FINAL REPORT

2018

（Jointly Carry Highest GPA）

JCHG

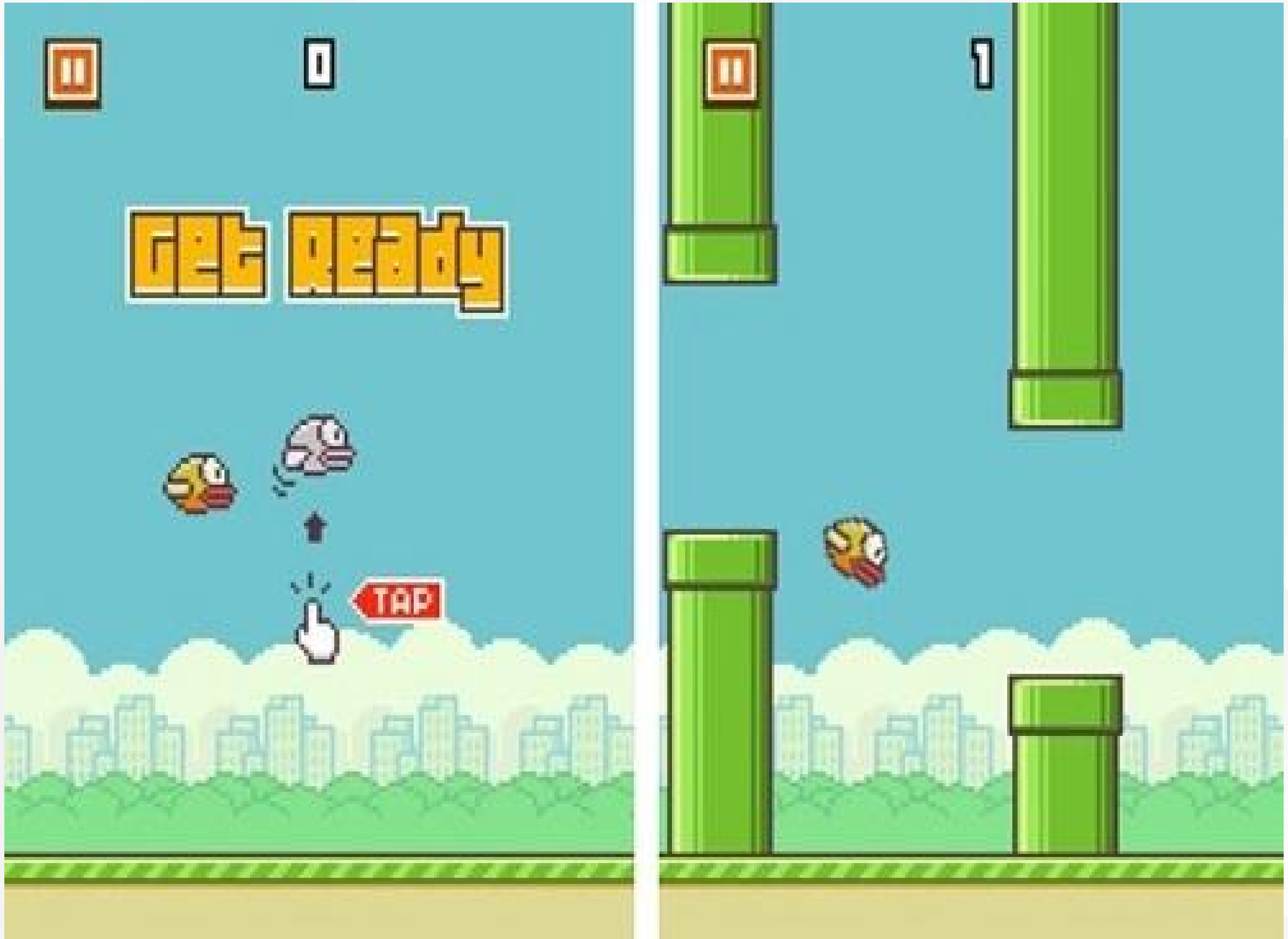
吉雨薇 陈欣 黄欣纯 高雅

01

Background

What inspired us?

