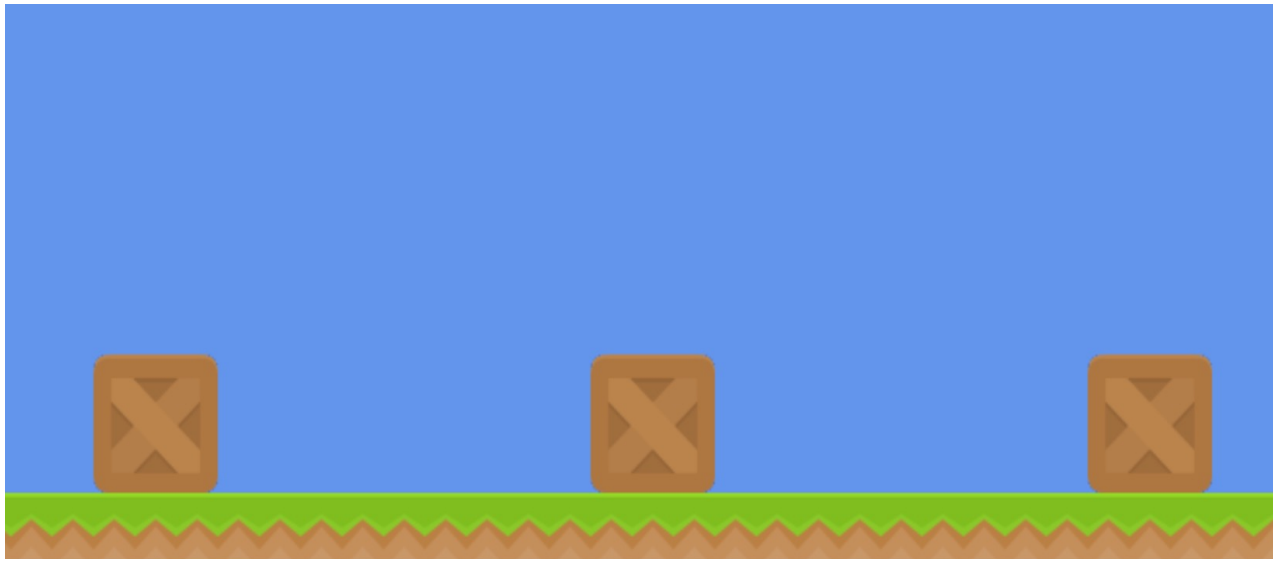
```python
import arcade
import os

# Constants
SCREEN_WIDTH = 800
SCREEN_HEIGHT = 600
SCREEN_TITLE = "Patformer"

# Constants used to scale our sprites from their original size
CHARACTER_SCALING = 0.8
TILE_SCALING = 0.5
COIN_SCALING = 0.5
SPRITE_PIXEL_SIZE = 128
GRID_PIXEL_SIZE = (SPRITE_PIXEL_SIZE * TILE_SCALING)
```

2.Sprite addition

After opening a new game window, the next step is to put in some graphics. "Sprites" are the graphical items that you can interact with. We create a sprite by using "Sprite" class.



3. Character definition.

```python
# Textures
main_path = ":resources:images/animated_characters/male_adventurer/maleAdventurer"

# Textures for idle standing
self.idle_texture_pair = load_texture_pair(f"{main_path}_idle.png")
self.jump_texture_pair = load_texture_pair(f"{main_path}_jump.png")
self.fall_texture_pair = load_texture_pair(f"{main_path}_fall.png")
# Textures for walking
self.walk_textures = []
for i in range(8):
    texture = load_texture_pair(f"{main_path}_walk{i}.png")
    self.walk_textures.append(texture)

# Initial texture (go to 0)
self.texture = self.idle_texture_pair[0]
self.set_hit_box(self.texture.hit_box_points)
```

By writing this code, we're defining our player's character.

