Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73

P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

Penerapan library pygame dalam game RPG“the adventure”

[1]Yogie Wilvren Saragih, [2] Samuel Krispama Lumbantoruan, [3]Edo Armando, [4]Jefri Junita

Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional “Veteran” Jawa Timur

Email : 20081010050@student.upnjatim.ac.id,
20081010066@student.upnjatim.ac.id, 2002020009@mhs.univbinainsan.ac.id,
2002020069@mhs.univbinainsan.ac.id

Received: 31 Desember 2022, **Revised:** 10 Januari 2023, **Accepted:** 12 Januari 2023

Abstrak

Perkembangan game semakin pesat diikuti juga oleh tools untuk membuat game. Terdapat banyak genre game, Role Playing Game (RPG) merupakan salah satunya. Game RPG merupakan sebuah game dimana pemain akan berperan sebagai tokoh utama dan mengikuti alur cerita permainan. Game ini terinspirasi dari game-game rpg biasa. Terdapat beberapa masalah pada game, seperti jalan cerita yang belum terealisasi dan beberapa bug pada area tertentu tidak bisa dilewati.Game RPG ini juga selalu disematkan item-item pendukung seperti senjata, item, atau hal-hal yang menunjang peningkatan level permainan.

Kata kunci: *Role playing Game, Genre Game, Item*

Abstract

Game development is getting faster followed by tools for making games. There are many game genres, Role Playing Game (RPG) is one of them. Game RPG is a game where players will act as the main character and follow the storyline of the game. This game is inspired from the usual rpg games. There is some issues with the game, such as unrealized storyline and some bugs in certain areas it cannot be passed. This RPG game is also always pinned with items support such as weapons, items, or things that support increasing the level of the game.

Keywords: *Role playing Game, Genre Game, Item*

1 Pendahuluan

Pygame adalah sebuah library Python yang dirancang untuk membuat game. Dengan Pygame, Dapat dengan mudah membuat game 2D seperti game platformer, game arcade, game puzzle, dan game role-playing game (RPG). Pygame menyediakan berbagai fitur seperti kemampuan untuk menampilkan grafik, mengelola event, memainkan audio dan video, dan banyak lagi.Untuk membuat game RPG dengan Pygame, Perlu menginstal library Pygame. Ini dapat dilakukan dengan menjalankan perintah "pip install pygame" di terminal atau command prompt. Kemudian, Mulai membuat game RPG dengan Pygame dengan cara mengimport library Pygame dan menggunakan berbagai fitur yang disediakannya.

2 Metode Penelitian

- **Analisis Sistem**

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

Pada tahapan ini analisis sistem yang dilakukan dengan melakukan evaluasi terhadap sistem aplikasi game yang akan dibangun seperti, deskripsi aplikasi, analisis kebutuhan user dan analisis kebutuhan fungsional.

- **Desain Interface**

Tahapan ini merancang tampilan interface/antarmuka yang menjadi perantara antara pengguna (user) dengan program yang berupa komponen tampilan dan tata letak.

- **Implementasi**

Tahap implementasi merupakan tahap penciptaan perangkat lunak ke dalam bahasa pemrograman yang dimengerti oleh komputer, serta merupakan tahap kelanjutan dari kegiatan perancangan sistem.

- **Pengujian**

Tahapan pengajuan dilakukan untuk menemukan kekurangan pada gim agar dapat dikembangkan kedepannya.

3 Hasil dan Pembahasan

Gim ini dibangun dengan beberapa asset data penting didalamnya dimana setiap data tersebut berkesinambungan dalam code python dan terletak pada file main,py.

Berikut beberapa data aset pada gim “The Adventure” Berbasis python ini :

1. Data

Data ini berisi “saved.db” merupakan proses save file yang ada pada game tetapi masih rampung dalam pembuatannya karena masih dalam tahap 50%.

2. Fonts

Aset fonts merupakan aset penentuan font atau jenis tulisan yang akan dimasukkan ke dalam gim.

