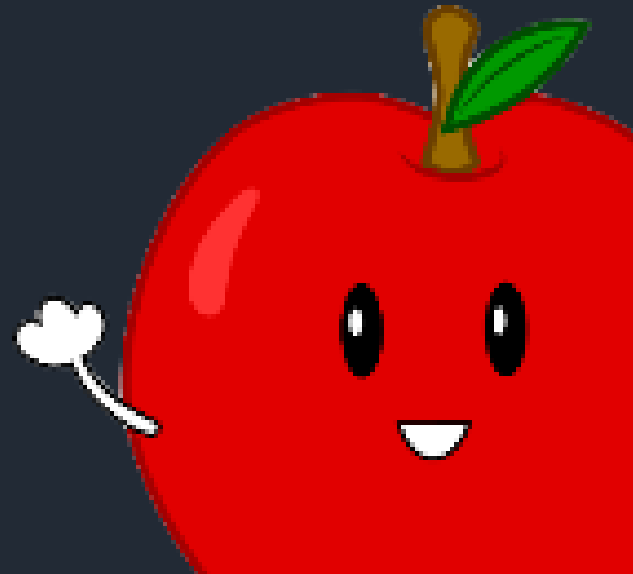


FAT WORM

By the group
WE NEED CACHE

VANESSA HOW 517120990081
KARINA LIANG 517120990057
JINWOO JEONG 517120990035
HYUNTAE JANG 517120990075



Abstract



Inspired by our childhood game,
SNAKE GAME



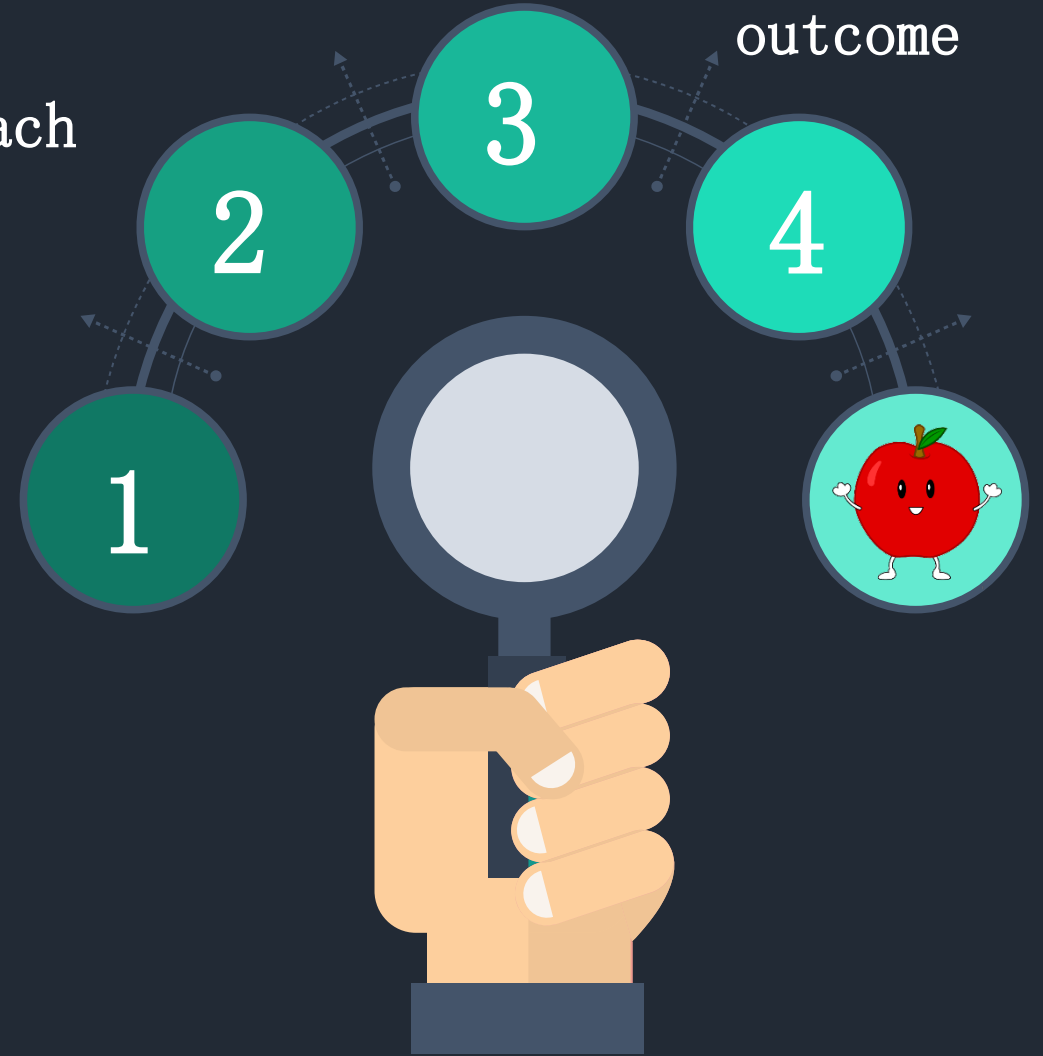
We created
OUR OWN!

