#########################################

# File Name: ToonyWars.py

# Description: Game

# Author: Jonathan Yam, Leo

# Date: 11/29/2018

#########################################

import pygame

import math

import random

pi = math.pi

pygame.init()

WIDTH = 1200

HEIGHT = 700

TOP = 0

BOTTOM = 500

gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))

BLACK = (0, 0, 0)

RED = (255, 0, 0)

YELLOW = (255, 255, 0)

ORANGE = (255, 128, 64)

GREEN = ( 0, 128, 0)

BLUE = (0, 255, 255)

WHITE = (255, 255, 255)

GREY = (72, 72, 72)

BROWN = (185, 122, 87)

outline = 0

gameDirectory = "images"

clock = pygame.time.Clock()

windowRealitiveX = 0

mapRealitiveX = 0

frame = 0

menuFrame = 0

#variables to keep which keys are pressed

key1 = False

key2 = False

key3 = False

#player 1 lists for the values and states of troops

player1TroopNumList = []

player1TroopTypeList = []

player1TroopStateList = []

player1TroopXList = []

player1TroopYList = []

player1TroopHealthList = []

#player 2 lists for the values and states of troops

player2TroopNumList = []

player2TroopTypeList = []

player2TroopStateList = []

player2TroopXList = []

player2TroopYList = []

player2TroopHealthList = []

#lists to keep track of the animations player 1

swordManPic = []

fireDragonPic = []

bombManPic = []

#lists to keep track of the animations player 2

player2SwordManPic = []

player2FireDragonPic = []

player2BombManPic = []

#Tower Properties

tower = "tower"

player1TowerPic = pygame.image.load("tower1.png").convert\_alpha()

player2TowerPic = pygame.image.load("tower2.png").convert\_alpha()

player1TowerX = 20 + mapRealitiveX

player1TowerY = BOTTOM-300

player2TowerX = 2900 + mapRealitiveX

player2TowerY = BOTTOM - 300

player1TowerHP = 1000000

player2TowerHP = 1000000

#player 1 troop properties

#Sword Man Properties

swordMan = "swordMan"

numOfSwordManPics = 6

for i in range(numOfSwordManPics):

swordManPic.append(pygame.image.load("swordman"+str(i)+".png").convert\_alpha())

swordManPicNum = 0

swordManX = 100 + mapRealitiveX

swordManY = 350

swordManSpeedX = 2

swordManHP = 10000

swordManW = 130

swordManH = 130

player1SwordmanRange = -50

player1SwordmanDamage = 200

#bomb man properties

bombMan = "bombMan"

numOfBombManPics = 6

for i in range(numOfBombManPics):

bombManPic.append(pygame.image.load("bombman"+str(i)+".png").convert\_alpha())

bombManPicNum = 0

bombManX = 100 + mapRealitiveX

bombManY = 350

bombManSpeedX = 2

bombManHP = 5000

bombManW = 130

bombManH = 130

player1BombManRange = -300

player1BombManDamage = 500

#dragon Properties

dragon = "dragon"

numOfFireDragonPics = 6

for i in range(numOfFireDragonPics):

fireDragonPic.append(pygame.image.load("dragon"+str(i)+".png").convert\_alpha())

fireDragonPicNum = 0

fireDragonX = 100 + mapRealitiveX

fireDragonY = 250

fireDragonSpeedX = 3

fireDragonHP = 30000

fireDragonW = 130

fireDragonH = 130

player1DragonRange = -200

player1DragonDamage = 1000

#player 2 troop properties

#swordman properties

swordMan = "swordMan"

numOfSwordManPics = 6

for i in range(numOfSwordManPics):

player2SwordManPic.append(pygame.image.load("player2swordman"+str(i)+".png").convert\_alpha())

swordManPicNum = 0

player2SwordManX = 2900 + mapRealitiveX

swordManY = 350

swordManSpeedX = 2

swordManHP = 10000

swordManW = 130

swordManH = 130

player2SwordmanRange = 50

player2SwordmanDamage = 200

#bomb man properties

bombMan = "bombMan"

numOfBombManPics = 6

for i in range(numOfBombManPics):

player2BombManPic.append(pygame.image.load("player2bombman"+str(i)+".png").convert\_alpha())

bombManPicNum = 0

player2BombManX = 2900 + mapRealitiveX

bombManY = 350

bombManSpeedX = 2

bombManHP = 5000

bombManW = 130

bombManH = 130

player2BombManRange = 300

player2BombManDamage = 500

#dragon Properties

dragon = "dragon"