3. GameConfig

Data ini berisikan mengenai penempatan objek,sign teks beserta penentuan level dalam game atau map yang disusun berdasarkan pemrograman json dan setiap level disusun berdasarkan txt yang ditempatkan sesuai keinginan programmer.

Teks berikut disusun dalam txt dan diatur sesuai keinginan editor dalam mengembangkan gamenya.

// All the level designs are created here. The code and puzzles are in the individual python files.

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
// # - rocks
// @ - dead vines
// ^ - next level/room door
// ! - movable object
// * - sign
// $ - tree
// & - tree grove
// [empty] - ground
```

Dan berikut merupakan penempatan setiap level atau map dalam game yang di dalamnya terdapat objek yang telah ditentukan.

```
// Level One
###############
###############
#$$$$$$$$$$$$$#
#$     $$$$$$#
# @   #######
#
#$$
#$$       *
#$$$$
####$$$ #######
#$$&    *&    #
#           #
###############
###############

// Level Two
#####     #####
#####     #####
#####     #####
#           #
#       $$ #
        &$   #
        *$   #
        &$   #
        $$   #
#           #
#   $&*&$     #
#   $$$$$     #
###############
###############

// Level Three
###############
###############
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
################
#       $$$$#
 $$$$$$$$ $$$$#
#$$$     $ $$#
#$$$ $$$$$$ $$#
#$$$ $$$ $ $$#
#$$$ $$$$ $$#
#$$$ $$$$$$ $$#
#  $   &*&   #
#  $$    $$ #
#####    #####
#####    #####

// Level Four
####      ####
#####     ####
#$ $       $#
#$ $       $#
#$ $       $#
#
# $ $ $ $ $
# $ $ $ $ $
# $ $ $ $ $
#$ $ $ $ $ $ $#
#$ $ $ $ $ $ $#
#$ $ $ $ $ $ $#
################
################

// Level Five
################
################
#$ $       $#
#$ $       $#
#$ $       $#
#
# $ $ $ $ $
# $ $ $ $ $
# $ $ $ $ $
#$ $ $ $ $ $ $#
#$ $ $ $ $ $ $#
#$ $ $ $ $ $ $#
### ###
###       ###
```

Sementara sign teks berikan tentang teks atau petunjuk arah yang akan diberikan kepada player

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
{
   "1" :
   {
      "11, 7" : ["Selamat Datang di The Adventure."],
      "8, 10" : ["Gudang Perlengkapan/Ambil Item"]
   },
   "2" :
   {
      "9, 6" : ["Selamat Datang di The Adventure."],
      "6, 10" : ["Cobalah tekan E untuk mengambil dan Q untuk mendrop Item"]
   },
   "3" :
   {
      "8, 10": ["Telusuri hutan dan temukan jalan keluar"]
   }
}
```

4. Levels

Data level dibagi menjadi beberapa level yang ditentukan oleh programmer dan pada game ini kami menambahkan level 5 sebagai tambahan map.

Berikut merupakan gambaran code yang ada pada setiap level :

```
import AppEngine
from AppEngine import *

import ObjectClasses.Consumable as co
import ObjectClasses.Weapon as wn
import ObjectClasses.Item as im

import LevelParser
import Spritesheet

import random
```

Mengambil library “AppEngine” yang berfungsi untuk tampilan aplikasi pada python lalu object class yang berfungsi dalam mengambil objek baik itu objek item yang dikonsumsi player,senjata dan item-item lainnya

```
class StageFive():
   def__init_(self):
      self.groundTiles = []
      self.obstacleTiles = []
      self.hazards = []
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
        self.lvl5 = LevelParser.Parser("Level Five")
        self.lvl5.parse()

        self.itemInfo = [

        ]

        self.items = []

    def generateGround(self):
        self.lvl5.generate_ground(self.groundTiles)
```