4.Adding the user control.

```python
# Player speed
PLAYER_MOVEMENT_SPEED = 5
GRAVITY = 1
PLAYER_JUMP_SPEED = 20
```

```python
def on_update(self, delta_time):
    # We're calling physics engine

    self.physics_engine.update()

    # Update animations
    if self.physics_engine.can_jump():
        self.player_sprite.can_jump = False
    else:
        self.player_sprite.can_jump = True

    if self.physics_engine.is_on_ladder() and not self.physics_engine.can_jump():
        self.player_sprite.is_on_ladder = True
        self.process_keychange()
    else:
        self.player_sprite.is_on_ladder = False
        self.process_keychange()
```

```python
def on_key_press(self, key, modifiers):                    # Keyboard functions
    if key == arcade.key.UP or key == arcade.key.W:
        self.up_pressed = True
    elif key == arcade.key.DOWN or key == arcade.key.S:
        self.down_pressed = True
    elif key == arcade.key.LEFT or key == arcade.key.A:
        self.left_pressed = True
    elif key == arcade.key.RIGHT or key == arcade.key.D:
        self.right_pressed = True

    self.process_keychange()

def on_key_release(self, key, modifiers):

    if key == arcade.key.UP or key == arcade.key.W:
        self.up_pressed = False
        self.jump_needs_reset = False
    elif key == arcade.key.DOWN or key == arcade.key.S:
        self.down_pressed = False
    elif key == arcade.key.LEFT or key == arcade.key.A:
        self.left_pressed = False
    elif key == arcade.key.RIGHT or key == arcade.key.D:
        self.right_pressed = False
```

5.Adding coins and sound

```
# Coins
self.coin_list = arcade.tilemap.process_layer(my_map, coins_layer_name, TILE_SCALING)
```

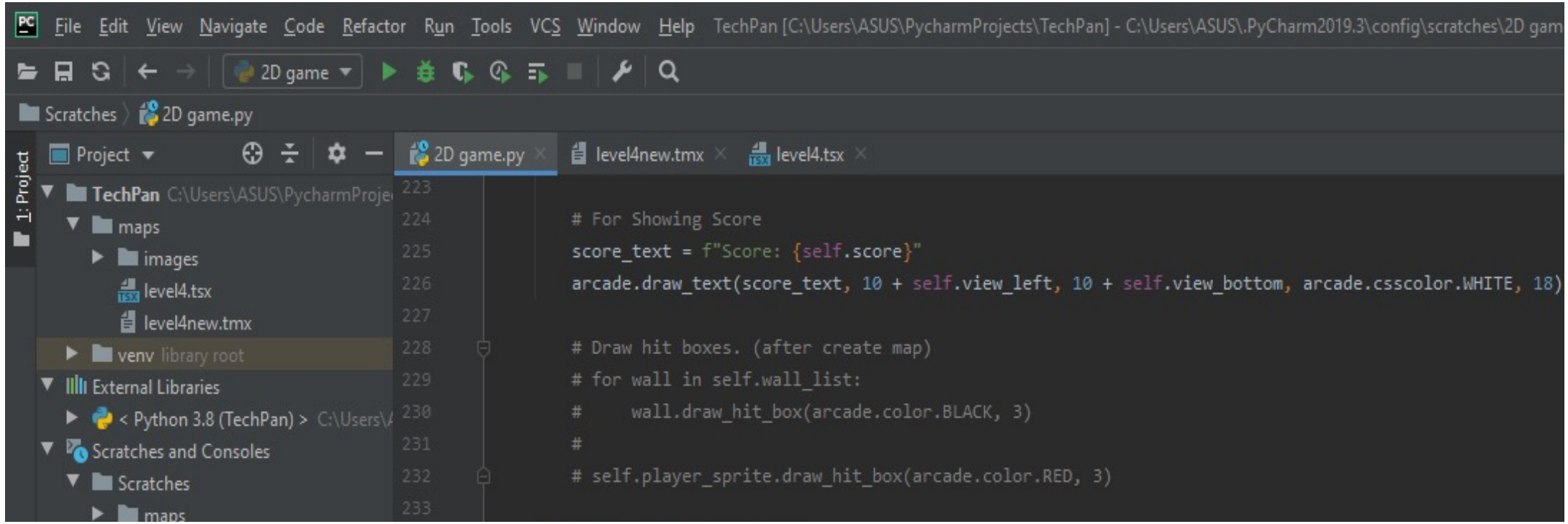
```
# if you hit any coins
coin_hit_list = arcade.check_for_collision_with_list(self.player_sprite, self.coin_list)

for coin in coin_hit_list:
    if 'Points' not in coin.properties:   # özellikler
        print("Warning, collected a coin without a Points property.")
    else:
        points = int(coin.properties['Points'])
        self.score += points
    # Remove the coin
    coin.remove_from_sprite_lists()
    # Play a sound
    arcade.play_sound(self.collect_coin_sound)
```

