

PLANTS VS ZOMBIES

COMPETITIVE VERSION

By Shuxuan Liu, Huanci Wang

MOTIVATION:



QUESTION:

- How to make plants vs zombies a competitive game?
- How to bring interaction to this game?

GAME LOGIC:

- Two pea shooters fight against zombies
- They will die if zombies enter the house
- The one with the higher score wins

BASIC CLASS

Player

Draw

Move

Game

Canvas

```
class Player():

    width, height = 50, 50
    FPS = 60

    def __init__(self, startx, starty, color=(255,0,0)):
        self.x = startx
        self.y = starty
        self.velocity = 8
        self.color = color

    def draw(self, g):
        #pygame.draw.rect(g, self.color, (self.x, self.y, self.width, self.height), 0)
        playerImage = pygame.image.load('SnowPea.gif')
        playerRect = playerImage.get_rect()
        #playerRect = playerImage.get_rect()
        playerRect.centerx = self.x
        playerRect.centery = self.y
        g.blit(playerImage, playerRect)

    def move(self, dirn):
        """
        :param dirn: 0 - 3 (right, left, up, down)
        :return: None
        """
        if dirn == 0:
            self.x = self.x + self.velocity
        elif dirn == 1:
            self.x = self.x - self.velocity
        elif dirn == 2:
            self.y = self.y - self.velocity
        else:
            self.y = self.y + self.velocity

class Canvas:

    def __init__(self, w, h, name="None"):
        self.width = w
        self.height = h
        self.screen = pygame.display.set_mode((w,h))
        pygame.display.set_caption(name)

    @staticmethod
    def update():
        pygame.display.update()

    '''def draw_text(self, text, size, x, y):
        pygame.font.init()
        font = pygame.font.SysFont("comicsans", size)
        render = font.render(text, 1, (0,0,0))

        self.screen.draw(render, (x,y))'''

    def grow(self, x, y):
        playerImage = pygame.image.load('SnowPea.gif')
        playerRect = playerImage.get_rect()
        playerRect.centerx = x
        playerRect.centery = y
        self.screen.blit(playerImage, playerRect)

    def get_canvas(self):
        return self.screen

    def draw_background(self):
        #self.screen.fill((255,255,255))
        backgroundImage = pygame.image.load('background.png')
        rescaledBackground = pygame.transform.scale(backgroundImage, (1024, 600))

        # show the "Start" screen
        self.screen.blit(rescaledBackground, (0, 0))
```

```
class Game:

    def __init__(self, w, h):
        self.net = Network()
        self.width = w
        self.height = h
        self.player = Player(200, 180) #initial position of the two players.
        self.player2 = Player(100,100)
        self.canvas = Canvas(self.width, self.height, "Plant VS Zombie")

    def run(self):
        clock = pygame.time.Clock()

        WINDOWWIDTH = 1024
        WINDOWHEIGHT = 600
        ADDNEWBULLETRATE = 10
        ZOMBIESIZE = 80 #includes newKindZombies
        ADDNEWZOMBIERATE = 50
        ADDNEWKINDZOMBIE = ADDNEWZOMBIERATE
        MAXGOTTENPASS = 1
        BULLETSPED = 10
        NORMALZOMBIESPEED = 6
        NEWKINDZOMBIESPEED = NORMALZOMBIESPEED / 2

        TEXTCOLOR = (255, 255, 255)
        RED = (255, 0, 0)

        run = True
        shoot = False
        bullets = []
        zombies = []
        newKindZombies = []
        bulletAddCounter = 40
        zombieAddCounter = 0
        newKindZombieAddCounter = 0

        score = 0
        zombiesGottenPast = 0

        backgroundImage = pygame.image.load('background.png')
        rescaledBackground = pygame.transform.scale(backgroundImage, (1024, 600))

        pygame.init()
        pygame.mixer.music.load('grasswalk.mp3')
        pygame.mixer.music.play(-1)
        gameOverSound = pygame.mixer.Sound('gameover.wav')

        font = pygame.font.SysFont(None, 48)

    def drawText(text, font, surface, x, y):
        textobi = font.render(text, 1, TEXTCOLOR)
```

BASICS

```
backgroundImage = pygame.image.load('background.png')
rescaledBackground = pygame.transform.scale(backgroundImage, (1024, 600))

windowSurface = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
windowSurface.blit(rescaledBackground, (0, 0))
pygame.mouse.set_visible(True)

bulletImage = pygame.image.load('SnowPeashooterBullet.gif')
bulletRect = bulletImage.get_rect()
playerImage = pygame.image.load('SnowPea.gif')
playerRect = playerImage.get_rect()
zombieImage = pygame.image.load('BucketheadZombie.png')
newKindZombieImage = pygame.image.load('ConeheadZombieAttack.gif')
```

STEP 1: SINGLE PLAYER

ZOMBIES APPEAR & MOVE: DICTIONARY AS ZOMBIES

```
zombieAddCounter += 1
if zombieAddCounter == 50:
    zombieAddCounter = 0
    zombieSize = 50
    newZombie = {'rect':
pygame.Rect(1024,random.randint(10,600-zombieSize-10),
zombieSize,zombieSize),'surface':pygame.transform.scale(zomb
ieImage, (zombieSize, zombieSize))}
}
```


ZOMBIES APPEAR & MOVE: DICTIONARY AS ZOMBIES

```
['rect'].move_ip(-1*NORMALZOMBIESPEED, 0)
```

```
self.canvas.screen.blit(z['surface'], z['rect'])
```

```
self.canvas.update()
```

```
}
```