numOfFireDragonPics = 6

for i in range(numOfFireDragonPics):

player2FireDragonPic.append(pygame.image.load("player2dragon"+str(i)+".png").convert\_alpha())

fireDragonPicNum = 0

player2FireDragonX = 2900 + mapRealitiveX

fireDragonY = 250

fireDragonSpeedX = 3

fireDragonHP = 30000

fireDragonW = 130

fireDragonH = 130

player2DragonRange = 200

player2DragonDamage = 1000

background = pygame.image.load("background.jpg").convert\_alpha()

mainMenuScreen= pygame.image.load("main menu.png").convert\_alpha()

player1UI = pygame.image.load("player1UI.png").convert\_alpha()

player2UI = pygame.image.load("player2UI.png").convert\_alpha()

font1 = pygame.font.SysFont("ALGERIAN",10)

font2 = pygame.font.SysFont("ALGERIAN",20)

player1Mana = 100

player2Mana = 100

backgroundX = 0

backgroundY = 0

letterX = 320

letterY = 240

backgroundXY = background.get\_rect()

player1Tower = True

player2Tower = True

#money variables

player1Money = 0

player1StartingMoney = 1000

player1MoneyIncrease = 100

player2Money = 0

player2StartingMoney = 1000

player2MoneyIncrease = 100

#flag to close program when hit escape key

inPlay = True

gameOver = False

#variables to keep track of screen

menu = True

singlePlayer = False

multiplayer = False

#function for mouse clicking buttons in game

def mousePosition(x1, x2, y1, y2):

if event.type == pygame.MOUSEBUTTONDOWN:

print "mouse", mouseX, mouseY

if mouseX > x1 and mouseX < x2 and mouseY > y1 and mouseY < y2:

return True

else:

return False

#function for grid

def grid():

for x in range(0, WIDTH, 10):

pygame.draw.line(gameWindow, BLUE, (x, 0), (x, HEIGHT), 1)

for y in range(0, HEIGHT, 10):

pygame.draw.line(gameWindow, BLUE, (0, y), (WIDTH, y), 1)

for x in range(0, WIDTH, 10 \* 10):

pygame.draw.line(gameWindow, GREY, (x, 0), (x, HEIGHT), 2)

for y in range(0, HEIGHT, 10 \* 10):

pygame.draw.line(gameWindow, GREY, (0, y), (WIDTH, y), 2)

#function for main menu

def mainMenu():

gameWindow.fill(WHITE)

gameWindow.blit(mainMenuScreen , (0, 0))

#function for multiplayer map

def redrawGameWindow():

gameWindow.fill(WHITE)

gameWindow.blit(background, ( mapRealitiveX, - 140))

#function to draw UI elements

def drawPlayer1UI(mana):

gameWindow.blit(player1UI, (0, 500))

manaHeight = mana \* 1.4

manaY = 140 - manaHeight

pygame.draw.rect(gameWindow, BLUE, (500, 530 + manaY, 70, manaHeight), 0)

def drawPlayer2UI(mana):

gameWindow.blit(player2UI, (600, 500))

manaHeight = mana \* 1.4

manaY = 140 - manaHeight

pygame.draw.rect(gameWindow, BLUE, (1100, 530 + manaY, 70, manaHeight), 0)

#function to draw player 1 tower

def drawAllyTower():

if any("tower" in s for s in player1TroopTypeList):

if player1TroopHealthList[player1TroopTypeList.index("tower")] > 0:

gameWindow.blit(player1TowerPic, (player1TowerX + mapRealitiveX, player1TowerY))

letterGraphics = font2.render("HP: " +str(player1TroopHealthList[player1TroopTypeList.index("tower")]),1,BLACK)

gameWindow.blit(letterGraphics, (player1TowerX + mapRealitiveX, player1TowerY))

#function to draw layer 2 tower

def drawEnemyTower():

if any("tower" in s for s in player2TroopTypeList):

if player2TroopHealthList[player2TroopTypeList.index("tower")] > 0 :

gameWindow.blit(player2TowerPic, (player2TowerX + mapRealitiveX, player2TowerY))

letterGraphics = font2.render("HP: " +str(player2TroopHealthList[player2TroopTypeList.index("tower")]),1,BLACK)

gameWindow.blit(letterGraphics, (player2TowerX + mapRealitiveX, player2TowerY))

#function for deploying different types of troops for player 1

def player1DeployTroop(keys, troopType, troopState, troopX, troopY, troopHealth):

if keys == True:

player1TroopTypeList.append(troopType)

player1TroopStateList.append(troopState)

player1TroopXList.append(troopX)

player1TroopYList.append(troopY)

player1TroopHealthList.append(troopHealth)