problem

approach

coding
process

outcome



Introduction



Our Project “Fat Worm”

Is based on the “Snake Game” .



What is Snake Game?

A video game concept where the player maneuvers a line which grows in length, with the line itself being a primary obstacle.



From Blockade to...

Originated in 1976 from the arcade game Blockade. Ease of implementing Snake resulted in numerous versions.



Nokia’ s mobile phones

After a variant was preloaded on Nokia’ s mobile phones in 1998, many people have become interested . More than 300 snake games in ios!!!

Objective



Embody
the whole “Snake Game”

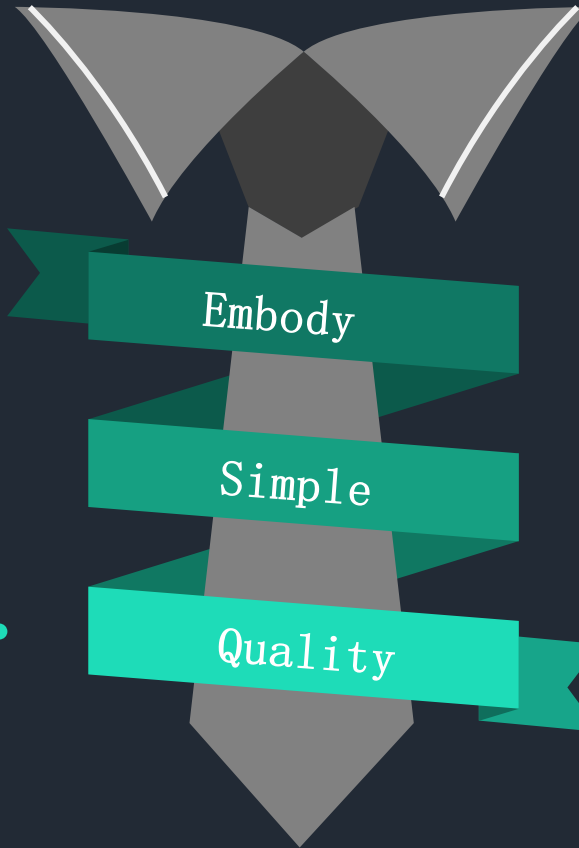
Embody

Simple

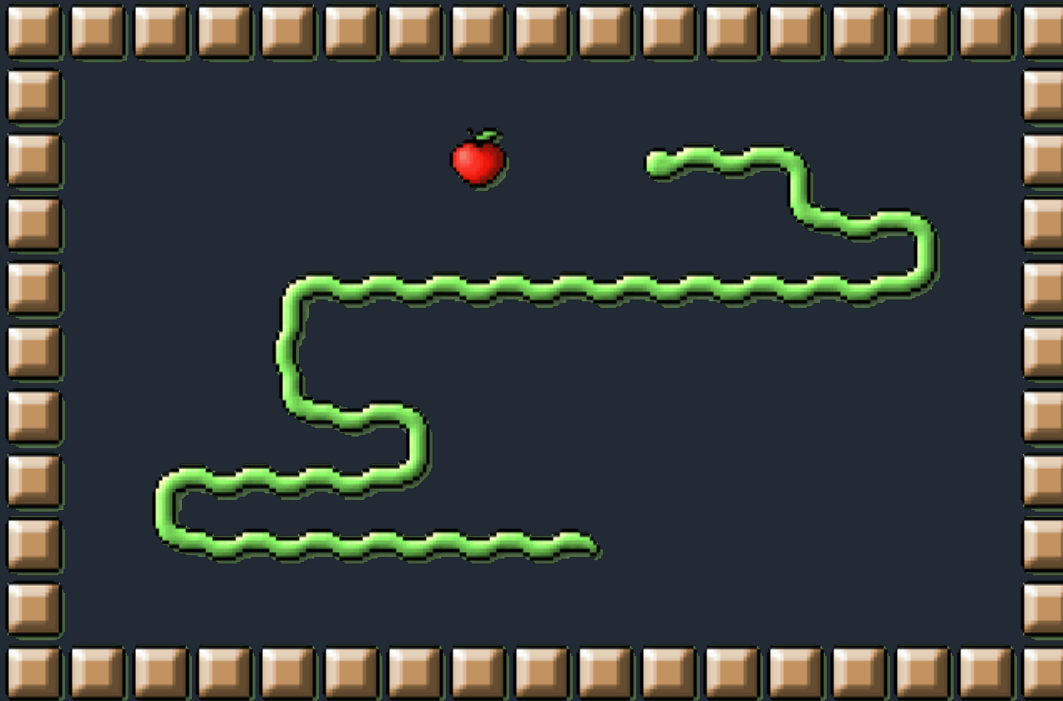
Simplest codes and
methods of Python

Make a High Quality game
that can be enjoyed by public

Quality



Game Play



1. Control dot, square or object on the plane
2. Snake leaves trail behind as it moves
3. Snake gets longer
4. Touches border or hits its body itself
5. GAME ENDS

Data Collection



pygame

News

About

Screenshots

Downloads

Documentation

Tutorials

Pygame FAQ

Help (irc, lists)

Bugs & Patches

Mercurial

Links

Resources

What's New

Wiki

Cookbook

RSS Feeds

Project Comments

projects

arcade (605)

2d (569)

pygame (520)

game (250)

puzzle (235)

shooter (187)

python (159)

strategy (153)

libraries (149)

other (142)

action (127)

space (109)

spam (101)

rog (94)

multiplayer (93)

applications (91)


gpl (78)

platformer (78)

pygamel (71)

pyweek (66)

more tags



Search the pygame.org website

News

New pygame website now the default front page - Aug 15, 2015

The 'old' version of the pygame website has been running for over six months now, but now it is the default page and not hiding itself. The old website stuff is still running and can be found at: <http://pygame.org/news.html>. You can follow development of the website via bitbucket issues <https://bitbucket.org/pygame/pygame/issues?component=website>. Also please log any issues you have with the website under the "website" component. More and more parts of the old website will slowly be replaced with the new stuff. Until then try and enjoy the Frankenstein abomination that is pygame.org

