Computer Science Project

Mystic swords

Using Python

**Done by**

**S.Manoj Srivatsava**

**ACKNOWLEDGEMENT**

We hereby acknowledge all those people, who have helped for the successful completion of our project to a great extent.

We would also like to convey our sincere gratitude to our principal Smt. J. Geetha and the D.A.V School management for having provide us with complete access to all materials and information source available in the computer laboratory.

We express our deepest gratitude to our Computer Science teachers for their encouragement during the course of the project titled “Mystic Swords”

We would also like to extend our thanks to other members of the computer department for their co-operation.

**Team Members:**

25 - S.Manoj Srivatsava

**Table of Contents**

Introduction……………………………………………...5

Software used……………………………………………6

Flow diagram……………………………………………7

Source Code……………………………………………..8

Application Flow Home Screens

Loading screen………………….………………….37

Main menu………………….………………………37

How to play screen ………………….……………..38

Settings………………….………………………….39

Game screen………………….……………………..39

Pause menu………………….………………………40

Game over screen………………….………………..40

Report Module

Highscore………………….………………………..41

Files part of this project

Executable Application……………………………..42

Binary File ………………………………………….42

Bibliography………………………………………………43

**Introduction**

Video games otherwise know as computer games where the pinnacle of entertainment around the 1970’s. These video games are classified based on the platform their played in, like arcade game’s console game’s…etc.

After the arrival of the Internet multiplayer games over long distances was possible which skyrocketed the popularity of video games.

One category of video games at that time were called top down games, some examples are space invaders,lunar lander and pong the first top down game ever made was called spacewar! Made in 1962.

These top down games got better with time and evolved to games like pokemon ruby, pokemon emerald.The game that I will be presenting as the cs project is also of the top down variety

**MYSTIC SWORDS:**

This game is partially based on Greek mythology as the main character in the game is based on Theseus and the sword that he wields which was given to him by Ariadne

In this games story Theseus uses the sword that was given to him to prevent the ghosts from reaching his city of Troezen.The game progresses by waves timing how long “Theseus” can defend his city from the ghost.The time is recorded and stored in a binary file which can be accessed later in game for the high scores.The speed and number of ghosts increase along with the wave.

This game was made using python and a popular library in python for game development called pygame and open cv for the loading screen

**Software Used**

Python 3.11 on Windows 11

Python Modules Used:

* pygame
* threading
* random
* time
* pickle
* open cv
* datetime

The scores are stored into a separate binary file named highscore.dat

Textures were loaded from

The font used in the game were from

• [https://www.fontspace.com/angel-rhapsody-font-f492000](https://www.fontspace.com/angel-rhapsody-font-f49200)/ (For the text used)

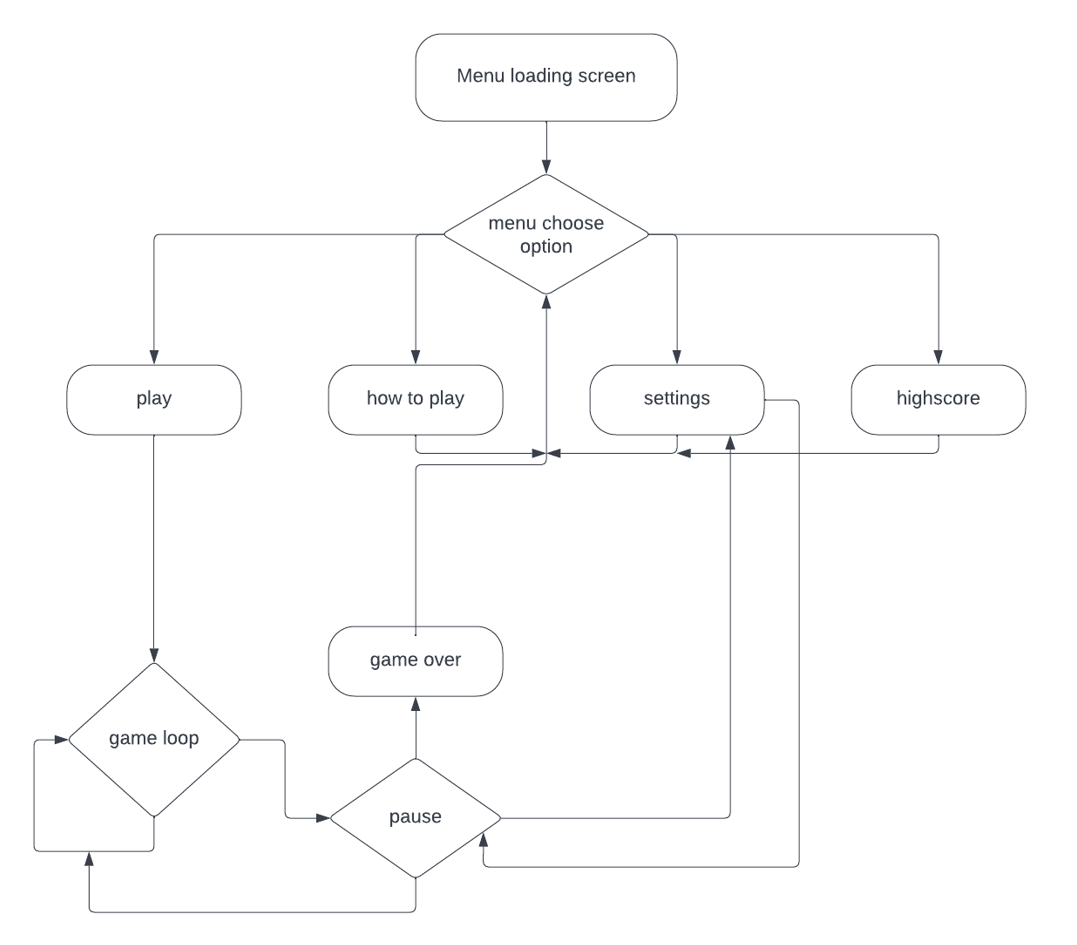
The background map was made using the textures from