PRESS SPACE TO SHOOT PEA

```
if bulletAddCounter >=
    ADDNEWBULLETRATE and shoot == True:
        ...
        newBullet =
{'rect':pygame.Rect(bullet_x,bullet_
y, bulletRect.width,
bulletRect.height),
'surface':pygame.transform.scale(bul
letImage, (bulletRect.width,
bulletRect.height)}}
bullets.append(newBullet)
```

```
for event in pygame.event.get():
    if event.type == KEYDOWN:
        if event.key == K_SPACE:
            shoot = True
    if event.type == KEYUP:
        if event.key == K_SPACE:
            shoot = False
```

BULLET HITS ZOMBIE

```
if c['rect'].left < 0:

    newKindZombies.remove(c)

    zombiesGottenPast += 1

    if bulletHasHitCrawler(bullets,
newKindZombies):

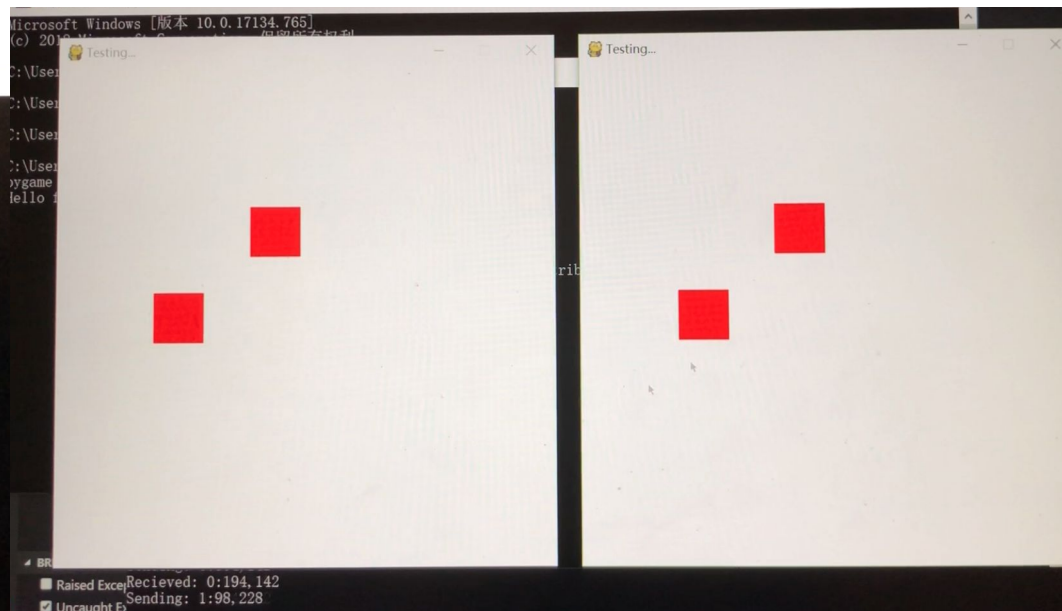
        score += 1

    newKindZombies.remove(c)
```

```
for event in pygame.event.get():
    if event.type == KEYDOWN:
        if event.key == K_SPACE:
            shoot = True
    if event.type == KEYUP:
        if event.key == K_SPACE:
            shoot = False
```

STEP 2: DOUBLE PLAYER

BASICS

[illegible]

UPDATE PLAYER 2 POSITION

```
#send position, receive player2  
position  
  
def send_data(self):  
    data = str(self.net.id) +  
    ":" + str(self.player.x) + "," +  
    str(self.player.y)  
    reply = self.net.send(data)  
    return reply
```

```
def send(self, data):  
    try:  
  
    self.client.send(str.encode(data))  
        reply =  
    self.client.recv(2048).decode()  
        return reply  
    except socket.error as e:  
        return str(e)
```

UPDATE PLAYER 2 POSITION

```
# Update Canvas  
self.canvas.draw_background()  
self.player.draw(self.canvas.get_canvas())  
self.player2.draw(self.canvas.get_canvas())
```

SEE OTHER PLAYER'S SCORE

```
def send_data(self, x,y,s):  
    data = str(self.net.id) + ":" + str(x) + "," + str(y) + ',' +  
    str(s)  
    reply = self.net.send(data)  
    return reply  
  
def parse_data(data):  
    try:  
        d = data.split(":")[1].split(",")  
        return int(d[0]), int(d[1]), int(d[2])
```


SEE OTHER PLAYER'S SCORE

```
drawText('score: %s' % (score), font, self.canvas.screen, 20, 50)
```

```
drawText('opponent score: %s' % (score2),  
font, self.canvas.screen, 20, 100)
```

OTHER FEATURES:

Score

Background music

Beginning page/Gameover

}

BG, BEGINNING PAGE, GAME OVER

```
pygame.init()

pygame.mixer.music.load('grasswalk.m
p3')

pygame.mixer.music.play(-1)

gameOverSound =
pygame.mixer.Sound('gameover.wav')

pygame.mixer.music.stop()

gameOverSound.play()

time.sleep(8)
```

```
def waitForPlayerToPressKey():
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                terminate()
            if event.type == KEYDOWN:
                if event.key == K_ESCAPE: # pressing escape quits
                    terminate()
                if event.key == K_RETURN:
                    return
```



OUR DESIGNS

- Press 'Enter' to start the game
-
- Can see the other user joining and playing
-
- Each plays his/her own game
-
- When the game is over users can still see their scores
-
- Press 'Enter' to escape

THANK YOU

Shuxuan Liu, Huanci Wang