pygame sprint - Aug 15, 2015

There's another pygame sprint on this weekend. This time just in IRC/bitbucket/mailling list. (August 15th - 16th)

pygame sprint - Jul 17, 2015

There's a [pygame sprint](#) this weekend (July 18th - 19th). Checkout the facebook event for details.
<https://www.facebook.com/events/1426411697688474/>

pyglet 1.2.3 released - Jun 22, 2015

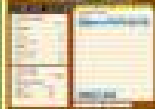
Rob van der Meid has [announced](#) the release of pyglet 1.2.3. pyglet is a cross-platform windowing and multimedia library for Python. Links:

- <http://www.pyglet.org/>
- <https://pypi.python.org/pypi/pyglet/1.2.3>

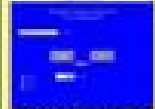
PyPy 2.6 released - Jun 15, 2015

PyPy 2.6 has been released. PyPy is an implementation of a fast, strict subset of the Python 2.7 language. PyPy features a just-in-time compiler that enables

spotlight



Altoon - 2.2




SGC - 0.2.1

our projects

pygame.org welcomes all python game, art, music, sound, video and multimedia projects. If they use pygame or not

recent releases

Jan 30, 2016



IchorPy - 1.5.1

Jan 24, 2016

no screenshot available

Data Collection



Frets on Fire



Dangerous High
School Girls in
Trouble

“FAT WORM”

Pygame



- Cross-platform set of Python modules
- Written by Pete Shinnars
- Community project
under the open source free software



Built over the Simple DirectMedia Layer (SDL) library
Allowing real-time computer game development without the low-level mechanics
of the C programming language and its derivatives



The most expensive functions inside games can be abstracted from the game logic!
Use a high-level programming language = Python, to structure the game