Setiap class-class merupakan pembagian ground dan obstacle atau objek pada game

```
    def generateHazards(self):
        self.lvl5.generate_hazards(self.hazards)

    def generateObstacles(self):
        self.lvl5.generate_obstacles(self.obstacleTiles)

    def destroy(self):
        for i in self.obstacleTiles:
            i.sprite.destroy()
        #for c in self.hazards:
        #    c.sprite.destroy()
        for f in self.groundTiles:
            f.destroy()
        self.obstacleTiles.clear()
        self.hazards.clear()
        self.groundTiles.clear()
        self.stopMusic()

    def spawnTreasure(self):
        if len(self.items) > 0:
            temp = []
            for item in self.items:
                if item.spriteImage.y != 905:
                    item.spriteImage.destroy()
                    item.spriteImage = None
                    del item
                else:
                    temp.append(item)
            self.items = temp


        else:
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
        for x in range(len(self.itemInfo)):
            if self.itemInfo[x][3] == "co":
                self.items.append(co.Consumable(self.itemInfo[x][0], self.itemInfo[x][1],
self.itemInfo[x][2]))
            elif self.itemInfo[x][3] == "wn":
                self.items.append(wn.Weapon(self.itemInfo[x][0], self.itemInfo[x][1],
self.itemInfo[x][2]))
            elif self.itemInfo[x][3] == "im":
                self.items.append(im.Item(self.itemInfo[x][0], self.itemInfo[x][1],
self.itemInfo[x][2]))

        return self.items

    def startMusic(self):
        au.playMusic("Music/Level1.ogg", -1)

    def stopMusic(self):
        au.stopMusic()
```

Menampilkan musik yang berada pada data aset musik agar terdengar di level ini

```
    def checkCollision(self, hero):
        heroRect = hero.main.get_rect(topleft=(hero.x, hero.y))
        heroRect.inflate_ip(10, 10)
        for f in self.obstacleTiles:
            fRect = f.sprite.main.get_rect(topleft=(f.sprite.x, f.sprite.y))
            if f.id_ == "stationary":
                if fRect.colliderect(heroRect):
                    if fRect.collidepoint(heroRect.midleft):
                        return "west"
                    if fRect.collidepoint(heroRect.midright):
                        return "east"
                    if fRect.collidepoint(heroRect.midtop):
                        return "north"
                    if fRect.collidepoint(heroRect.midbottom):
                        return "south"
        return ""
```

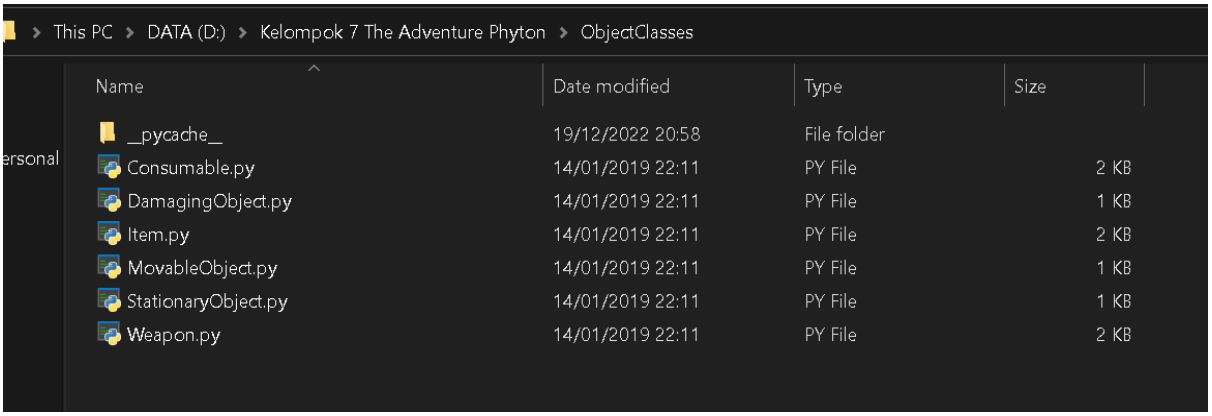
5. Music

Pada data musik berfungsi agar setiap code diatas dapat ditampilkan pada game. Musik ini diambil dari data “freemusic.net” sehingga aman dalam masalah copyright atau hak cipta musik.

