#Ver 3

#changes = The way that the shop operates (menu not a free roam

#contents:

#1) The main game

#2) loading

#3) map1

#4) map2

#5) Pause menu

#6) Shop

#7) Turn based combat

#8) Menus

#9) Configuration

import pyglet

from tkinter import \*

from pyglet.window import key

import random

import time

from pyglet.window import mouse

#------------------------------------The main game------------------------------------------

#This class will contain all of the code for the main game form all the gameplay aspects

#to the sprites etc

class Main\_game:

#This is the initiation class which will start the main game

def \_\_init\_\_(self):

self.loading(1)

#------------------------------------------loading------------------------------------------

#This meathod is going to be the loading screen for my game, it is going to handle all of

#the pathways, for example, if the user wants to switch maps, they have to go through

#the loading screen. This makes it easier to track and change as the program gets longer

# (dont have to scroll all the way through my program to find one specific variable).

def loading(self,flag):

#This creates the window which the result will be shown on

# loading\_screen=pyglet.window.Window(fullscreen=True,visible=True)

#These three commands are the different loading screen images that are going to show

# loading\_text\_1=pyglet.resource.image("loading1.png")

# loading\_text\_2=pyglet.resource.image("loading2.png")

#loading\_text\_3=pyglet.resource.image("loading3.png")

#This is the sequence of images that are going to become the animation for the loading

#screen (nessesary when manually creating an animation through pyglet).

#loading\_sequence=(loading\_text\_1,loading\_text\_2,loading\_text\_3)

# loading\_animation=pyglet.image.Animation.from\_image\_sequence(loading\_sequence,0.5,True)

#loading=pyglet.sprite.Sprite(img=loading\_animation,x=0,y=0)

#Draws the specified objects onto the screen

#@loading\_screen.event

#def on\_draw():

# loading\_screen.clear()

# loading.draw()

#Makes the loading screen visible

#loading\_screen.set\_visible(True)

#Different pathways that the loading screen can take which will lead to a different part of the

#game being triggered.

if flag==1:

Main\_game.map1(Main\_game,500,1000)

elif flag==2:

Main\_game.map2(Main\_game,500)

elif flag==3:

Main\_game.shop1(Main\_game)

elif flag==4:

Main\_game.map3(Main\_game,600)

#-----------------------------------------map 1----------------------------------------------

#This meathod will contain all of the code for the free roam section of the game

def map1(self,spawny,spawnx):

#This creates a screen and sets it to false, this is so the program can load

#all of the data before it is show to the user, this will hopefully increase

#the efficiency of my program

world=pyglet.window.Window(visible=False, fullscreen=True)

#these two lines of code create the characters in\_game sprite

character\_image=pyglet.resource.image("sprite\_alpha.png")

character=pyglet.sprite.Sprite(img=character\_image,

x=spawnx,y=spawny)

interactable\_image=pyglet.resource.image("interactable\_alpha.png")

interactable=pyglet.sprite.Sprite(img=interactable\_image,

x=world.width//3, y=world.height//2)

#This is going to be the menu button

menu\_button=pyglet.image.load("menu.png")

#this line of code is going to define when the switch to turn based combat happens

#This value is a constant, so it wont change

encounter=5

#These lines of code clears the screen and puts the specified objects onto

#the screen

@world.event

def on\_draw():

world.clear()

character.draw()

interactable.draw()

menu\_button.blit(x=world.width-100, y=world.height-50)

#These lines of code work with the button inputs. When a specific key on the keyboard has

#been pressed, a specific action will be carried out (if any)

@world.event

def on\_key\_press(symbol, modifier):

#all proceeding lines of code will deal with character movement based on specific keyboard

#inputs and work with the boundries of the sprite.

if symbol==key.A or symbol==key.LEFT:

if character.x!=0:

character.x-=20

#This is the random number that defines weather a fight is going to

#happen

fight=random.randrange(1,20)

#This function triggers the turn based combat section

if fight==encounter:

Main\_game.combat(Main\_game)

else:

world.set\_visible(False)

Main\_game.loading(Main\_game,2)

#This line of code (which is the same line of code on all of the other

#3 following input functions) deals with the conversion of the free roam

#area to the shop/hospital area

#

#if character.y==interactable.y and character.x==interactable.x-4:

# Main\_game.shop1(Main\_game)

if character.y<531 and character.y>400:

if character.x>580 and character.x<620:

Main\_game.shop1(Main\_game)

if symbol==key.S or symbol==key.DOWN:

if character.y!=0:

character.y-=20

#This is the random number that defines weather a fight is going to

#happen

fight=random.randrange(1,20)

#This function triggers the turn based combat section

if fight==encounter:

Main\_game.combat(Main\_game)

#if character.y==interactable.y and character.x==interactable.x-4:

# Main\_game.shop1(Main\_game)

if character.x>610 and character.x<512:

if character.y<540 and character.y>520:

character.y+=20

if symbol==key.D or symbol==key.RIGHT:

if character.x!=int(world.width)-20:

character.x+=20

#This is the random number that defines weather a fight is going to

#happen

fight=random.randrange(1,20)

#This function triggers the turn based combat section

if fight==encounter:

Main\_game.combat(Main\_game)

#if character.y==interactable.y and character.x==interactable.x-4:

# Main\_game.shop1(Main\_game)

if character.y<532 and character.y>434:

if character.x>500 and character.x<520:

character.x-=20

if symbol==key.W or symbol==key.UP:

if character.y!=int(world.height)-20:

character.y+=20

#This is the random number that defines weather a fight is going to

#happen

fight=random.randrange(1,20)

#This function triggers the turn based combat section

if fight==encounter:

Main\_game.combat(Main\_game)

#if character.y==interactable.y and character.x==interactable.x-4:

# Main\_game.shop1(Main\_game)

if character.x>511 and character.x<611:

if character.y>520 and character.y<540:

character.y-=20

#These lines of code close the window after making it invisible to the user (temporary

#emergency exit from the program

if symbol==key.ESCAPE:

world.set\_visible(False)

quit()

#this is going to handle the inputinto the pause menu located on the top right of the screen

@world.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

if x>1436 and y<864:

if x<1533 and y<864:

if x<1533 and y>816:

if x>1436 and y>816:

Main\_game.pause\_menu(Main\_game)

print ("x:"+str(x)+" y:"+str(y))

#These lines of code make the screen visible to the user and runs the pyglet code

world.set\_visible(True)

pyglet.app.run()

#--------------------------------------map 2------------------------------------

#This is going to be the meathod that contains the code for the 2nd map

def map2(self,spawny):

#This line of code creates the screen that the player can see

world=pyglet.window.Window(fullscreen=True,visible=True)

#These line of code create the character sprite

character\_image=pyglet.resource.image("sprite\_alpha.png")

character=pyglet.sprite.Sprite(img=character\_image,x=1436,y=int(spawny))

interactable\_image=pyglet.resource.image("interactable\_alpha.png")

interactable=pyglet.sprite.Sprite(img=interactable\_image,

x=20, y=20)

#This is going to be the menu button

menu\_button=pyglet.image.load("menu.png")

#These lines of code place stuff onto the screen

@world.event

def on\_draw():

world.clear()

character.draw()

menu\_button.blit(x=world.width-100, y=world.height-50)

interactable.draw()

#These lines of code work with the different keyboard inputs

@world.event

def on\_key\_press(symbol,modifier):

if symbol==key.A or symbol==key.LEFT:

if character.x!=0:

character.x-=20

if symbol==key.D or symbol==key.RIGHT:

if character.x!=1436:

character.x+=20

else:

world.set\_visible(False)

Main\_game.map1(Main\_game,int(character.y),0)

if symbol==key.W or symbol==key.UP:

if character.y!=int(world.height):

character.y+=20

if symbol==key.S or symbol==key.DOWN:

if character.y!=0:

character.y-=20

if character.y==20:

world.set\_visible(False)

Main\_game.loading(Main\_game,4)

#this is going to handle the inputinto the pause menu located on the top right of the screen

@world.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

if x>1436 and y<864:

if x<1533 and y<864:

if x<1533 and y>816:

if x>1436 and y>816:

Main\_game.pause\_menu(Main\_game)

print ("x:"+str(x)+" y:"+str(y))

#world.set\_visible(True)

pyglet.app.run()