Coding Process



```
import pygame  
import time  
import random
```

Coding Process



❖ The colors
that you will
use for your
game

❖ The font and
color of the
texts that
will appear

❖ How big the
display window
will be
❖ The name on top
of the window



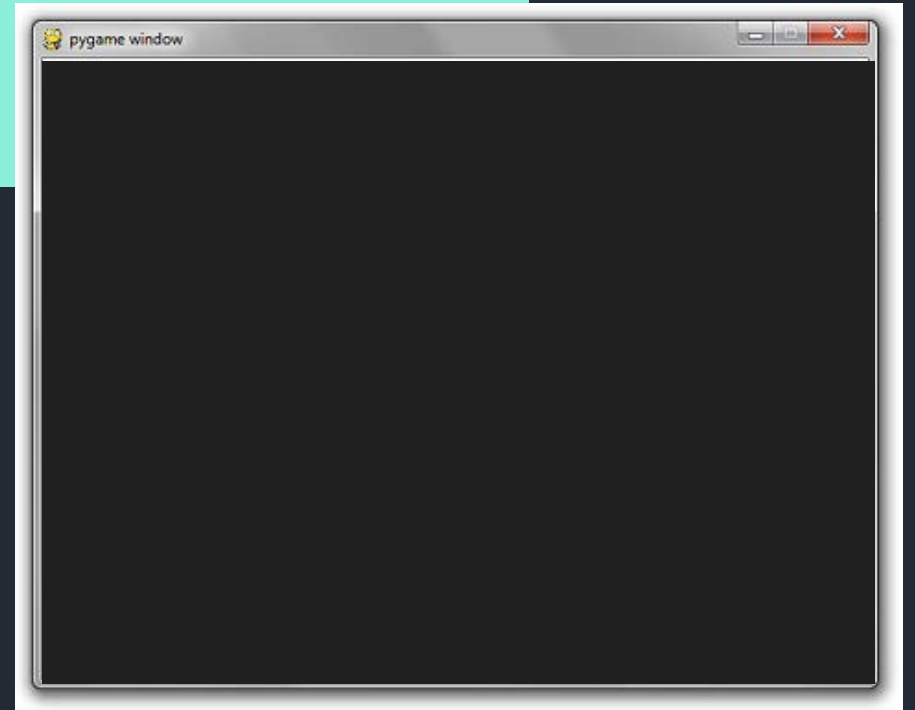
❖ Sound effects and
music throughout
the game!

Window display



```
display_width = 800
display_height = 600
size = display_width, display_height
gameDisplay = pygame.display.set_mode(size)
pygame.display.set_caption("Worm Game!")

icon = pygame.image.load("icon.png")
pygame.display.set_icon(icon)
```



Colors



```
white = (255, 255, 255)
black = (0, 0, 0)
red = (255, 0, 0)
green = (0, 155, 0)
purple = (255, 10, 255)
```

Texts

The code for the font, size and color!

```
smallfont = pygame.font.SysFont("comicsansms", 20)
medfont = pygame.font.SysFont("times", 35)
largefont = pygame.font.SysFont("宋体", 60)
```

```
def text_objects(text, color, size):
    if size == "small":
        textSurface = smallfont.render(text, True, color)
    elif size == "medium":
        textSurface = medfont.render(text, True, color)
    elif size == "large":
        textSurface = largefont.render(text, True, color)
    return textSurface, textSurface.get_rect()
```

This defines how a text message will show on the screen

```
def message_to_screen(msg, color, y_displace=0, size="small"):
    textSurf, textRect = text_objects(msg, color, size)
    textRect.center = (display_width / 2), (display_height / 2) + y_displace
    gameDisplay.blit(textSurf, textRect)
```

Features of the Worm Game



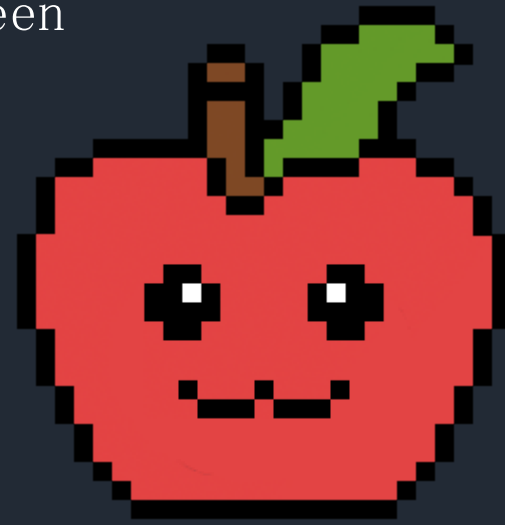
Bob the Worm

The player
controls the
worm's head



Apple

Apples that will
randomly appear on
the screen



Bob the Worm



Load the desired image, must be png format!

```
img = pygame.image.load("wormhead.png")
```