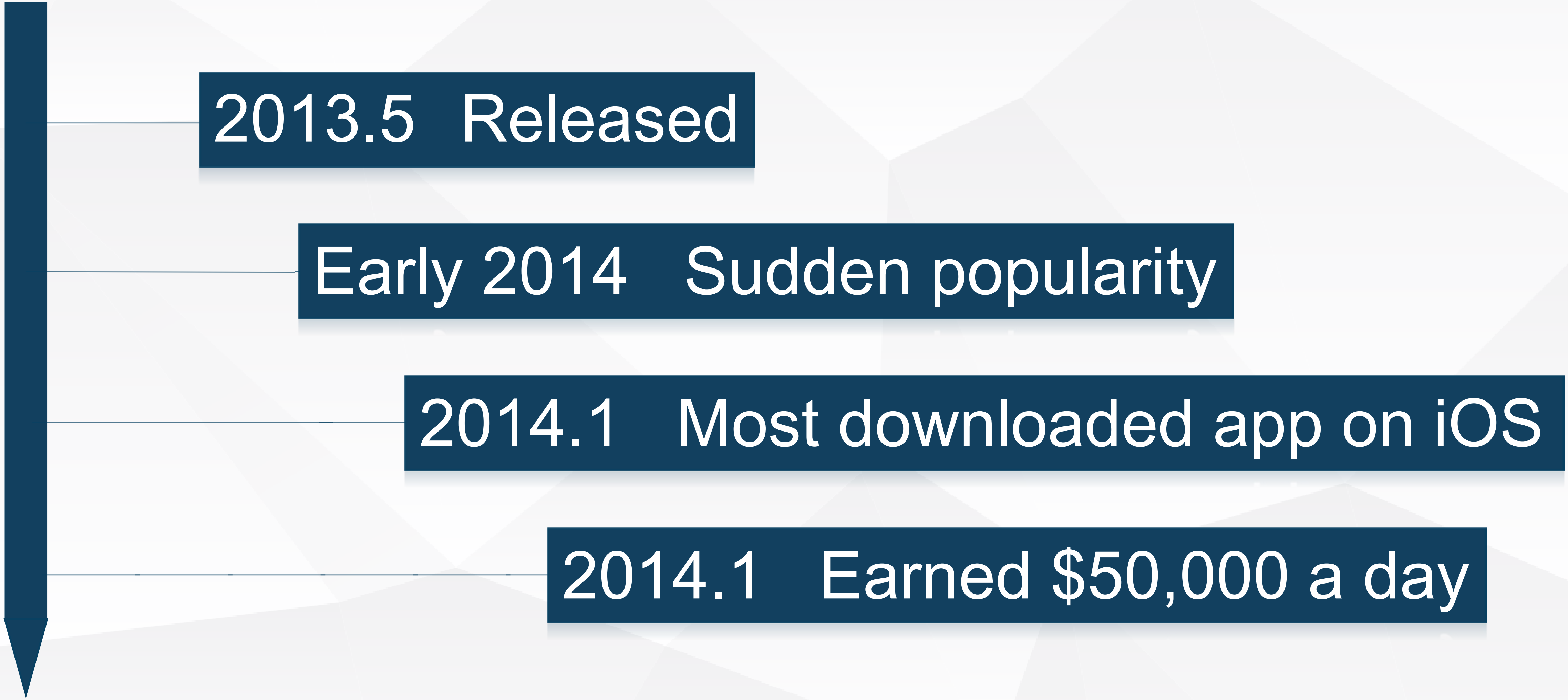
Background



Flappy Bird

a mobile game developed by Vietnamese video game artist and programmer Dong Nguyen.

Background



2013.5 Released

Early 2014 Sudden popularity

2014.1 Most downloaded app on iOS

2014.1 Earned $50,000 a day

We have made

Improvement

to realize audio control !