The codes in here adds coins that we can collect. When player hits any coins, or presses jump button it also adds sound.

6.Displaying the score.

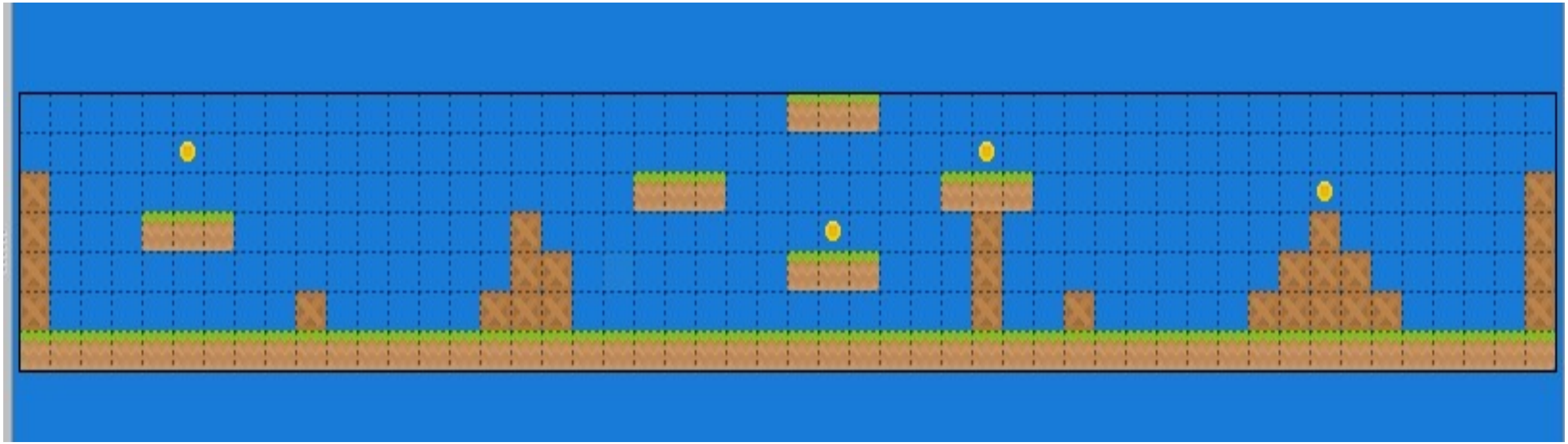
This is the the “Score addition” part of our code. Now that we can collect coins and get points.