The code for how Bob's body will move

```
if lead_x > randApplex and lead_x < randApplex + AppleThickness or  
lead_x + block_size > randApplex and lead_x + block_size < randApplex +  
AppleThickness:  
    if lead_y > randAppley and lead_y < randAppley + AppleThickness:  
        randApplex, randAppley = randAppleGen()  
        snakeLength += 1  
    elif lead_y + block_size > randAppley and lead_y + block_size <  
randAppley + AppleThickness:  
        randApplex, randAppley = randAppleGen()  
        snakeLength += 1
```

Apple



Load the desired image, must be png format!

```
appleimg = pygame.image.load("apple.png")
```



The code for the apple to appear randomly

```
AppleThickness = 30
def randAppleGen():
    randApplex = round(random.randrange(0, display_width -
    AppleThickness))
    randAppley = round(random.randrange(0, display_height -
    AppleThickness))
    return randApplex, randAppley
```


Gameplay



❖ How to pause and quit the game

❖ Bob dies which means game over .

❖ There are two situations in which Bob will die.

❖ How to play this game (how to control Bob)



❖ How to get score
❖ The score that you get will show on the screen.

Play

Firstly, some instructions for the player.

```
def game_intro():  
    intro = True  
    while intro:  
        for event in pygame.event.get():  
            if event.type == pygame.QUIT:  
                pygame.quit()  
                quit()  
            if event.type == pygame.KEYDOWN:  
                if event.key == pygame.K_c:  
                    intro = False  
                if event.key == pygame.K_q:  
                    pygame.quit()  
                    quit()  
        gameDisplay.fill(white)  
        bg = pygame.image.load("introbackground.png")  
        gameDisplay.blit(bg, [0,0])  
        pygame.display.update()
```

```
message_to_screen("Welcome to our Worm Game!", red, -100,"large")  
message_to_screen("As you know, the more apple you eat, the longer you become and the more scores you get!", black, -30)  
message_to_screen("But if you eat yourself or you run into the walls, you die!", black, 50)  
message_to_screen("Press C to play, P to pause, Q to quit.", black, 180)  
pygame.display.update()  
clock.tick(15)
```



Play

How to control Bob?

Player can press 'up' , ' down' , ' left' , ' right' to control Bob.

```
def snake(block_size, snakeList):  
    if direction == "right":  
        head = pygame.transform.rotate(img, 270)  
    if direction == "left":  
        head = pygame.transform.rotate(img, 90)  
    if direction == "up":  
        head = img  
    if direction == "down":  
        head = pygame.transform.rotate(img, 180)  
  
    gameDisplay.blit(head, (snakeList[-1][0], snakeList[-1][1]))  
  
    for XnY in snakeList[:-1]:  
        pygame.draw.rect(gameDisplay, green, (XnY[0], XnY[1], block_size, block_size))
```