02

Programming

How did we realize all these awesome things by Python?

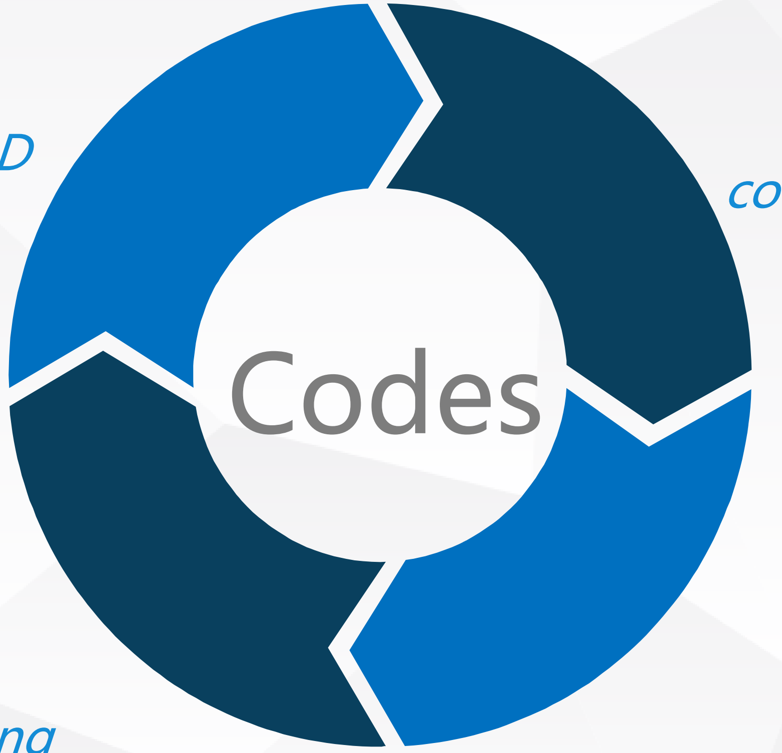
Programming

Audio control

The most brilliant part :-D

Module

Effective modules that contribute to programming



Codes