#function for deploying different types of troops for player 2

def player2DeployTroop(keys, troopType, troopState, troopX, troopY, troopHealth):

if keys == True:

player2TroopTypeList.append(troopType)

player2TroopStateList.append(troopState)

player2TroopXList.append(troopX)

player2TroopYList.append(troopY)

player2TroopHealthList.append(troopHealth)

#function for annimating and changing the state of troops

def player1AnnimateAndMoveTroops(troopType, troopRange, troopSpeed, troopPic, troopY):

#if troop is certain type and has health

if player1TroopTypeList[player1NumTroop] == troopType and player1TroopHealthList[player1NumTroop] > 0:

# changes state to moving when there are no enemies in front

player1TroopStateList[player1NumTroop] = "moving"

# if there are enemy troops other than enemy tower

if len(player2TroopTypeList) > 1:

# change state to attacking when enemy in front

if player1TroopXList[player1NumTroop] >= player2FirstTroop + troopRange:

player1TroopStateList[player1NumTroop] = "attacking"

# if there is only enemy tower

elif len(player2TroopTypeList) == 1:

# changes state to attacking when tower in front

if player1TroopXList[player1NumTroop] >= player2FirstTroop + troopRange:

player1TroopStateList[player1NumTroop] = "attacking"

if player1TroopStateList[player1NumTroop] == "moving":

#moving annimation

player1TroopXList[player1NumTroop] += troopSpeed

gameWindow.blit(troopPic[frame% 3 + 3], (player1TroopXList[player1NumTroop] + mapRealitiveX, player1TroopYList[player1NumTroop]))

letterGraphics = font1.render("HP: " + str(player1TroopHealthList[player1NumTroop]), 1, BLACK)

gameWindow.blit(letterGraphics, (player1TroopXList[player1NumTroop] + mapRealitiveX, troopY))

elif player1TroopStateList[player1NumTroop] == "attacking" and len(player2TroopHealthList) > 0:

#attacking animation

player2TroopHealthList[player2FirstTroopIndex] -= player1SwordmanDamage

gameWindow.blit(troopPic[frame % 3 ], (player1TroopXList[player1NumTroop] + mapRealitiveX, player1TroopYList[player1NumTroop]))

letterGraphics = font1.render("HP: " + str(player1TroopHealthList[player1NumTroop]), 1, BLACK)

gameWindow.blit(letterGraphics, (player1TroopXList[player1NumTroop] + mapRealitiveX, troopY))

def player2AnnimateAndMoveTroops(troopType, troopRange, troopSpeed, troopPic, troopY):

if player2TroopTypeList[player2NumTroop] == troopType and player2TroopHealthList[player2NumTroop] > 0:

# changes state to moving when there are no enemies in front

player2TroopStateList[player2NumTroop] = "moving"

#if there are enemy troops other than enemy tower

if len(player1TroopTypeList) > 1:

# change state to attacking when enemy in front

if player2TroopXList[player2NumTroop] <= player1FirstTroop + troopRange:

player2TroopStateList[player2NumTroop] = "attacking"

#if there is only enemy tower

elif len(player1TroopTypeList) == 1:

# changes state to attacking when tower in front

if player2TroopXList[player2NumTroop] <= player1FirstTroop + troopRange:

player2TroopStateList[player2NumTroop] = "attacking"

if player2TroopStateList[player2NumTroop] == "moving":

# moving annimation

player2TroopXList[player2NumTroop] -= troopSpeed

gameWindow.blit(troopPic[frame % 3 + 3],

(player2TroopXList[player2NumTroop] + mapRealitiveX, player2TroopYList[player2NumTroop]))

letterGraphics = font1.render("HP: " + str(player2TroopHealthList[player2NumTroop]), 1, BLACK)

gameWindow.blit(letterGraphics, (player2TroopXList[player2NumTroop] + mapRealitiveX, troopY))

elif player2TroopStateList[player2NumTroop] == "attacking" and len(player1TroopHealthList) > 0:

# attacking animation

player1TroopHealthList[player1FirstTroopIndex] -= player2SwordmanDamage

gameWindow.blit(troopPic[frame % 3],

(player2TroopXList[player2NumTroop] + mapRealitiveX, player2TroopYList[player2NumTroop]))

letterGraphics = font1.render("HP: " + str(player2TroopHealthList[player2NumTroop]), 1, BLACK)

gameWindow.blit(letterGraphics, (player2TroopXList[player2NumTroop] + mapRealitiveX, troopY))

key1 = False

key2 = False

key3 = False

key4 = False

key7 = False

key8 = False

key9 = False

key0 = False

#def runGame():