•https://game-endeavor.itch.io/mystic-woods (For the map)

The menu background and loading screen are from

•https://www.youtube.com/watch?v=MPiILYNStd8

**Flow Diagram**



**Main.py**

import pygame

import menu

import singleplayergame

pygame.display.set\_caption("Mystic Sword")

icon = pygame.image.load("player/sright 2.png")

pygame.display.set\_icon(icon)

screen = pygame.display.set\_mode((1366,768))

#right left up down space \/ askii values

keylist=["100","97","119","115","32"]

menu.loadingmenu()

while True:

menu\_info,keylist = menu.menu(keylist)

pygame.display.update()

pygame.init()

if menu\_info[0] == "play":

singleplayergame.keys(keylist)

singleplayergame.main()

**Menu.py**

import cv2

import pygame

import highscoredisplay

import settingsdisplay

import time

import pickle

screen = pygame.display.set\_mode((1366,768),pygame.RESIZABLE)

def menu\_music():

pygame.mixer.init()

pygame.mixer.music.load("menu\menu.mp3")

pygame.mixer.music.play(1)

def loadingmenu():

vidcap = cv2.VideoCapture('menu\menu frame.mkv')

menu\_music()

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

frame\_loaded,menu\_frame = vidcap.read()

if frame\_loaded == True:

menu\_frame = cv2.cvtColor(menu\_frame, cv2.COLOR\_RGB2BGR)

menu\_frame = cv2.rotate(menu\_frame, cv2.ROTATE\_90\_COUNTERCLOCKWISE)

menu\_frame = cv2.flip(menu\_frame, 0)

menu\_frame = pygame.pixelcopy.make\_surface(menu\_frame)

frame\_x,frame\_y=menu\_frame.get\_size()

screen\_x,screen\_y=screen.get\_size()

screen\_ratio=screen\_x/screen\_y

img\_ratio=frame\_x/frame\_y

if screen\_ratio < img\_ratio:

frame\_x=screen\_x

frame\_y=int(frame\_x/img\_ratio)

elif screen\_ratio > img\_ratio:

frame\_y=screen\_y

frame\_x=int(frame\_y\*img\_ratio)

menu\_frame = pygame.transform.scale(menu\_frame, (frame\_x, frame\_y))

screen.blit(menu\_frame,((screen\_x-frame\_x)//2,(screen\_y-frame\_y)//2))

pygame.display.update()

else:

break

def menu(keylist):

screen\_x,screen\_y=screen.get\_size()

pygame.mouse.set\_pos(screen\_x/2,screen\_y/2)

next = pygame.image.load(r'menu\next.png')

back = pygame.image.load(r'menu\back.png')

play = pygame.image.load('menu\play.png')

how\_to\_play = pygame.image.load('menu\how to play.png')

settings = pygame.image.load('menu\settings.png')

highscore = pygame.image.load('menu\highscore.png')

clearscore = pygame.image.load('menu\clear.png')

htp = [pygame.image.load('menu\htp1.png'),pygame.image.load('menu\htp2.png')]

sword\_left = pygame.image.load('menu\sword left.png')

sword\_right = pygame.image.load('menu\sword right.png')

nextno = 0

menu\_dict={"play":True,"settings":True,"how to play":True,"highscore":True,"singleplayer":False,"multiplayer":False,"create":False,"join":False,"back":False,"highmenu":False,"settmenu":False,"htpmenu":False,"next":False,"clearscore":False}