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

6. ObjectClasses

Data ObjectClasses merupakan data python yang berisi tentang bagaimana item seperti senjata dan objek lain dimunculkan dalam game dibagi menjadi consumable,Damageobject,Item,MovableObject,StationaryObject dan weapon yang dsusun berdasarkan class masing-masing.



7. Sprites

Data Sprites merupakan data-data player,objek dan item dalam game yang disusun berdasarkan format gambar atau png sehingga setiap data objek berkesinambungan dalam code yang digunakan serta class-class yang disusun dalam python.

8. Main.Py

Berisikan mengenai code python yang mengatur keseluruhan game supaya berjalan dengan dengan baik. Code ini sangat rumit untuk dijabarkan karna pembagian setiap class-classnya serta level-level yang berada pada gim.

```
import AppEngine
from AppEngine import *

import Levels.LevelOne as LevelOne
import Levels.LevelTwo as LevelTwo
import Levels.LevelThree as LevelThree
import Levels.LevelFour as LevelFour
import Levels.LevelFive as LevelFive

import JsonParser
import MainMenu
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
import Character
import ObjectClasses.Weapon as Weapon
import Spritesheet

import random, time, sys, os
```

Mengambil library appengine,level, mainmenu, charachter, ObjectClasses,weapon serta Spritesheet atau gambaran player dalam bentuk gambar atau png.

```
set_window("The Adventure", 960, 960) # { 15, 14 (*64) [960, 896] }
pygame.display.set_icon(pygame.image.load("icon.png"))


levelIndex = {
   1 : LevelOne.StageOne(),
   2 : LevelTwo.StageTwo(),
   3 : LevelThree.StageThree(),
   4 : LevelFour.StageFour(),
   5 : LevelFive.StageFive()
}

rooms = {
   1 : [2, "east"], # from room 1, go to room 2 via direction east.
   2 : [3, "north", 1, "west"], # from room 2, go to room 3 via direction north. Go back to
room 1 via direction west.
   3 : [4, "west", 2, "south"],
   4 : [5, "north", 3, "east"],
   5 : [6, "south", 4, "north"]
}

roomBorders = {
   "east" : "hero.x + hero.width >= 960",
   "north" : "hero.y <= 1",
   "west" : "hero.x <= 1",
   "south" : "hero.y + hero.height >= 895"
}

oppDir =  {
   "east" : "west",
   "west" : "east",
   "south" : "north",
   "north" : "south"
}

startPositions = {
   "east" : (20, 440),
   "north" : (440, 800),
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
    "west" : (930, 440),
    "south" : (440, 80)
}


parser = JsonParser.Parser()
parser.parse("GameConfig/config.json")
musicActive = parser.settings['musicSettings']['music']
sfxActive = parser.settings['musicSettings']['SFX']
character = parser.settings['levelSettings']['player']
currentLevel = parser.settings['levelSettings']['starting_level']

previousLevel = 0
toExit = None

ss = Spritesheet.spritesheet("Sprites/" + character + "/character.png")
alpha = (0, 0, 0, 0)

parser.parse("Sprites/" + character + "/walkCycle.json")

charWalkCycleDown = parser.settings['Down']
charWalkCycleUp = parser.settings['Up']
charWalkCycleRight = parser.settings['Right']
charWalkCycleLeft = parser.settings['Left']

cycles = [charWalkCycleDown, charWalkCycleLeft, charWalkCycleRight,
charWalkCycleUp]
for element in cycles:
    for r in range(len(element)):
        element[r] = ss.image_at(tuple(element[r]), alpha)

parser.parse("GameConfig/signText.json")
signDisplay = parser.settings

menuActive = True
heroSpawned = False
inventorySpawned = False

HPred = 0
HPgreen = 255

inventorySlots = []
itemList = []

currentSelected = 0
nextAvailableSlot = 320
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
def start(heroCoords = None):
    global menuActive, itemList, obstCoords
    menuActive = False
    if 'heroChar' in globals() or 'heroChar' in locals():
        pass
    else:
        inventorySlots.clear()
        for x in range(4, 11):
            inventorySlots.append(sprite("Sprites/Inventory/inventory_slot.png", x * 64, 903,
"slot" + str(x)))

    levelIndex[currentLevel].generateGround()
    levelIndex[currentLevel].generateObstacles()