#------------------------------------------map 3---------------------------------------------

def map3(self,spawnx):

world=pyglet.window.Window(fullscreen=True)

character\_image=pyglet.resource.image("sprite\_alpha.png")

character=pyglet.sprite.Sprite(img=character\_image,x=spawnx,y=world.height-40)

interactable\_image=pyglet.resource.image("interactable\_alpha.png")

interactable=pyglet.sprite.Sprite(img=interactable\_image,

x=world.width-100, y=40)

menu\_button=pyglet.image.load("menu.png")

@world.event

def on\_draw():

world.clear()

character.draw()

menu\_button.blit(x=world.width-100, y=world.height-50)

interactable.draw()

@world.event

def on\_key\_press(symbol,modifier):

if symbol==key.A or symbol==key.LEFT:

if character.x!=0:

character.x-=20

if symbol==key.D or symbol==key.RIGHT:

if character.x!=1436:

character.x+=20

else:

world.set\_visible(False)

Main\_game.map1(Main\_game,int(character.y),0)

if symbol==key.W or symbol==key.UP:

if character.y!=int(world.height):

character.y+=20

if symbol==key.S or symbol==key.DOWN:

if character.y!=0:

character.y-=20

if character.x>1430 and character.x<1530:

if character.y>125 and character.y<155:

character.y-=20

Main\_game.shop1(Main\_game)

@world.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

if x>1436 and y<864:

if x<1533 and y<864:

if x<1533 and y>816:

if x>1436 and y>816:

Main\_game.pause\_menu(Main\_game)

print ("x:"+str(x)+" y:"+str(y))

#---------------------------------------pause menu-------------------------------------------

def pause\_menu(self):

menu=pyglet.window.Window(fullscreen=True,resizable=True)

quit\_button=pyglet.image.load("quit\_button.png")

@menu.event

def on\_draw():

menu.clear()

quit\_button.blit(x=0, y=menu.height-300)

#This code is going to handle all of the mouse events within the program

@menu.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

#This code is so if the user clicks outside of the window, the

#menu will close

#

#if x<0 or x>menu.width:

# menu.set\_visible(False)

#if y<0 or y>menu.height:

# menu.set\_visible(False)

#This code is creating a box like input for the quit button

if x>0 and y<765:

if x<498 and y<765:

if x<498 and y>568:

if x>0 and y>568:

#This sets the screen to an inactive state

menu.set\_visible(False)

#This is commented out because while I am using the code

#in an IDE, I need to set all of the windows to inactive,

#I dont know how to set the world to inactive based on a

#compleatley 'unrelated' event, so for now, I am just going

#to set the command to make the pause menu to inactive

#

#quit()

pyglet.app.run()

#---------------------------------------shop-------------------------------------------------

#This meathod will define a shop instance

def shop1(self):

#This creates the window

shop=pyglet.window.Window(fullscreen=True,visible=False)

#These lines of code build the sprites

buy\_image=pyglet.resource.image("buy\_button.png")

buy\_image\_2=pyglet.resource.image("buy\_button\_2.png")

buy\_button=pyglet.sprite.Sprite(img=buy\_image,x=20,y=shop.height-300)

sell\_image=pyglet.resource.image("sell\_button.png")

sell\_image\_2=pyglet.resource.image("sell\_button\_2.png")

sell\_button=pyglet.sprite.Sprite(img=sell\_image,x=20,y=shop.height-600)

exit\_image=pyglet.resource.image("exit\_button.png")

exit\_image\_2=pyglet.resource.image("exit\_button\_2.png")

exit\_button=pyglet.sprite.Sprite(img=exit\_image,x=20,y=shop.height-900)

#This function puts the objectsonto the screen

@shop.event

def on\_draw():

buy\_button.draw()

sell\_button.draw()

exit\_button.draw()

#This function handle all events triggered by a mouse click

@shop.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

#This creates a box type input

#------------buy-----------

if x>19 and y<864:

if x<1019 and y<864:

if x<1019 and y>554:

if x>19 and y>554:

print ("BUY")

buy\_button.image=buy\_image\_2

#-------------sell----------

if x>20 and y<513:

if x<1018 and y<513:

if x<1018 and y>266:

if x>20 and y>266:

print ("SELL")

sell\_button.image=sell\_image\_2

#----------exit-------------

if x>19 and y<266:

if x<1018 and y<266:

if x>19 and y>1:

if x>19 and y>1:

exit\_button.image=exit\_image\_2

shop.set\_visible(False)