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

mouse\_x,mouse\_y = pygame.mouse.get\_pos()

menu\_bg(mouse\_x,mouse\_y)

screen\_x,screen\_y=screen.get\_size()

play\_rect = pygame.Rect(screen\_x/2-115,screen\_y/2-49-270,230,98)

how\_to\_play\_rect = pygame.Rect(screen\_x/2-287,screen\_y/2-49-85,574,98)

settings\_rect = pygame.Rect(screen\_x/2-231,screen\_y/2-49+85,463,98)

highscore\_rect = pygame.Rect(screen\_x/2-258,screen\_y/2-49+270,504,98)

back\_rect = pygame.Rect(screen\_x-240,screen\_y-140,217,98)

clear\_rect = pygame.Rect(screen\_x-240,screen\_y-140-150,217,98)

next\_rect = pygame.Rect(screen\_x-240,screen\_y-140,217,98)

if menu\_dict["back"]==True:

screen.blit(back,(screen\_x-240,screen\_y-140))

if menu\_dict["back"]==True and back\_rect.collidepoint(mouse\_x,mouse\_y):

if event.type == pygame.MOUSEBUTTONDOWN:

menu\_dict["back"] = False

menu\_dict["play"] = True

menu\_dict["settings"] = True

menu\_dict["highscore"] = True

menu\_dict["how to play"] = True

menu\_dict["highmenu"] = False

menu\_dict["settmenu"] = False

menu\_dict["singleplayer"] = False

menu\_dict["multiplayer"] = False

menu\_dict["next"] = False

menu\_dict["htpmenu"] = False

menu\_dict["clearscore"] = False

menu\_dict["create"] = False

menu\_dict["join"] = False

nextno = 0

if menu\_dict["htpmenu"]==True:

screen.blit(htp[nextno-1],(10,10))

pygame.display.update()

if menu\_dict["next"]==True:

screen.blit(next,(screen\_x-240,screen\_y-140))

if menu\_dict["next"]==True and next\_rect.collidepoint(mouse\_x,mouse\_y):

if event.type == pygame.MOUSEBUTTONDOWN:

if nextno == 1:

menu\_dict["back"] = True

menu\_dict["next"] = False

nextno +=1

else:

nextno +=1

time.sleep(0.1)

if menu\_dict["settmenu"]==True:

keylist = settingsdisplay.display(event,keylist)

if menu\_dict["highmenu"]==True:

highscoredisplay.display()

if menu\_dict["clearscore"] == True:

screen.blit(clearscore,(screen\_x-240,screen\_y-140-150))

if menu\_dict["clearscore"] == True and clear\_rect.collidepoint(mouse\_x,mouse\_y):

if event.type == pygame.MOUSEBUTTONDOWN:

fh = open("highscore.dat","wb")

pickle.dump([0,0,"--/--/--"],fh)

fh.flush()

if menu\_dict["play"]==True:

screen.blit(play,(screen\_x/2-115,screen\_y/2-49-270))

if menu\_dict["play"] == True and play\_rect.collidepoint(mouse\_x,mouse\_y):

screen.blit(sword\_left,(screen\_x/2-115-169,screen\_y/2-42-270))

screen.blit(sword\_right,(screen\_x/2+115,screen\_y/2-42-270))

pygame.display.update()

if event.type == pygame.MOUSEBUTTONDOWN:

return ["play"],keylist

if menu\_dict["settings"]==True:

screen.blit(settings,(screen\_x/2-231,screen\_y/2-49+85))

if menu\_dict["settings"] == True and settings\_rect.collidepoint(mouse\_x,mouse\_y):

screen.blit(sword\_left,(screen\_x/2-231-169,screen\_y/2-42+85))

screen.blit(sword\_right,(screen\_x/2+231,screen\_y/2-42+85))

pygame.display.update()

if event.type == pygame.MOUSEBUTTONDOWN:

menu\_dict["play"] = False

menu\_dict["settings"] = False

menu\_dict["highscore"] = False

menu\_dict["how to play"] = False

menu\_dict["back"] = True

menu\_dict["settmenu"] = True

if menu\_dict["how to play"]==True:

screen.blit(how\_to\_play,(screen\_x/2-287,screen\_y/2-49-85))

if menu\_dict["how to play"] == True and how\_to\_play\_rect.collidepoint(mouse\_x,mouse\_y):

screen.blit(sword\_left,(screen\_x/2-287-169,screen\_y/2-42-85))

screen.blit(sword\_right,(screen\_x/2+287,screen\_y/2-42-85))

pygame.display.update()

if event.type == pygame.MOUSEBUTTONDOWN:

menu\_dict["play"] = False

menu\_dict["settings"] = False

menu\_dict["highscore"] = False

menu\_dict["how to play"] = False

menu\_dict["htpmenu"] = True

menu\_dict["next"] = True

nextno += 1

if menu\_dict["highscore"]==True:

screen.blit(highscore,(screen\_x/2-252,screen\_y/2-49+270))

if menu\_dict["highscore"] == True and highscore\_rect.collidepoint(mouse\_x,mouse\_y):

screen.blit(sword\_left,(screen\_x/2-252-169,screen\_y/2-42+270))

screen.blit(sword\_right,(screen\_x/2+252,screen\_y/2-42+270))

pygame.display.update()

if event.type == pygame.MOUSEBUTTONDOWN:

menu\_dict["play"] = False

menu\_dict["settings"] = False

menu\_dict["highscore"] = False

menu\_dict["how to play"] = False

menu\_dict["highmenu"] = True

menu\_dict["back"] = True

menu\_dict["clearscore"] = True

pygame.display.update()

screen.fill((0,0,0))

def menu\_bg(mouse\_x,mouse\_y):

move\_menu\_frame = pygame.image.load('menu\menu main2.png')

screen\_x,screen\_y=screen.get\_size()

move\_x,move\_y=(1920-680-mouse\_x,1080-mouse\_y-400)

screen.blit(move\_menu\_frame,(screen\_x-680-mouse\_x,screen\_y-mouse\_y-400))

**Highscoredisplay.py**

import pygame

import pickle

pygame.init()

screen = pygame.display.set\_mode((1366,768),pygame.RESIZABLE)

font = pygame.font.Font('AngelRhapsoy.ttf',64)

def display():

fh = open("highscore.dat","rb")

data = []

while True:

try:

rec = pickle.load(fh)

data.append(rec)

except EOFError:

fh.close()

break

datay = 50

tempdata = font.render("Position",True,(255,0,0))

screen.blit(tempdata,(100,datay))

tempdata = font.render("Time",True,(255,0,0))

screen.blit(tempdata,(400,datay))

tempdata = font.render("Date",True,(255,0,0))

screen.blit(tempdata,(700,datay))

datay += 100

x=1

for i in data:

if x > 10:

break

x+=1

tempdata = font.render(str(i[0]),True,(255,0,0))

screen.blit(tempdata,(100,datay))

tempdata = font.render(str(i[1]),True,(255,0,0))

screen.blit(tempdata,(400,datay))

tempdata = font.render(str(i[2]),True,(255,0,0))

screen.blit(tempdata,(700,datay))

datay += 60

pygame.display.update()

**Settingsdisplay.py**

import pygame

pygame.init()

screen = pygame.display.set\_mode((1366,768),pygame.RESIZABLE)

font = pygame.font.Font('AngelRhapsoy.ttf',64)

spacefont = pygame.font.Font('AngelRhapsoy.ttf',48)

tile = pygame.image.load(r'menu\tile.png')

colltile = [pygame.Rect(362,242,100,100),pygame.Rect(122,242,100,100),pygame.Rect(242,122,100,100),pygame.Rect(242,242,100,100),pygame.Rect(552,242,100,100)]

def display(event,keylist):

screen.blit(tile,(240,120))

screen.blit(tile,(120,240))

screen.blit(tile,(240,240))

screen.blit(tile,(360,240))

screen.blit(tile,(550,240))

screen.blit(font.render("Change Controls",True,(255,0,0)),(10,10))

if keylist[0] == "32":

right = spacefont.render(chr(int(keylist[0])),True,(255,0,0))

screen.blit(up,(275-30,146))

else:

right = font.render(chr(int(keylist[0])),True,(255,0,0))

screen.blit(right,(395,266))

if keylist[1] == "32":

left = spacefont.render(chr(int(keylist[1])),True,(255,0,0))

screen.blit(left,(155-30,266))

else:

left = font.render(chr(int(keylist[1])),True,(255,0,0))

screen.blit(left,(155,266))

if keylist[2] == "32":

up = spacefont.render(chr(int(keylist[2])),True,(255,0,0))

screen.blit(down,(275-30,266))

else:

up = font.render(chr(int(keylist[2])),True,(255,0,0))

screen.blit(up,(275,146))

if keylist[3] == "32":

down = spacefont.render(chr(int(keylist[3])),True,(255,0,0))

screen.blit(down,(275-30,266))

else:

down = font.render(chr(int(keylist[3])),True,(255,0,0))

screen.blit(down,(275,266))

if keylist[4] == "32":

attack = spacefont.render("space",True,(255,0,0))

screen.blit(attack,(585-30,266))

else:

attack = font.render(chr(int(keylist[4])),True,(255,0,0))

screen.blit(attack,(585,266))

colltileno = 0

for i in colltile:

if i.collidepoint(pygame.mouse.get\_pos()) and event.type == pygame.MOUSEBUTTONDOWN:

loop = True

while loop:

screen.blit(font.render("enter a control other than space",True,(255,0,0)),(630,20))

for event2 in pygame.event.get():

if event.type == pygame.QUIT:

quit()

pygame.display.update()

if event2.type == pygame.QUIT:

pygame.quit()

if event2.type == pygame.KEYDOWN:

if str(event2.key) not in keylist:

keylist[colltileno] = str(event2.key)

return keylist

colltileno += 1

return keylist

**Singleplayergame.py**

from datetime import datetime

import pygame

import time

import threading

import random

import pickle

import settingsdisplay

screen = pygame.display.set\_mode((1366,768))

pygame.init()

keylist=[]

playerdata=[1200,380,"down",10]

font = pygame.font.Font('AngelRhapsoy.ttf',64)

map = pygame.image.load('map/map.png')

up1 = pygame.image.load('player/up 1.png')

up2 = pygame.image.load('player/up 2.png')

up3 = pygame.image.load('player/up 3.png')

down1 = pygame.image.load('player/down 1.png')

down2 = pygame.image.load('player/down 2.png')

down3 = pygame.image.load('player/down 3.png')

right1 = pygame.image.load('player/right 1.png')

right2 = pygame.image.load('player/right 2.png')

right3 = pygame.image.load('player/right 3.png')

left1 = pygame.image.load('player/left 1.png')

left2 = pygame.image.load('player/left 2.png')

left3 = pygame.image.load('player/left 3.png')

rightlist = [right1,right2,right3]

leftlist = [left1,left2,left3]

uplist = [up1,up2,up3]

downlist = [down1,down2,down3]

increase = 56

for i in range(0,3):

rightlist[i] = pygame.transform.scale(rightlist[i], (increase,increase))

rightlist[i].set\_colorkey((255,255,255))

uplist[i] = pygame.transform.scale(uplist[i], (increase,increase))

uplist[i].set\_colorkey((255,255,255))

downlist[i] = pygame.transform.scale(downlist[i], (increase,increase))

downlist[i].set\_colorkey((255,255,255))

leftlist[i] = pygame.transform.scale(leftlist[i], (increase,increase))

leftlist[i].set\_colorkey((255,255,255))

moveno = [0,0,0,0]

sup1 = pygame.image.load('player/sup 1.png')

sup2 = pygame.image.load('player/sup 2.png')

sup3 = pygame.image.load('player/sup 3.png')

sdown1 = pygame.image.load('player/sdown 1.png')

sdown2 = pygame.image.load('player/sdown 2.png')

sdown3 = pygame.image.load('player/sdown 3.png')

sright1 = pygame.image.load('player/sright 1.png')

sright2 = pygame.image.load('player/sright 2.png')

sright3 = pygame.image.load('player/sright 3.png')

sleft1 = pygame.image.load('player/sleft 1.png')

sleft2 = pygame.image.load('player/sleft 2.png')

sleft3 = pygame.image.load('player/sleft 3.png')

sup1 = pygame.transform.scale(sup1, (increase,increase))

sup2 = pygame.transform.scale(sup2, (increase,increase))

sup3 = pygame.transform.scale(sup3, (increase,increase))

sdown1 = pygame.transform.scale(sdown1, (54,68))

sdown2 = pygame.transform.scale(sdown2, (66,68))

sdown3 = pygame.transform.scale(sdown3, (88,92))

sright1 = pygame.transform.scale(sright1, (66,66))

sright2 = pygame.transform.scale(sright2, (78,60))

sright3 = pygame.transform.scale(sright3, (84,50))

sleft1 = pygame.transform.scale(sleft1, (66,66))

sleft2 = pygame.transform.scale(sleft2, (78,60))

sleft3 = pygame.transform.scale(sleft3, (84,50))

ghostmove = pygame.image.load('player\gmove.png')

ghostmove1 = pygame.image.load('player\gmove1.png')

ghostmove2 = pygame.image.load('player\gmove2.png')

ghoststeady = pygame.image.load('player\gsteady.png')

ghostspeed = 10

ghostlist = []

maxghostno = 1

waveno = 0

steadyno = 1

escapeloop = False

sword\_rect = sword\_rect = pygame.Rect(-100,-100,10,10)

swordno = [0,0,0,0]

pygame.init()

runtime = 1

running = True

def resetvalues():

global runtime,ghostlist,waveno,maxghostno,playerdata,running

runtime,ghostlist,waveno,maxghostno,playerdata,running = 1,[],0,1,[1200,380,"down",10],True

def gameover():

global ghostlist,running

ghostlist = []

screen.blit(font.render("GAME OVER",True,(255,0,0)),(490,150))

screen.blit(font.render("Time : "+str(runtime),True,(255,0,0)),(530,280))

screen.blit(font.render("Return To Menu",True,(255,0,0)),(452,410))

return\_rect = pygame.Rect(452,410,830-452,460-100)

pygame.display.update()

addhighscore()

gameoverloop = True

while gameoverloop:

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

if return\_rect.collidepoint(pygame.mouse.get\_pos()):

if event.type == pygame.MOUSEBUTTONDOWN:

running = False

gameoverloop = False

def escape():

global keylist,running,escapeloop

screen.blit(font.render("GAME PAUSED!!!!!",True,(255,0,0)),(420,50))

screen.blit(font.render("Resume",True,(255,0,0)),(540,150+100))

screen.blit(font.render("Settings",True,(255,0,0)),(530,280+100))

screen.blit(font.render("End game",True,(255,0,0)),(526,410+100))

back\_rect = pygame.Rect(1366-240,768-140,217,98)

resume\_rect = pygame.Rect(540,150+100,720-540,300-250)

settings\_rect = pygame.Rect(530,280+100,720-530,330-280)

return\_rect = pygame.Rect(526,410+100,830-452,460-100)

back = pygame.image.load(r'menu\back.png')

pygame.display.update()

escapeloop = True

while escapeloop:

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

if resume\_rect.collidepoint(pygame.mouse.get\_pos()):

if event.type == pygame.MOUSEBUTTONDOWN:

escapeloop = False

if return\_rect.collidepoint(pygame.mouse.get\_pos()):

if event.type == pygame.MOUSEBUTTONDOWN:

running = False

escapeloop = False

if settings\_rect.collidepoint(pygame.mouse.get\_pos()):

if event.type == pygame.MOUSEBUTTONDOWN:

setloop = True

while setloop:

for event2 in pygame.event.get():

screen.blit(map,(0,0))

keylist = settingsdisplay.display(event2,keylist)

screen.blit(back,(1366-240,768-140))

pygame.display.update()

if back\_rect.collidepoint(pygame.mouse.get\_pos()) and event2.type == pygame.MOUSEBUTTONDOWN:

setloop = False

escapeloop = False

def addhighscore():

fh = open("highscore.dat","rb")

data = []

while True:

try:

rec = pickle.load(fh)

data.append([rec[1],rec[2]])

except EOFError:

fh.close()

break

data.append([runtime,datetime.now().strftime(r"%d/%m/%Y")])

fh = open("highscore.dat","wb")

pos = 1

for i in sorted(data)[::-1]:

pickle.dump([pos,i[0],i[1]],fh)

pos += 1

fh.flush()

def keys(l):

global keylist

keylist=l

def keysheld(pressedlist,event):

if event.type == pygame.KEYDOWN and event.key not in pressedlist:

pressedlist.append(str(event.key))

if event.type == pygame.KEYUP:

try:

pressedlist.remove(str(event.key))

except:

pass

return pressedlist

def timedis():

if runtime == runtime//1:

screen.blit(font.render(str(runtime)+"0",True,(255,0,0)),(1200,10))

else:

screen.blit(font.render(str(runtime)+str(random.randint(0,9)),True,(255,0,0)),(1200,10))

def ghostdis():

for i in ghostlist:

if steadyno > 10:

x = random.randint(1,3)

if x==1:

screen.blit(ghostmove,(i[0],i[1]))

elif x==2:

screen.blit(ghostmove1,(i[0],i[1]))

else:

screen.blit(ghostmove2,(i[0],i[1]))

else:

screen.blit(ghoststeady,(i[0],i[1]))

def swordframe():

timedis()

ghostdis()

pygame.display.update()

time.sleep(0.04)

def mapdis():

screen.blit(map,(0,0))

def sword():

global sword\_rect

if playerdata[2] == "down":

sword\_rect = pygame.Rect(playerdata[0]-2,playerdata[1]-2,50,100)