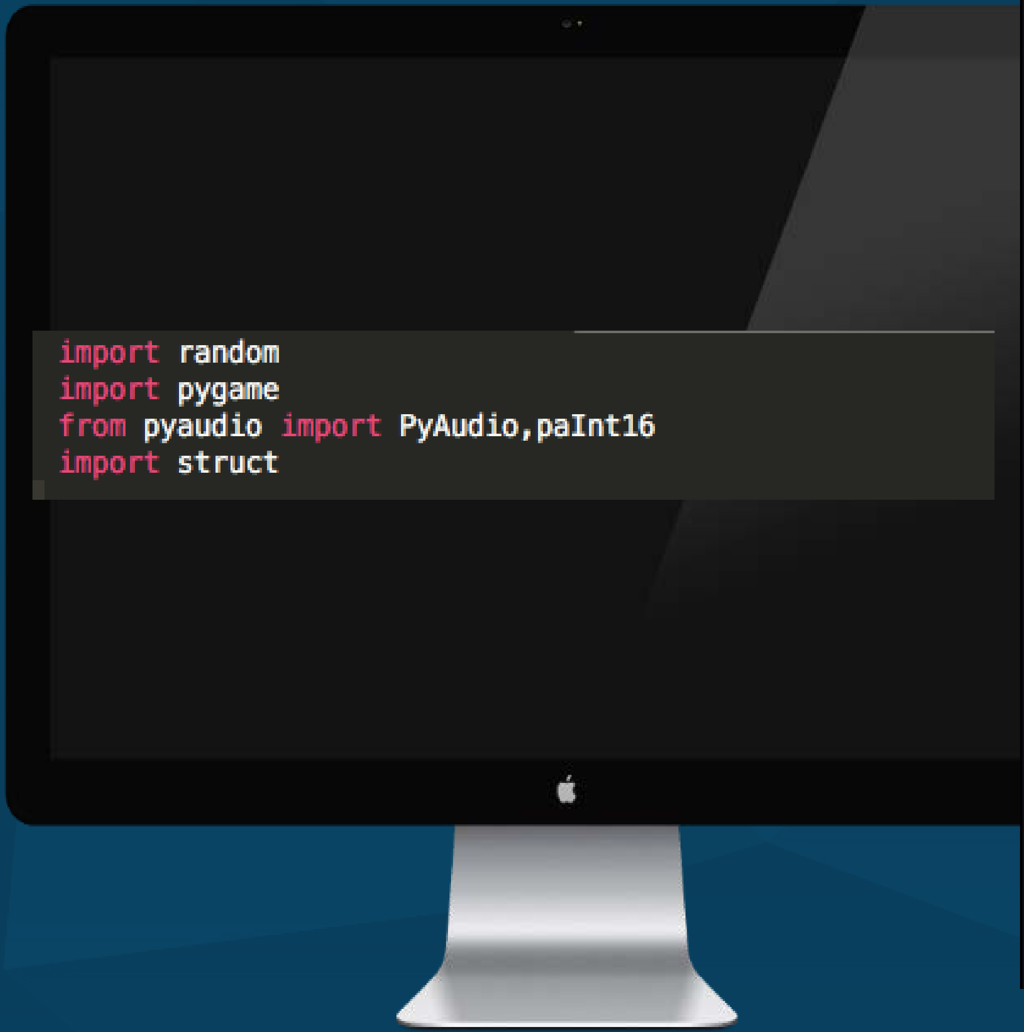
Main loop

What happens when ur playing

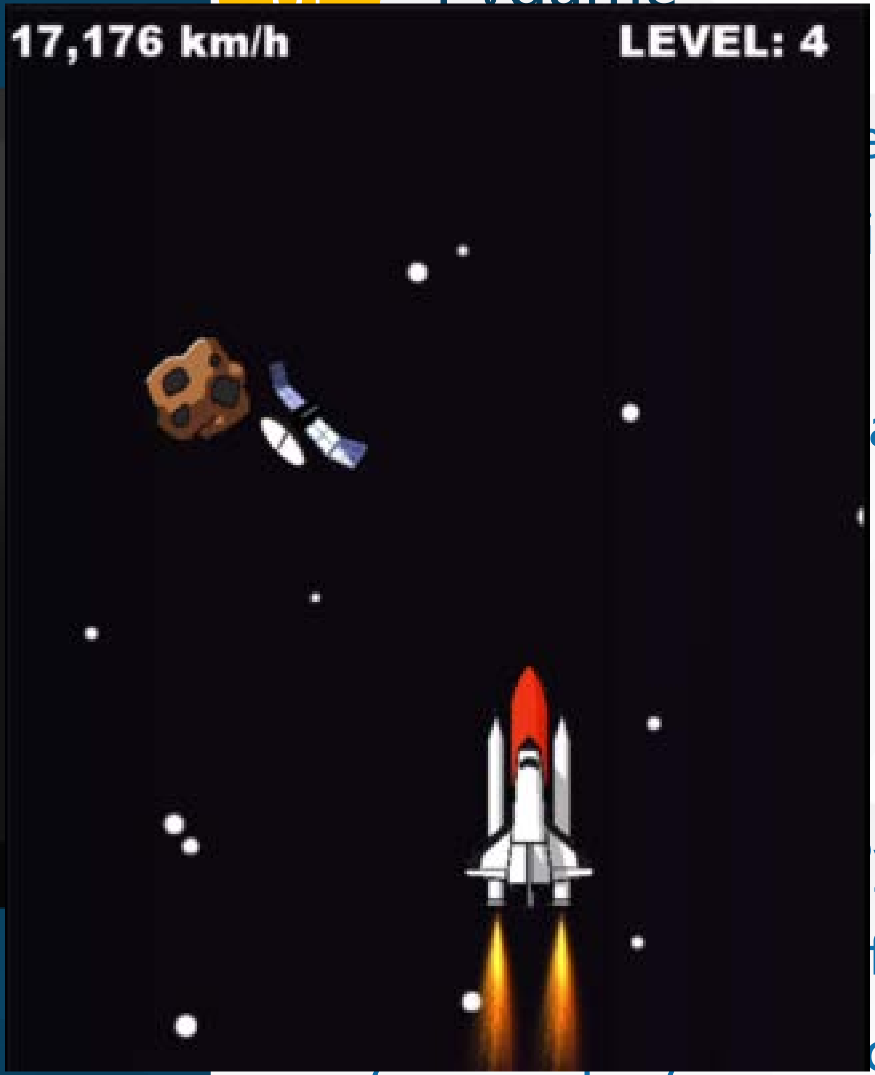
Game Logic

How can the bird fly?

Part 1 Module



```
import random
import pygame
from pyaudio import PyAudio,paInt16
import struct
```