#This finction is triggered every time the user releases the mouse button they pressed

@shop.event

def on\_mouse\_release(x,y,button,modifier):

if button==mouse.LEFT:

#Box type input

if x>19 and y<864:

if x<1019 and y<864:

if x<1019 and y>554:

if x>19 and y>554:

#This line of code changes the image of the sprite

buy\_button.image=buy\_image

if x>20 and y<513:

if x<1018 and y<513:

if x<1018 and y>266:

if x>20 and y>266:

sell\_button.image=sell\_image

if x>19 and y<266:

if x<1018 and y<266:

if x>19 and y>1:

if x>19 and y>1:

exit\_button.image=exit\_image

shop.set\_visible(True)

pyglet.app.run()

#--------------------------------------turn based combat section----------------------------

def combat(Self):

#This code creates the screen and makes it invisible

screen=pyglet.window.Window(visible=False,fullscreen=True)

#This line of code creates a text label written attack

attack\_text=pyglet.text.Label("ATTACK",

x=screen.width//3-300, y=screen.height//4,

font\_size=80)

recruit\_text=pyglet.text.Label("RECRUIT",

x=screen.width//3-330, y=screen.height//4-190,

font\_size=80)

item\_text=pyglet.text.Label("ITEM",

x=screen.width//3+350,y=screen.height//4,

font\_size=80)

run\_text=pyglet.text.Label("RUN",

x=screen.width//3+350,y=screen.height//4-190

,font\_size=80)

#These lines of code place the text onto the screen for the user to see

@screen.event

def on\_draw():

screen.clear()

attack\_text.draw()

recruit\_text.draw()

item\_text.draw()

run\_text.draw()

#This code draws a atraight line across the screen

pyglet.graphics.draw(2,pyglet.gl.GL\_LINES,(

"v2i",(2,screen.height//2-100,screen.width,

screen.height//2-100)))

pyglet.graphics.draw(2,pyglet.gl.GL\_LINES,(

"v2i",(2,screen.height//2-280,screen.width,

screen.height//2-280)))

pyglet.graphics.draw(2,pyglet.gl.GL\_LINES,(

"v2i",(screen.width//2,0,screen.width//2,screen.height)))

#These lines of code dea with specific keyboard inputs

#@screen.event

#def on\_key\_press(symbol,modifier):

# if symbol==key.ESCAPE:

# screen.set\_visible(False)

#these lines of code will handle the mouse inputs for the turn-based part of

#the program

#the relevant help was gained from

#https://pyglet.readthedocs.io/en/pyglet-1.2-maintenance/programming\_guide/mouse.html

@screen.event

def on\_mouse\_press(x,y,button,modifier):

if button==mouse.LEFT:

#attack

if x>=0 and y<=329:

if x<=762 and y<=329:

if x<=762 and y>=161:

if x>=0 and y>=161:

print ("ATTACK")

#item

if x>=769 and y<=329:

if x<=1533 and y<=329:

if x<=1533 and y>=161:

if x>=769 and y>=161:

print ("ITEM")

#recruit

if x>=0 and y<=154:

if x<=762 and y<=154:

if x<= 762 and y>=6:

if x>=0 and y>=6:

print ("RECRUIT")

#run

if x>=769 and y<=154:

if x<=1533 and y<=154:

if x<=1533 and y>=6:

if x>=769 and y>=6:

screen.set\_visible(False)

#This line of code makes the screen visible to the user

screen.set\_visible(True)

#This allows the code to be exicuted and looped

pyglet.app.run()

#----------------------------------------------------menus----------------------------------

class Menu(Frame):

def \_\_init\_\_(self,master=None):

Frame.\_\_init\_\_(self,master)

self.master=master

self.main\_menu()

def main\_menu(self):

self.master.title("Main menu")

self.pack(fill=BOTH,expand=1)

start\_button=Button(self,text="Start game",command=self.start\_game)

start\_button.place(x=150,y=150)

settings\_button=Button(self,text="Setting", command=self.settings)

settings\_button.place(x=150,y=180)

def start\_game(self):

root.destroy()

Main\_game()

def settings(self):

quit()

#----------------------------------------------configuration--------------------------------

root=Tk()

root.geometry("400x400")

app=Menu(root)

root.mainloop()