#pygame.draw.rect(screen, (255,0,0), sword\_rect)

mapdis()

screen.blit(sdown1,(playerdata[0]-4,playerdata[1]-14))

swordframe()

mapdis()

screen.blit(sdown2,(playerdata[0],playerdata[1]))

swordframe()

mapdis()

screen.blit(sdown3,(playerdata[0]-19,playerdata[1]-12))

swordframe()

elif playerdata[2] == "right":

sword\_rect = pygame.Rect(playerdata[0]-2,playerdata[1]-2,100,50)

mapdis()

screen.blit(sright1,(playerdata[0]-14,playerdata[1]-18))

swordframe()

mapdis()

screen.blit(sright2,(playerdata[0],playerdata[1]))

swordframe()

mapdis()

screen.blit(sright3,(playerdata[0],playerdata[1]))

swordframe()

elif playerdata[2] == "left":

sword\_rect = pygame.Rect(playerdata[0]-52,playerdata[1]-2,100,50)

mapdis()

screen.blit(sleft1,(playerdata[0],playerdata[1]-18))

swordframe()

mapdis()

screen.blit(sleft2,(playerdata[0]-20,playerdata[1]))

swordframe()

mapdis()

screen.blit(sleft3,(playerdata[0]-14,playerdata[1]))

swordframe()

elif playerdata[2] == "up":

sword\_rect = pygame.Rect(playerdata[0]-2,playerdata[1]-52,50,100)

mapdis()

screen.blit(sup1,(playerdata[0],playerdata[1]))

swordframe()

mapdis()

screen.blit(sup2,(playerdata[0],playerdata[1]))

swordframe()

mapdis()

screen.blit(sup3,(playerdata[0]-10,playerdata[1]-10))

swordframe()

def movement(pressedlist):

directionlist = []

xyposreturn = [playerdata[0],playerdata[1]]

for i in pressedlist:

if pressedlist[0] == "27":

escape()

pressedlist.remove("27")

if i in keylist[0:4]:

directionlist.append(i)

if i in keylist[4:5]:

sword()

sword\_rect = pygame.Rect(-100,-100,10,10)

return playerdata[0],playerdata[1]

if len(directionlist) == 1:

if directionlist[-1] == keylist[0]: #right

if moveno[0] == 2:

moveno[0] = 0

else:

moveno[0] +=1

playerdata[2] = "right"

xyposreturn = playerdata[0]+playerdata[3],playerdata[1]

if directionlist[-1] == keylist[1]: #left

if moveno[1] == 2:

moveno[1] = 0

else:

moveno[1] +=1

playerdata[2] = "left"

xyposreturn = playerdata[0]-playerdata[3],playerdata[1]

if directionlist[-1] == keylist[2]: #up

if moveno[2] == 2:

moveno[2] = 0

else:

moveno[2] +=1

playerdata[2] = "up"

xyposreturn = playerdata[0],playerdata[1]-playerdata[3]

if directionlist[-1] == keylist[3]: #down

if moveno[3] == 2:

moveno[3] = 0

else:

moveno[3] +=1

playerdata[2] = "down"

xyposreturn = playerdata[0],playerdata[1]+playerdata[3]

elif len(directionlist) == 2:

if directionlist[-1] == keylist[0] and directionlist[-2] == keylist[2] or directionlist[-1] == keylist[2] and directionlist[-2] == keylist[0]: #righy and up

if moveno[0] == 2:

moveno[0] = 0

else:

moveno[0] +=1

playerdata[2] = "right"

xyposreturn = playerdata[0]+playerdata[3],playerdata[1]-playerdata[3]

if directionlist[-1] == keylist[0] and directionlist[-2] == keylist[3] or directionlist[-1] == keylist[3] and directionlist[-2] == keylist[0]: #right and down

if moveno[0] == 2:

moveno[0] = 0

else:

moveno[0] +=1

playerdata[2] = "right"

xyposreturn = playerdata[0]+playerdata[3],playerdata[1]+playerdata[3]

if directionlist[-1] == keylist[1] and directionlist[-2] == keylist[2] or directionlist[-1] == keylist[2] and directionlist[-2] == keylist[1]: #left and up

if moveno[1] == 2:

moveno[1] = 0

else:

moveno[1] +=1

playerdata[2] = "left"

xyposreturn = playerdata[0]-playerdata[3],playerdata[1]-playerdata[3]

if directionlist[-1] == keylist[1] and directionlist[-2] == keylist[3] or directionlist[-1] == keylist[3] and directionlist[-2] == keylist[1]: #left and down

if moveno[1] == 2:

moveno[1] = 0

else:

moveno[1] +=1

playerdata[2] = "left"

xyposreturn = playerdata[0]-playerdata[3],playerdata[1]+playerdata[3]

if directionlist[-1] == keylist[1] and directionlist[-2] == keylist[0]: #right and left

if moveno[0] == 2:

moveno[0] = 0

else:

moveno[0] +=1

playerdata[2] = "right"

xyposreturn = playerdata[0]+playerdata[3],playerdata[1]

if directionlist[-1] == keylist[0] and directionlist[-2] == keylist[1]: #left and right

if moveno[1] == 2:

moveno[1] = 0

else:

moveno[1] +=1

playerdata[2] = "left"

