# Tile based Mini RPG Project

A Level Computer Science Project Taster

Contents

[Tile based Mini RPG Project 1](#_Toc54268605)

[Introduction 3](#_Toc54268606)

[Task 1 -Initial setup 4](#_Toc54268607)

[Questions 4](#_Toc54268608)

[Task 2 – Walls 6](#_Toc54268609)

[Extension 6](#_Toc54268610)

[Task 3 – Adding text and user information 7](#_Toc54268611)

[Extension 7](#_Toc54268612)

[Task 4 - Adding opponents and randomness 8](#_Toc54268613)

[Extension 8](#_Toc54268614)

[Task 5 – Collision and Combat using groups and onscreen text 9](#_Toc54268615)

[Extension 9](#_Toc54268616)

[Task 6 – New levels and beyond 10](#_Toc54268617)

[Final Extension 11](#_Toc54268618)

[Your Presentation 16](#_Toc54268619)

## Introduction

In this project we will build a mini tile-based Role Playing Game using object-oriented techniques with PyGame. This is not a step-by-step guide to follow but it will give you enough to get started and give you some help. Many of the concepts have been covered in theory already. This will you give a further opportunity to implement them in practice. You will be shown various approaches to complete the tasks in class.

You may choose to work collaboratively at some points, but all the work should be your own. You could usefully seek user feedback from your peers or target audience for example about what works well and what areas could be improved. Many of the practices you follow here will be replicated in a larger scale in your major project so it is in your interests to complete the work independently and to the highest standard.You should consider the tasks to be the minimum required and feel free to extend in any direction of your own interest beyond that required below.

You will be marked both on this document, the end product and the presentation to the class at the conclusion. You should add all your answers and any relevant code and screenshots to this document which will be assigned, marked and returned in the usual manner in Teams.

## Task 1 -Initial setup

Create a white screen 1000x1000 with the usual colours and check it runs. You can use the basic template from

<http://programarcadegames.com/python_examples/f.php?file=pygame_base_template.py>

as your starting point. Create a Player Class.

https://del.dog/ihatemylifetxt.py

### Questions

What is a class?

What attributes does your class have?

What methods do you anticipate it having?

What is inheritance? Where and how is it used in your Player class?

Now instantiate a player object.

What is an object?

Add your player to a group. What does a group do in PyGame?

Write code to enable your player class to be controlled by the keyboard and paste below with an explanation of what is happening. The code should allow 20 pixels movement with each key press moving up, down, left or right.

Your code

How would you alter this code to allow the keys to be held down and carry on moving? Add the code below with an explanation.

keys = pygame.key.get\_pressed()

if keys[pygame.K\_LEFT]:

PC.rect.x -= tilesize // 4

elif keys[pygame.K\_RIGHT]:

PC.rect.x += tilesize // 4

if keys[pygame.K\_UP]:

PC.rect.y -= tilesize // 4

elif keys[pygame.K\_DOWN]:

PC.rect.y += tilesize // 4

## Task 2 – Walls

Write code to add red walls along the perimeter of your screen. The walls should be objects of a Wall class. Each object should be a 40x40 square. You will find the fastest way is to iterate through a list creating a wall when the list indicates.

Paste your code below and explain in your own words what is happening. You should mention the Wall class itself, any groups the walls may be in and programming constructs you have used to create the structure.

Your code

class outerwall(pygame.sprite.Sprite):

# Define the constructor for the wall

def \_\_init\_\_(self, color, width, height, x\_ref, y\_ref):

# Call the sprite constructor

super().\_\_init\_\_()

# Create a sprite and fill it with colour

self.image = pygame.Surface([width\*tilesize,height\*tilesize])

self.image.fill(color)

self.rect = self.image.get\_rect()

# Set the position of the wall's attributes

self.rect.x = x\_ref

self.rect.y = y\_ref

class innerwall(pygame.sprite.Sprite):

# Define the constructor for the wall

def \_\_init\_\_(self, color, width, height, x\_ref, y\_ref):

# Call the sprite constructor

super().\_\_init\_\_()

# Create a sprite and fill it with colour

self.image = pygame.Surface([width\*tilesize,height\*tilesize])

self.image.fill(color)

self.rect = self.image.get\_rect()

# Set the position of the wall's attributes

self.rect.x = x\_ref

self.rect.y = y\_ref

## --== instantiates the map

for y in range(maxsizey):

for x in range (maxsizex):

if map[y][x] == "%":

new\_wall = outerwall((200,30,30), 1, 1, x\*tilesize, y \*tilesize)

wall\_list.add(new\_wall)

all\_sprites\_list.add(new\_wall)

if map[y][x] == "#":

new\_wall = innerwall(RED, 1, 1, x\*tilesize, y \*tilesize)

wall\_list.add(new\_wall)

all\_sprites\_list.add(new\_wall)

Alter your code so the player spawns in the middle of the map and is now unable to pass through walls. Paste your code below with an explanation of how you have achieved this.