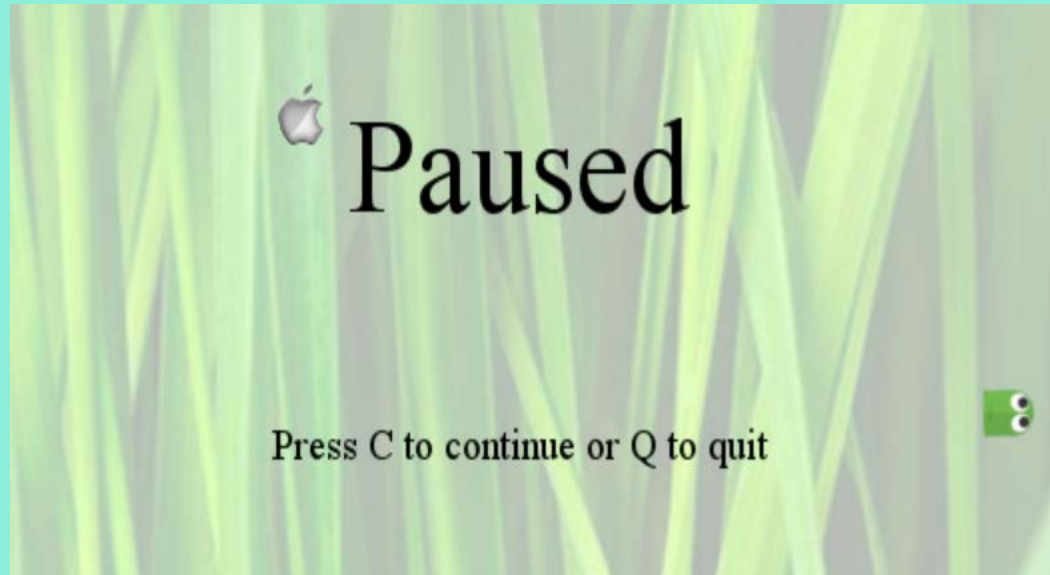
Pause & Quit

The player can press 'P' to pause the game if he wants to.

```
def pause():  
    paused = True  
    message_to_screen("Paused", white, -100, size="large")  
    message_to_screen("Press C to continue or Q to quit", white, 40)  
    pygame.display.update()
```

And while paused, player can press 'C' to continue the game or press 'Q' to quit the game.

```
while paused:  
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
            pygame.quit()  
            quit()  
        if event.type == pygame.KEYDOWN:  
            if event.key == pygame.K_c:  
                paused = False  
            elif event.key == pygame.K_q:  
                pygame.quit()  
                quit()  
    clock.tick(5)
```



Game over

There are two situations in which Bob will die.

1). Bob dies if it crashes into the wall.

```
if lead_x >= display_width or lead_x < 0 or lead_y < 0 or lead_y >= display_height:  
    gameOver = True  
    dead_sound.play()
```

2). Bob dies if it eats itself.

```
if len(snakeList) > snakeLength:  
    del snakeList[0]  
for eachSegment in snakeList[:-1]:  
    if eachSegment == snakeHead:  
        gameOver = True  
        dead_sound.play()
```

Game over

“gameOver” is False in the gameLoop, this is the lines for when gameOver is True, that is when Bob dies and it is Game Over.

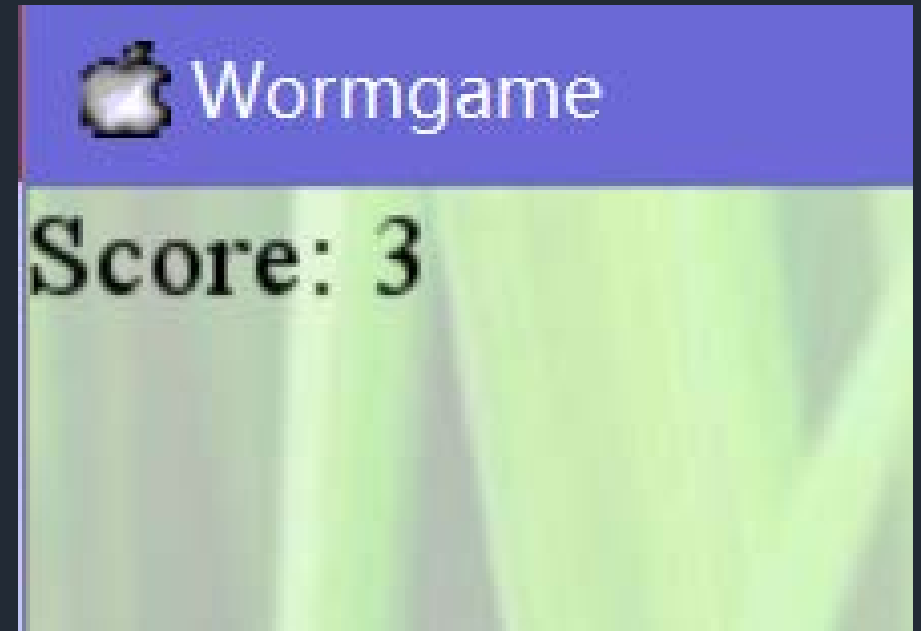
```
while running:  
    if gameOver == True:  
        message_to_screen("Game over",red ,-50,size="large")  
        message_to_screen("Press C to play again, or Q to quit", white, 50,  
size="medium")  
        pygame.display.update()
```