#return

#loops whole program

while inPlay:

while menu == True:

mainMenu()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_ESCAPE:

inPlay = False

menu = False

multiplayer = False

if event.key == pygame.K\_m:

menu = False

multiplayer = True

mouseX = pygame.mouse.get\_pos()[0]

mouseY = pygame.mouse.get\_pos()[1]

if event.type == pygame.MOUSEBUTTONDOWN:

if mouseX > 480 and mouseX < 740 and mouseY > 300 and mouseY < 390 :

menu = False

singlePlayer = True

if event.type == pygame.MOUSEBUTTONDOWN:

if mouseX > 450 and mouseX < 760 and mouseY > 400 and mouseY < 490:

menu = False

multiplayer = True

#grid()

clock.tick(60)

print clock.get\_fps()

pygame.display.update()

while multiplayer == True:

# reset the keys prerssed

key1 = False

key2 = False

key3 = False

key4 = False

key7 = False

key8 = False

key9 = False

key0 = False

redrawGameWindow()

drawPlayer1UI(player1Mana)

drawPlayer2UI(player2Mana)

#get mouse inputs

for event in pygame.event.get():

mouseX = pygame.mouse.get\_pos()[0]

mouseY = pygame.mouse.get\_pos()[1]

#mouse position for moving the screen

mousePosition(0, 100, 0, 500)

mousePosition(1100, 1200, 0, 500)

#click buttons to deploy player 1 troops

mousePosition(30, 110, 530, 590)

mousePosition(135, 215, 530, 590)

mousePosition(30, 110, 610, 670)

mousePosition(135, 215, 530, 590)

#player 2 buttons

mousePosition(630, 710, 530, 590)

mousePosition(735, 815, 530, 590)

mousePosition(630, 710, 610, 670)

mousePosition(735, 815, 530, 590)

#get keyboard inputs

pygame.event.get()

keys = pygame.key.get\_pressed()

#moves the map

if keys[pygame.K\_RIGHT] and mapRealitiveX > - backgroundXY[2] + WIDTH :

mapRealitiveX -= 20

elif mousePosition(1100, 1200, 0, 500) and mapRealitiveX > - backgroundXY[2] + WIDTH :

mapRealitiveX -= 20

if keys[pygame.K\_LEFT] and mapRealitiveX < 0 :

mapRealitiveX += 20

elif mousePosition(0, 100, 0, 500) == True and mapRealitiveX < 0:

mapRealitiveX += 20

#close game when press escape

if keys[pygame.K\_ESCAPE]:

inPlay = False

menu = False

multiplayer = False

#player 1 press keys 1,2,3 to summon troop

if keys[pygame.K\_1] or mousePosition(30, 110, 530, 590) == True:

if player1Mana >= 5:

player1Mana = player1Mana - 5

key1 = True

if keys[pygame.K\_2] or mousePosition(135, 215, 530, 590) == True:

if player1Mana >= 5:

player1Mana = player1Mana - 5

key2 = True

if keys[pygame.K\_3] or mousePosition(30, 110, 610, 670) == True:

if player1Mana >= 5:

player1Mana = player1Mana - 5

key3 = True

if keys[pygame.K\_4] or mousePosition(135, 215, 530, 590) == True:

key4 = True

#player 2 press keys 7,8,9 to summon troop

if keys[pygame.K\_7] or mousePosition(630, 710, 530, 590) == True:

if player2Mana >= 5:

player2Mana = player2Mana - 5

key7 = True

if keys[pygame.K\_8] or mousePosition(735, 815, 530, 590) == True:

if player2Mana >= 5:

player2Mana = player2Mana - 5

key8 = True

if keys[pygame.K\_9] or mousePosition(630, 710, 610, 670) == True:

if player2Mana >= 5:

player2Mana = player2Mana - 5

key9 = True

if keys[pygame.K\_0] or mousePosition(735, 815, 530, 590) == True:

key0 = True

#create Towers

player1DeployTroop(player1Tower, tower, "null", player1TowerX, player1TowerY, player1TowerHP)

player1Tower = False

player2DeployTroop(player2Tower, tower, "null", player2TowerX, player2TowerY, player2TowerHP)

player2Tower = False

#player 1 deploy troops

#deploy swordman

player1DeployTroop(key1, swordMan, "moving", swordManX, swordManY, swordManHP)