01

Pygame

e language library for

ications like games

atform and operating

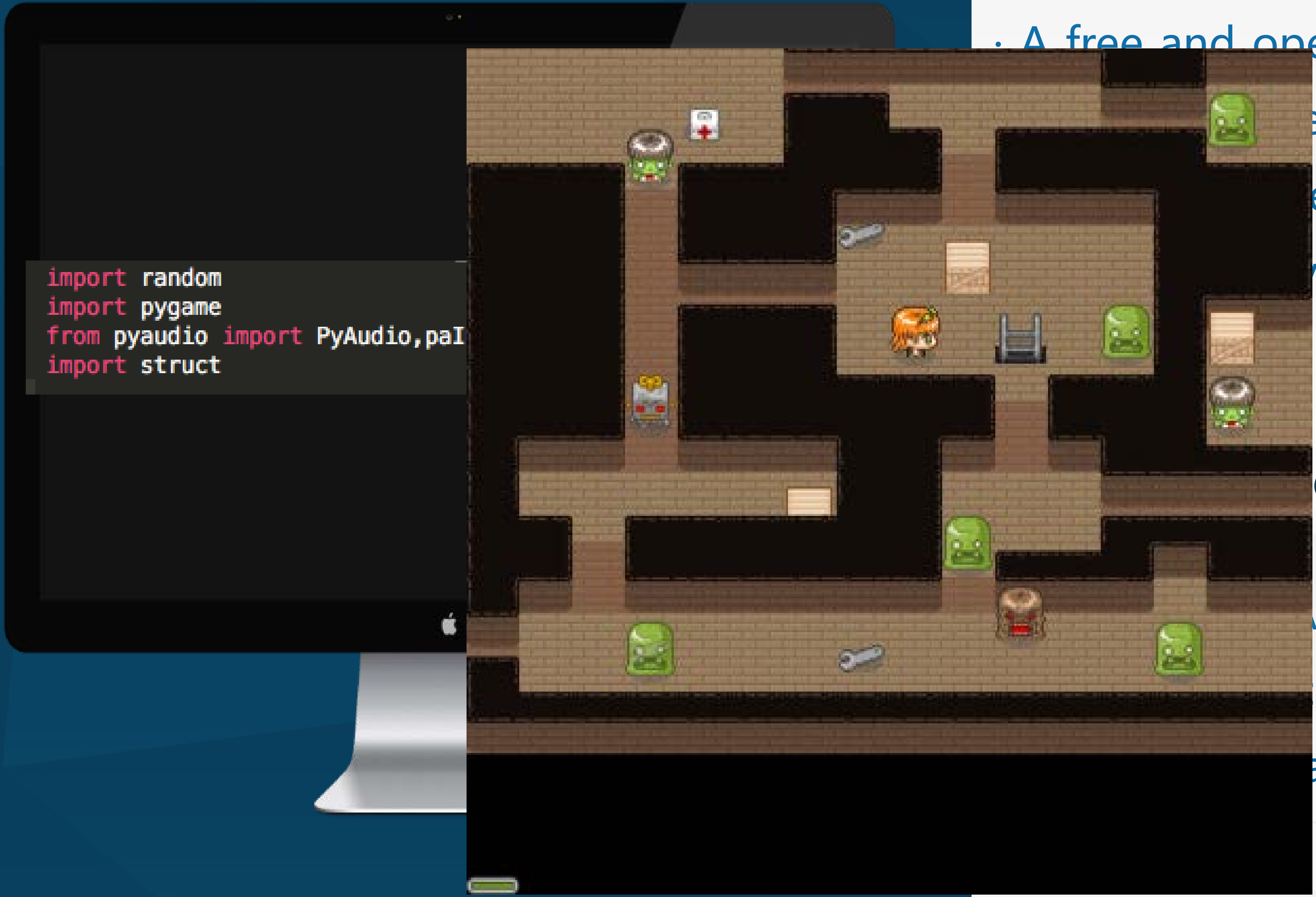
ython bindings for

form audio I/O library.

cord audio on a variety

of platforms

Part 1 Module



```
import random
import pygame
from pyaudio import PyAudio,paI
import struct
```