Score

Each time Bob eats one apple, the score increases by one.

```
def score(score):  
    text = smallfont.render("Score: " + str(score), True, white)  
    gameDisplay.blit(text, [0, 0])
```



OPTIMIZATION

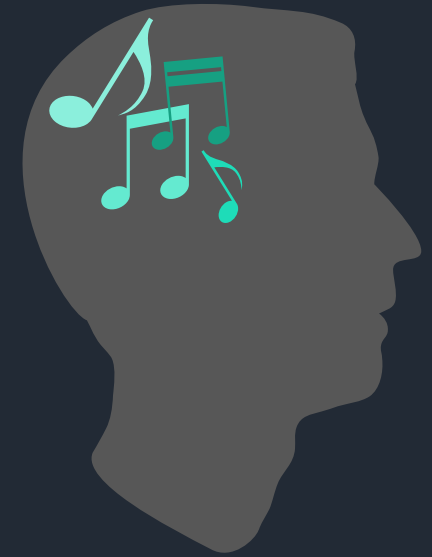
Background



```
gameDisplay.fill(white)  
bg = pygame.image.load("background.png")  
gameDisplay.blit(bg, [0,0])  
pygame.display.update()
```



OPTIMIZATION



Music



```
intro_sound = pygame.mixer.Sound("intro.wav")
```

```
intro_sound.play()
```

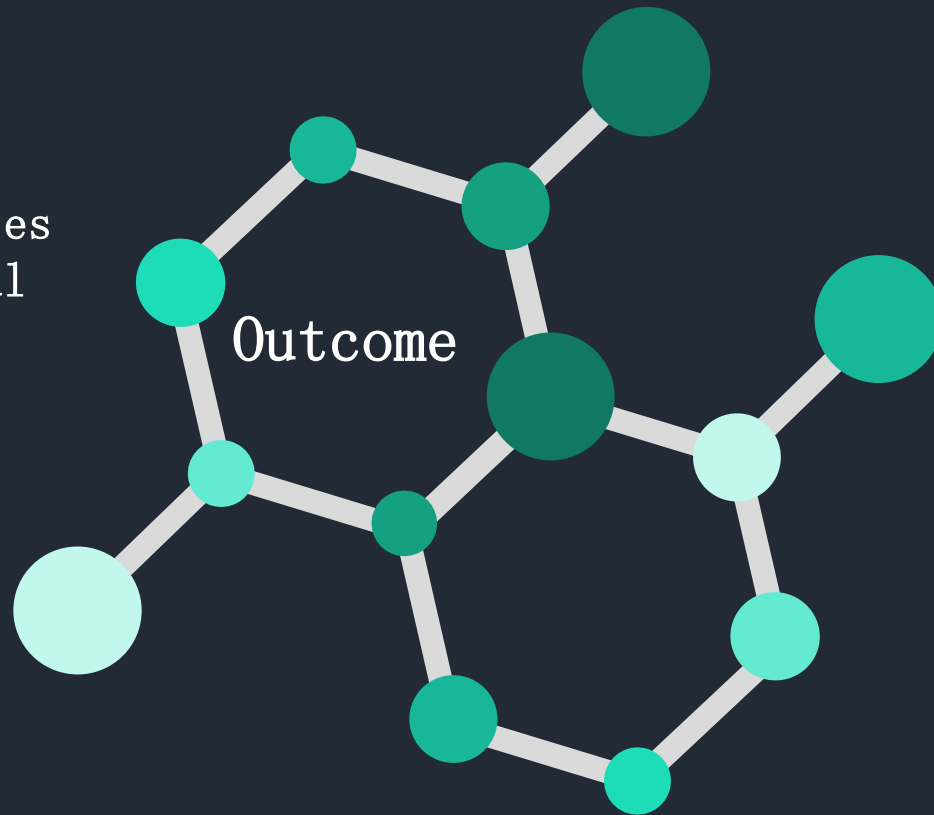
Result



Programming requires
math,
We gave a lot of tries
to make it successful



Basic, good for
beginners to
understand



2-Dimensional
Not appealing



But we enjoyed this
project,
Because we
experienced every
process

Challenges faced



Zero experience
in Pygame



01

02

Thank you Professor Baoyang!



Audio Problem
Why not working?
Audio should be .wav file