    levelIndex[currentLevel].generateHazards()
    itemList = itemList + levelIndex[currentLevel].spawnTreasure()
    if musicActive == True:
        levelIndex[currentLevel].startMusic()
    if heroCoords != None:
        spawnHero(heroCoords)
    else:
        spawnHero()


def spawnHero(heroCoords = None):
    global hero, heroChar, heroBox, heroSpawned, inventorySpawned

    if heroCoords != None:
        hero.moveToFront()
        hero.x = heroCoords[0]
        hero.y = heroCoords[1]
    else:
        hero = sprite(charWalkCycleRight[0], 700, 453, "hero")
    hero.setHP(100)
    heroSpawned = True
    inventorySpawned = True

    if 'heroChar' in globals() or 'heroChar' in locals():
        pass
    else:
        heroChar = Character.Character()
        Sword = Weapon.Weapon("Sword", 260, 905, "Sprites/BlueHairedHero/sword.png",
"Weapon")
        Sword.assignInvSlot(1)
        heroChar.addDimensions(Sword.spriteImage.width, Sword.spriteImage.height,
heroChar.availableSlot)
        heroChar.addToInventory(Sword)
        Sword.pickedUp = True
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
    itemList.append(Sword)


menu = MainMenu.Menu()
if musicActive == True:
    menu.startMusic()

hoverText = text("", 13, black, 0, 0)
signText = text("", 15, black, 0, 0)
signText.changeFont("Fonts/seagram.ttf")
textActive = False

optionClicked = False

walkCycleRate = 0
walkFrame = 0

currentDirection = "west"
tempx = 0
tempy = 0
cannotWalkHere = ""

while(True):

    if walkCycleRate != 4:
        walkCycleRate += 1
    else:
        if walkFrame != 3:
            walkFrame += 1
        else:
            walkFrame = 0
        walkCycleRate = 0

    if menuActive == True:
        menu.detectHovers()
        menu.detectLoadClick()
        if menu.detectInfoClick() == True:
            optionClicked = True
        if optionClicked == True:
            if menu.detectBackArrow() == True:
                optionClicked = False
        if menu.detectPlayClick() == True:
            start()


    for item in itemList:

        if heroChar.storage[currentSelected + 1] == item:
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
        hoverText.changeText(item.name, black)
        hoverText.x = item.spriteImage.x
        hoverText.y = item.spriteImage.y - 30

    if hero.collide(item.spriteImage) and kb.activeKeys[K_e]:
        if heroChar.findTotalFilled() == 7:
            pass
        else:
            if sfxActive == True:
                au.playSound("Music/Pick_Up.ogg")
            heroChar.addToInventory(item)
            heroChar.addDimensions(item.spriteImage.width, item.spriteImage.height,
heroChar.availableSlot - 1)
            item.assignInvSlot(heroChar.availableSlot - 1)
            item.spriteImage.main = pygame.transform.scale(item.spriteImage.main, (48, 48))
            item.pickedUp = True
            item.spriteImage.x = nextAvailableSlot
            item.spriteImage.y = 905
            done = False
            for x in range(len(heroChar.storage)):
                if done == False:
                    if heroChar.storage[x + 1] == "":
                        heroChar.availableSlot = x + 1
                        nextAvailableSlot = inventorySlots[x].x
                        done = True

    if kb.activeKeys[K_q] and heroChar.storage[currentSelected + 1] != "":
        if nextAvailableSlot < inventorySlots[currentSelected].x:
            pass
        else:
            nextAvailableSlot = inventorySlots[currentSelected].x

        if sfxActive == True:
            au.playSound("Music/Drop.ogg")
        hoverText.changeText("", black)
        itemSprite = heroChar.storage[currentSelected + 1].spriteImage
        itemSprite.moveToFront()
        dimen = heroChar.itemDimensions[currentSelected + 1].split(" ")
        itemSprite.main = pygame.transform.scale(itemSprite.main, ( int(dimen[0]) ,
int(dimen[1]) ))
        itemSprite.x = hero.x
        itemSprite.y = hero.y