01

Pygame

· A free and open source language library for

edia applications like games

e

y every platform and operating

o

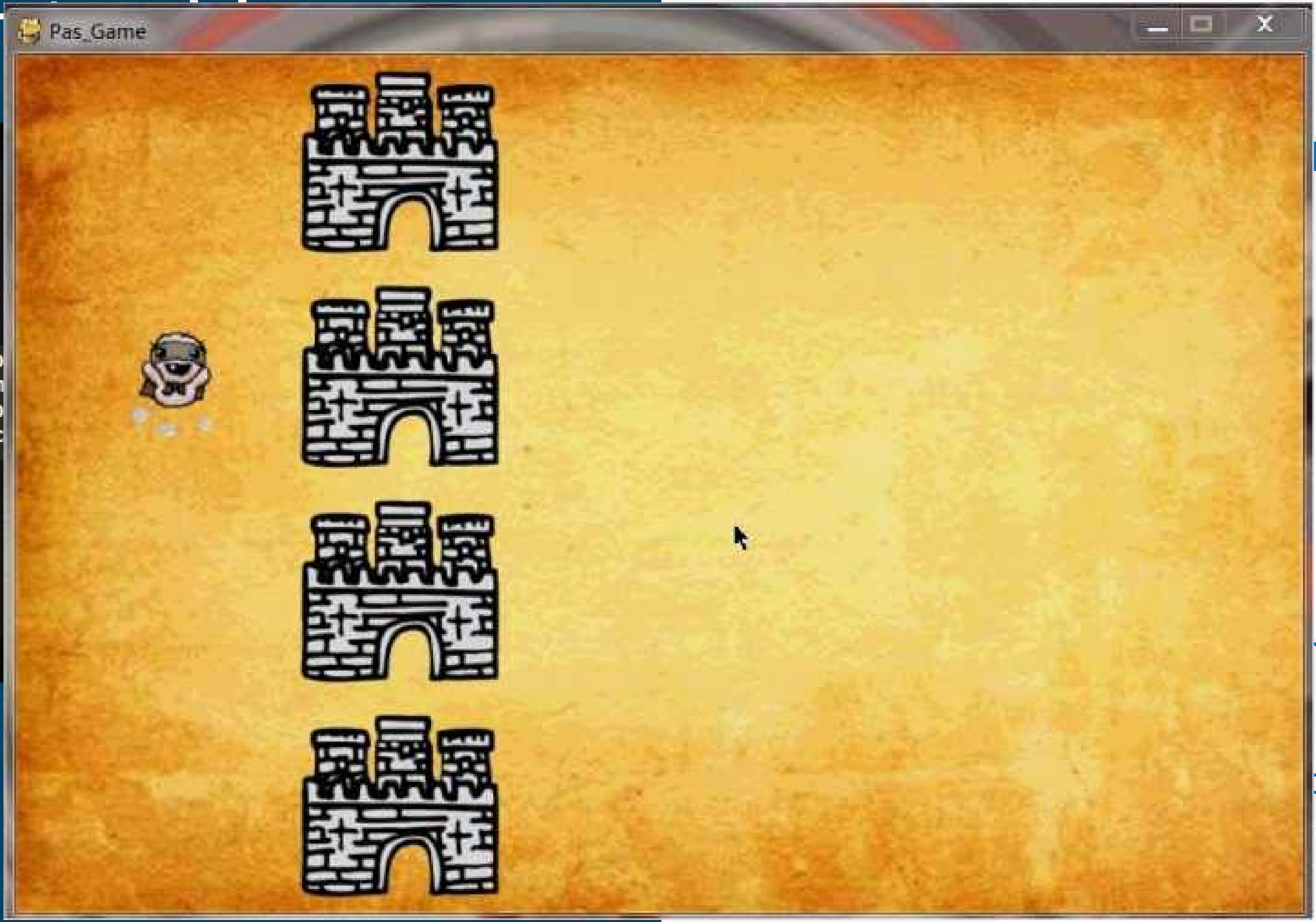
vides Python bindings for

ross-platform audio I/O library.

ay and record audio on a variety

Par

```
import rando
import pygam
from pyaudio
import struc
```



library for

games

operating

ndings for

I/O library.

on a variety

00

Do some preparation!

Important variables:

```
bird

pipe

gameState =1
          =2
          =3

*k
```

Display it on the screen:

```
WHITE = (255,255,255)
BLACK = (0,0,0)
PIPE = (117,190,49)
SKY = (78,192,202)
GROUND = (224,215,146)
DARK_GROUND = (124,115,46)
BIRD = (241,186,62)
size = (800,700)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Flappy Block")
```

Do some preparation!

Important variables:

```
bird

pipe

gameState  =1

           =2

           =3

*k
```

```
class Pipe():
    def __init__(self):
        self.centerY = random.randrange(130,520)
        self.x = 800
        self.size = 150
```