xyposreturn = playerdata[0]-playerdata[3],playerdata[1]

if directionlist[-1] == keylist[3] and directionlist[-2] == keylist[3]: #down and up

if moveno[3] == 2:

moveno[3] = 0

else:

moveno[3] +=1

playerdata[2] = "down"

xyposreturn = playerdata[0],playerdata[1]-playerdata[3]

if directionlist[-1] == keylist[3] and directionlist[-2] == keylist[3]: #up and down

if moveno[3] == 2:

moveno[3] = 0

else:

moveno[3] +=1

playerdata[2] = "up"

xyposreturn = playerdata[0],playerdata[1]+playerdata[3]

def emptret():

return playerdata[0],playerdata[1]

if xyposreturn[0] > 1310 and xyposreturn[1] > 694:

return emptret()

elif xyposreturn[0] < 4 and xyposreturn[1] < 40:

return emptret()

elif xyposreturn[0] > 1310 and xyposreturn[1] < 40:

return emptret()

elif xyposreturn[0] < 4 and xyposreturn[1] > 694:

return emptret()

elif xyposreturn[0] > 1310:

return playerdata[0],xyposreturn[1]

elif xyposreturn[0] < 4:

return playerdata[0],xyposreturn[1]

elif xyposreturn[1] > 694:

return xyposreturn[0],playerdata[1]

elif xyposreturn[1] < 40:

return xyposreturn[0],playerdata[1]

elif xyposreturn == playerdata[0:2]:

moveno[0],moveno[1],moveno[2],moveno[3] = 0,0,0,0

return playerdata[0],playerdata[1]

else:

return xyposreturn

def ghost(i):

global running

try:

disx = random.randint(ghostspeed-2,ghostspeed+2)

disy = random.randint(8,11)

ran = random.randint(1,3)

ghostlist[i][0] -= disx #displacment

if ran == 1 and 200 < ghostlist[i][1] + disy: #displacmenty

ghostlist[i][1] -= disy

if ran == 2 and 694 > ghostlist[i][1] + disy:

ghostlist[i][1] += disy

if ghostlist[i][0] < 250:

running = False

if sword\_rect.collidepoint(ghostlist[i][0]+10,ghostlist[i][1]+15):

ghostlist.remove(ghostlist[i])

time.sleep(0.08)

except IndexError:

pass

def wave():

global waveno,maxghostno

waveno += 1

maxghostno += random.randint(2,6)#noincrease

for i in range(0,maxghostno):

ghostlist.append([random.randint(1100,1310),random.randint(40,649),2])

while ghostlist != []:

if escapeloop == False:

for i in range(len(ghostlist)):

temp = len(ghostlist)

ghostthread = threading.Thread(target = ghost(i))

ghostthread.start()

time.sleep(0.02)

def timer():

global runtime

while running:

if not escapeloop:

time.sleep(0.01)

runtime = round(runtime + 0.01,2)

def main():

resetvalues()

global steadyno

pressedlist = []

timethread = threading.Thread(target = timer)

timethread.start()

while running:

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

pressedlist = keysheld(pressedlist,event)

playerdata[0],playerdata[1] = movement(pressedlist)

screen.blit(map,(0,0))

steadyno += 1

timedis()

ghostdis()

if playerdata[2] == "right":

screen.blit(rightlist[moveno[0]],(playerdata[0],playerdata[1]))

if playerdata[2] == "left":

screen.blit(leftlist[moveno[1]],(playerdata[0],playerdata[1]))

if playerdata[2] == "up":

screen.blit(uplist[moveno[2]],(playerdata[0],playerdata[1]))

if playerdata[2] == "down":

screen.blit(downlist[moveno[3]],(playerdata[0],playerdata[1]))

time.sleep(0.02)

pygame.display.update()

if ghostlist == []:

steadyno = 0

playerdata[0],playerdata[1] = 200,380

screen.blit(map,(0,0))

screen.blit(rightlist[moveno[0]],(playerdata[0],playerdata[1]))

screen.blit(font.render("WAVE "+str(waveno+1),True,(255,0,0)),(550,350))

pygame.display.update()

time.sleep(0.5)

wavethread = threading.Thread(target = wave)

wavethread.start()

time.sleep(0.04)

gameover()

**Application Flow:**

**Home screens**

**Loading screen**



**Main menu**



**How to play screen**



**Settings**



**Game screen**



**Pause menu**



**Game over screen**



**Report Moduel**

**Highscore**

**Files part of the project**



main.py



menu.py



settingsdisplay.py



Highscoredisplay.py



Singleplayer.py

**Binary files**

 Highscore.dat

**TrueType Font**

AngelRhapsoy.ttf

**Bibliography**

• https://www.pygame.org/docs/