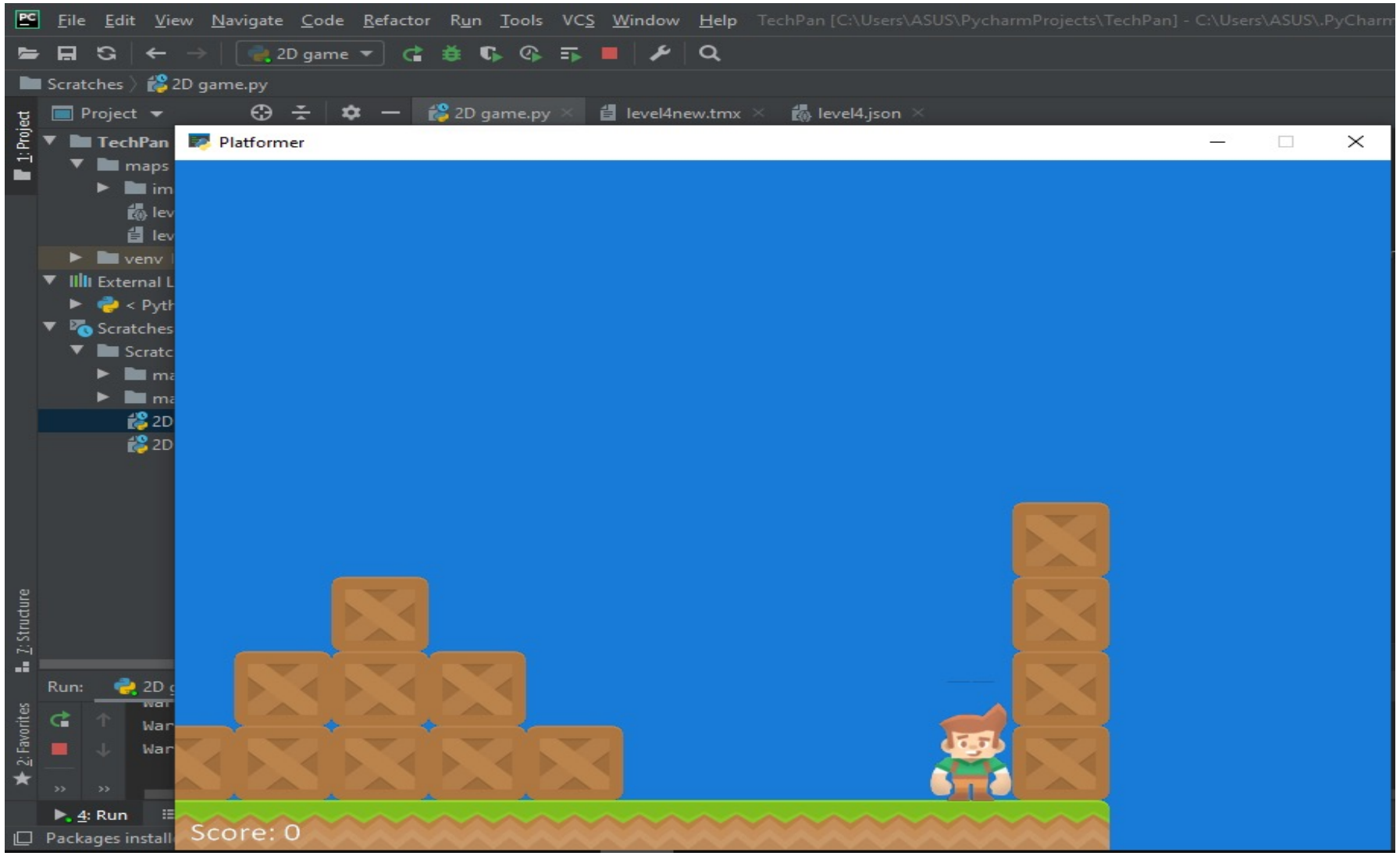
7.Map addition

In here, we're adding the map that we've created into our code. We're coding tmx&tsx files here.

And this is the map that we used for our 1 level based Prototype-1. We've created it by using "Tiled" map editor.



As you can see, this is the first model of our “2D platform game”.



Resources, training video addresses.

- Mostly we used “arcade.academy”. It was pretty useful. And it was our teacher’s recommendation.
- From Youtube we watched a plenty of videos from channel called “Tech with Tim”, ”DaFluffyPotato”, “Professor Craven” and etc.