        heroChar.storage[currentSelected + 1].pickedUp = False
        heroChar.storage[currentSelected + 1].invSlot = ""
        heroChar.removeFromInventory(currentSelected + 1)

    if kb.activeKeys[K_1]:
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
    currentSelected = 0
elif kb.activeKeys[K_2]:
    currentSelected = 1
elif kb.activeKeys[K_3]:
    currentSelected = 2
elif kb.activeKeys[K_4]:
    currentSelected = 3
elif kb.activeKeys[K_5]:
    currentSelected = 4
elif kb.activeKeys[K_6]:
    currentSelected = 5
elif kb.activeKeys[K_7]:
    currentSelected = 6

if inventorySpawned == True:
    for s in range(7):
        if s == currentSelected:
            inventorySlots[s].modifyImage("Sprites/Inventory/inventory_slot_selected.png")
        else:
            inventorySlots[s].modifyImage("Sprites/Inventory/inventory_slot.png")

if heroSpawned == True:
    hero.drawHealthText(hero.x - 3, hero.y - 20, 20, (HPred, HPgreen, 0), str(hero.HP))
    if (kb.activeKeys[K_w] or kb.activeKeys[K_UP]) and hero.edgeTop > 0:
        currentDirection = "north"
        hero.modifyImage(charWalkCycleUp[walkFrame])
        if cannotWalkHere != currentDirection:
            hero.y -= 4
    if (kb.activeKeys[K_s] or kb.activeKeys[K_DOWN]) and hero.edgeBottom < 896:
        currentDirection = "south"
        hero.modifyImage(charWalkCycleDown[walkFrame])
        if cannotWalkHere != currentDirection:
            hero.y += 4
    if (kb.activeKeys[K_a] or kb.activeKeys[K_LEFT]) and hero.edgeLeft > 0:
        currentDirection = "west"
        hero.modifyImage(charWalkCycleLeft[walkFrame])
        if cannotWalkHere != currentDirection:
            hero.x -= 4
    if (kb.activeKeys[K_d] or kb.activeKeys[K_RIGHT]) and hero.edgeRight < 960:
        currentDirection = "east"
        hero.modifyImage(charWalkCycleRight[walkFrame])
        if cannotWalkHere != currentDirection:
            hero.x += 4

    for item in levelIndex[currentLevel].obstacleTiles:
        if item.id_ == "damaging":
            if item.sprite.collide(hero):
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
            if hero.HP != 0:
                hero.HP -= 1

            if HPred != 255:
                HPred += 5.1
                HPred = round(HPred)
            elif HPgreen != 0:
                HPgreen -= 5.1
                HPgreen = round(HPgreen)

        tempItem = item.sprite.id_.split(" ")
        if tempItem[0] == "sign":
            if hero.x + hero.width + 32 >= item.sprite.x and hero.y + hero.height + 32 >=
item.sprite.y \
            and hero.x - 32 <= item.sprite.x + item.sprite.width and hero.y - 32<=
item.sprite.y + item.sprite.height \
            and kb.activeKeys[K_SPACE]:
                signText.changeText(signDisplay[str(currentLevel)][tempItem[1] + ", " +
tempItem[2]][0], black)
                textWidth = signText.textSurface.get_rect().width
                signWidth = item.sprite.width
                x_Offset = (textWidth - signWidth) / 2
                signText.x = item.sprite.x - x_Offset
                signText.y = item.sprite.y - 20
                textActive = True

        if kb.activeKeys[K_RETURN] and textActive == True:
            textActive = False
            signText.changeText("", black)
            signText.x = 0
            signText.y = 0

    if eval(roomBorders[rooms[currentLevel][1]]):
        previousLevel = currentLevel
        toExit = oppDir[rooms[currentLevel][1]]
        levelIndex[currentLevel].destroy()
        currentLevel = rooms[currentLevel][0]
        start(startPositions[rooms[previousLevel][1]])
    elif currentLevel != 1:
        if toExit != None:
            if eval(roomBorders[toExit]):
                previousLevel = currentLevel
                toExit = oppDir[toExit]
                levelIndex[currentLevel].destroy()
                currentLevel = rooms[currentLevel][2]
                start(startPositions[rooms[previousLevel][3]])
            elif eval(roomBorders[rooms[currentLevel][3]]):
```