Your code

PC = player(BLUE,tilesize ,tilesize ,tilesize\*(maxsizex//2),tilesize\*(maxsizey//2))

all\_sprites\_list.add(PC)

**player\_collide\_list = pygame.sprite.spritecollide(PC, wall\_list, False)**

**for foo in player\_collide\_list:**

**PC.rect.x = PC\_oldX**

**PC.rect.y = PC\_oldY**

## Extension

Add more walls to the centre of the map to create interesting patterns for the player to explore. Create a new wall class that is only for use in inner walls and add this to the list.

## Task 3 – Adding text and user information

We want to provide some information on the state of play to the end user. Add the attributes health, money, keys and score to the player class. Set health to 100 and money and score to 0. Lengthen the screen to 1200. Display the attributes in this new section at the bottom of the screen. Add your code below along with an explanation in your own words of how you have done it.

Your code

UIsectionsize = 5

size = (maxsizex \* tilesize,maxsizey\*tilesize + UIsectionsize \* tilesize)

text = font.render(" | HP: " + str(PC.health) \

+ " | Money: " + str(PC.money) + " | Score: " \

+ str(PC.score) + " | ", True, BLACK)

textRect = text.get\_rect()

textRect.center = ((maxsizex \* tilesize) // 2,\

maxsizey\*tilesize + (UIsectionsize \* tilesize) // 2)

screen.blit(text,textRect)

### Extension

It may be useful to know the position of each wall. Write code to display the position of each block within the list on the screen. E.g the top left wall should show 0,0. The bottom right wall should show 24,24. Add your code below with an explanation.

## Task 4 - Adding opponents and randomness

Create a new class called enemy. An enemy should be a 40x40 square again. Place three instances of the enemy in the corners of the map.

Add your code for the enemy class below and explain the attributes you have used and why.

Your code

### Extension

* Randomize the location of the enemy with the boundary of the map.
* Randomize the colour of the enemy

Your code

## Task 5 – Collision and Combat using groups and onscreen text

Using spritecollide or groupcollide, write code to record when the player meets the enemy. Write a combat system with rounds, printing out the results to the shell. You can assume the player starts with more health than the enemy or the player will die without having a fair chance of defeating all three enemies. The defeated enemy should disappear from the map. You will need a random function for each round of damage here otherwise it will be the same each time.

Write down your code below and explain what is happening

When defeated, the enemy will award the player a key. Update the score and the key count at the bottom of the screen. Choose your own scoring system. If the player dies, then end the game.

Your code

### Extension

Your own choice here – what would you like to add to Task 5?

## Task 6 – New levels and beyond

Once a player has gained three keys, a portal to the next level appears. Write code that creates a new block in the bottom right hand corner. When a player touches this block, the current level should disappear and a new map should be drawn with new enemies. The player score, wealth and health should stay the same. Keys should reset to 0.

Add your code below and explain what is happening.

### Final Extension

This game can be extended in numerous ways to make it more interesting and fun. Here are some ways we could improve the game.

* Moving enemies
* Instead of squares, using pictures or even animation
* Powerups for the player
* Enemies that shoot bullets
* Various weapons for the player
* A timer
* Difficult settings
* An opening and closing scene
* A big boss enemy for each stage
* Walls that close in on the player

What else can you think of to improve your game?

Think of at least 3 major improvements to implement. They may be from the above list or ones you have thought of yourself. Write down a flowchart with a paragraph of text explaining how you have done it below.

Improvement 1 Text

Improvement 1 Flowchart

Improvement 2 Text

Improvement 2 Flowchart

Improvement 3 Text

Improvement 3 Flowchart

Implement your improvements and add the code below with a screenshot of them in action.

## Your Presentation

You will present your fully working code to the class. Your presentation should cover the way you have implemented the functions in the 6 tasks above as well as the improvements in particular. You should be able to run the code as a demo and explain how it works. You will receive a mark based on the following:

* Style of presentation – is it interesting, clear, accessible for the audience, comprehensive?
* Effort of implementation of the code and the extensions – are all tasks completed? Are at least 3 extensions done? Is there a real effort to engage with the process?
* Final product e.g. comments in code, original ideas, testing, user interface, user feedback, colour schemes, onscreen help, instructions, user settings?

Add a printout out of your PowerPoint presentation below.