#deploy bombman

player1DeployTroop(key2, bombMan, "moving", bombManX, bombManY, bombManHP)

#deploy dragon

player1DeployTroop(key3, dragon, "moving", fireDragonX, fireDragonY, fireDragonHP)

#player 2 deploy troops

#deploy swordman

player2DeployTroop(key7, swordMan, "moving", player2SwordManX, swordManY, swordManHP)

#deploy bombman

player2DeployTroop(key8, bombMan, "moving", player2BombManX, bombManY, bombManHP)

#deploy dragon

player2DeployTroop(key9, dragon, "moving", player2FireDragonX, fireDragonY, fireDragonHP)

#mark closest player 1 troop to be targeted

if len(player1TroopTypeList)> 0 :

player1FirstTroop = max(player1TroopXList)

player1FirstTroopIndex = player1TroopXList.index(max(player1TroopXList))

#mark closest player 2 troop to be targeted

if len(player2TroopTypeList)> 0 :

player2FirstTroop = min(player2TroopXList)

player2FirstTroopIndex = player2TroopXList.index(min(player2TroopXList))

drawAllyTower()

drawEnemyTower()

#cycles through list of troops for player 1

for player1NumTroop in range(len(player1TroopTypeList)):

#sword man

player1AnnimateAndMoveTroops("swordMan", player1SwordmanRange, swordManSpeedX, swordManPic, swordManY)

#bomb man

player1AnnimateAndMoveTroops("bombMan", player1BombManRange, bombManSpeedX, bombManPic, bombManY)

#dragon

player1AnnimateAndMoveTroops("dragon", player1DragonRange, fireDragonSpeedX, fireDragonPic, fireDragonY)

#cycles through list of troops for player 2

for player2NumTroop in range(len(player2TroopTypeList)):

#swordman

player2AnnimateAndMoveTroops("swordMan", player2SwordmanRange, swordManSpeedX, player2SwordManPic, swordManY)

#bomb man

player2AnnimateAndMoveTroops("bombMan", player2BombManRange, bombManSpeedX, player2BombManPic, bombManY)

#dragon

player2AnnimateAndMoveTroops("dragon", player2DragonRange, fireDragonSpeedX, player2FireDragonPic, fireDragonY)

#remove troops after they go off the map to optimise performence

try:

if player1TroopXList[0] >= backgroundXY[2]:

del player1TroopTypeList[0]

del player1TroopStateList[0]

del player1TroopXList[0]

del player1TroopYList[0]

del player1TroopHealthList[0]

except:

pass

try:

if player2TroopXList[0] <= backgroundXY[0] :

del player2TroopTypeList [0]

del player2TroopStateList [0]

del player2TroopXList [0]

del player2TroopYList [0]

del player2TroopHealthList [0]

except:

pass

#need to remove troops from list once their hp < 0

try:

if player1TroopHealthList[player1FirstTroopIndex] < 0:

del player1TroopTypeList[player1FirstTroopIndex]

del player1TroopStateList[player1FirstTroopIndex]

del player1TroopXList[player1FirstTroopIndex]

del player1TroopYList[player1FirstTroopIndex]

del player1TroopHealthList[player1FirstTroopIndex]

except:

pass

try:

if player2TroopHealthList[player2FirstTroopIndex] < 0:

del player2TroopTypeList[player2FirstTroopIndex]

del player2TroopStateList[player2FirstTroopIndex]

del player2TroopXList[player2FirstTroopIndex]

del player2TroopYList[player2FirstTroopIndex]

del player2TroopHealthList[player2FirstTroopIndex]

except:

pass

#grid()

if player1Mana < 100:

player1Mana = player1Mana + 0.05

if player2Mana < 100:

player2Mana = player2Mana + 0.05

if menuFrame % 3 == 0:

frame += 1

menuFrame += 1

#print clock.get\_fps()

clock.tick(30)

pygame.display.update()

pygame.quit()

"""

important shit to do

- make everything on screen realitive to background x and y

- make all characters in a list

- loop through list, if troop state = false/dead, remove troop from list

- use map.get\_rect() to get dimensions of the map

- make towers and troop spawn at eg. left side of map + 100, right side of map - 100

- 1 list for troops

- make function with different prarmeters for deploying troops

- troops need to scan for enemies in front

- if enemies in front, state change from moveing to attacking

- create 1 function for deploying troops

- create 1 finction for animating troops

- create 1 function for everything

- when deployig troops, make y position random/ vaired so some troops in front, some in back.

- CREATE STARTING MENU

- create ui

- game over screen

"""