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

```
previousLevel = currentLevel
toExit = oppDir[rooms[currentLevel][3]]
levelIndex[currentLevel].destroy()
currentLevel = rooms[currentLevel][2]
start(startPositions[rooms[previousLevel][3]])


if heroSpawned == True:
    temp = levelIndex[currentLevel].checkCollision(hero)
    cannotWalkHere = temp


gameLoop(black)
```

Setelah setiap library data game digabungkan maka setiap class-classnya disusun berdasarkan level-level yang sudah ditentukan .

Desain Interface

Digunakan untuk menggambarkan tampilan sistem interface/antarmuka menjadi perantara yang menjabatani interaksi antara pengguna (user) dengan program, yang berupa komponen tampilan dan tata letak dari komponen tampilan tersebut.

4 Kesimpulan

Library Pygame merupakan salah satu pilihan yang populer untuk membuat game RPG dengan bahasa pemrograman Python. Dengan Pygame, Dapat dengan mudah membuat game 2D seperti game platformer, game arcade, game puzzle, dan game role-playing game (RPG). Pygame menyediakan berbagai fitur yang berguna untuk membuat game, seperti kemampuan untuk menampilkan grafik,

Jurnal Manajemen Informatika Jayakarta
Volume 3, Nomor 1,Februari 2023 : halaman 57-73
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v3i1.1013

mengelola event, memainkan audio dan video, dan banyak lagi. Selain itu, Pygame juga dapat menambahkan fitur seperti player, enemy, item, dan lainnya sesuai dengan kebutuhan game yang dibuat.

Dengan demikian, library Pygame merupakan pilihan yang baik untuk membuat game RPG dengan bahasa pemrograman Python, karena mudah digunakan dan menyediakan berbagai fitur yang berguna untuk membuat game.

Referensi (Reference)

[1] "Contributors to Pygame". GitHub, https://github.com/rohan-bansal/The-Rogue, diakses pada Desember, 2022

[2] Codepolitan, 2022, Learn How to Make 2D Games with Python and Arcade, https://www.codepolitan.com/learning-how-to-make-game-2d-with-python-dan-arcade5adff78d1db9b/ , diakses pada Desember, 2022

[3] Romzi, Muhammad & Kurniawan, Budi 2020, 'Python Programming Learning With Algorithmic Logic Approach', Journal of Informatics Engineering Mahakarya , vol. 3, no. 2, hh. 37-44.

[4] Codepolitan, 2022, Learn How to Make 2D Games with Python and Arcade, https://www.codepolitan.com/learning-how-to-make-game-2d-with-python-dan-arcade5adff78d1db9b/ , diakses pada Desember, 2022

[5] Youtube, 2022, Learn Python by Building Five Games - Full Course, *https://appkey.id/pembuatan-application/mobile-programming/python-ide*, diakses pada Desember, 2022