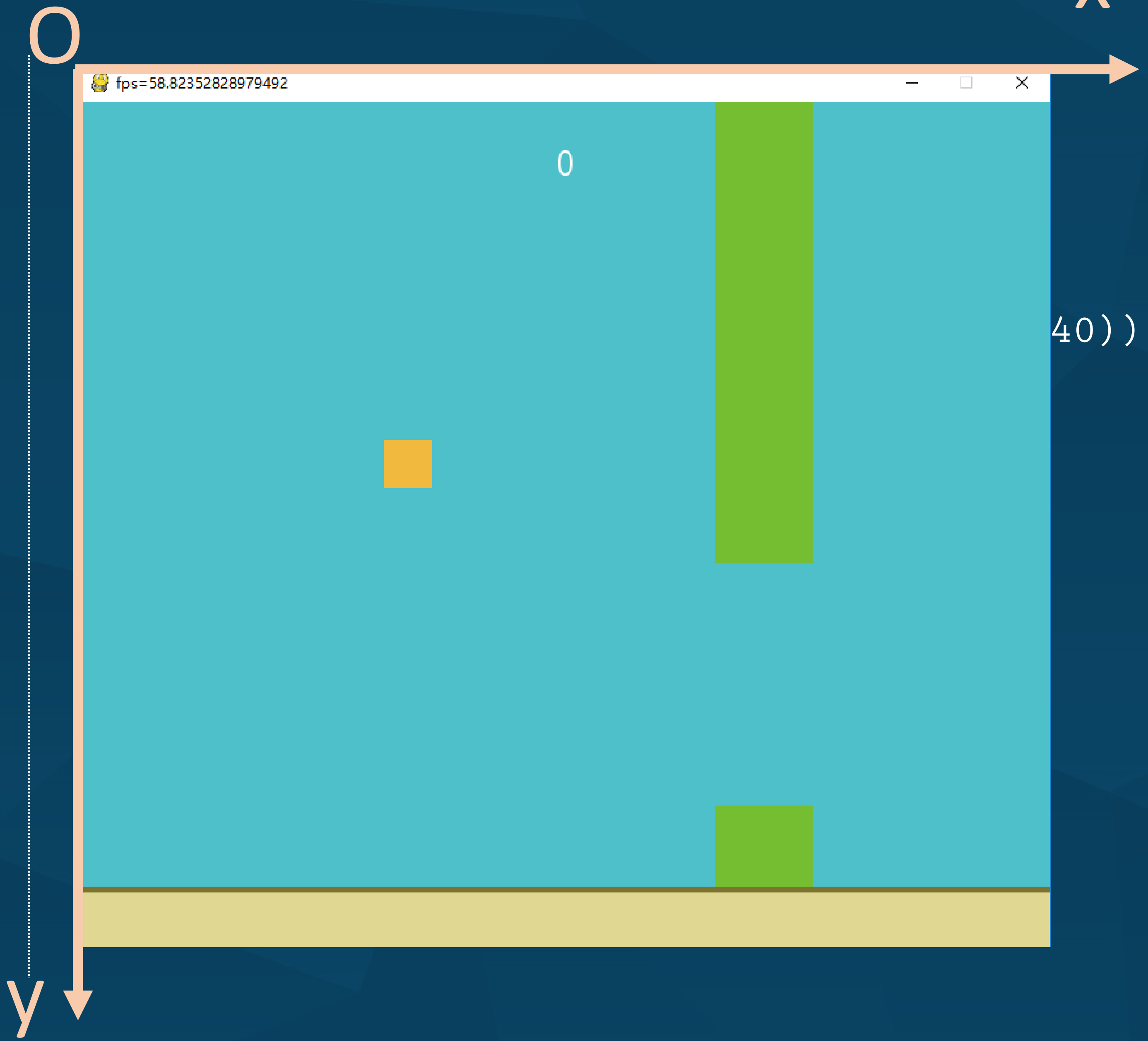
01

Make the birds fly!

```
class Bird:
    def __init__(self):#初始位置
        self.x = 250
        self.y = 250
       self.yV = 0
        # self.yV = △ self.y


  def flap(self):#控制鸟的上升
        self.yV=-0.0008*k


  def update(self):
        self.yV += 0.5#模拟重力加速度
        self.y += self.yV
    if self.y >= 600:
         self.y = 600
         self.yV = 0
    if self.yV > 20:
          self.yV = 20
```



40))

01

Make the birds fly!

```
class Bird:
    def __init__(self):#初始位置
        self.x = 250
        self.y = 250
       self.yV = 0
        # self.yV = △ self.y


  def flap(self):#控制鸟的上升
        self.yV=-0.0008*k


  def update(self):
        self.yV += 0.5#模拟重力
加速度
        self.y += self.yV
    if self.y >= 600:
         self.y = 600
         self.yV = 0
    if self.yV > 20:
          self.yV = 20
```

```
def draw(self):
      pygame.draw.rect(screen,
                       BIRD,
                       (self.x,self.y,40,40))

def reset(self):#重设鸟的位置
    self.x = 250
        self.y = 250
        self.yV = 0

bird = Bird()
```

02

When the bird collides with the pipe..

```
def draw(self):
    pygame.draw.rect(screen,PIPE,(self.x,0,80,(self.centerY - self.size)))
    pygame.draw.rect(screen,PIPE,(self.x,(self.centerY + self.size),80,(548 - self.centerY)))

if self.x >= 170:
    and self.x <= 290
    and bird.y <= (self.centerY - self.size)
    or
    self.x >= 170
    and self.x <= 290
    and (bird.y + 40) >= (self.centerY + self.size):
    gameState = 3#结束状态
```

Part 3 Main Loop

□ gameState=1

Draw pipes randomly.

Print instuctions.

□ gameState=2

Fly or flop!

□ gameState=3

Print new instuctions.

Print highest score.

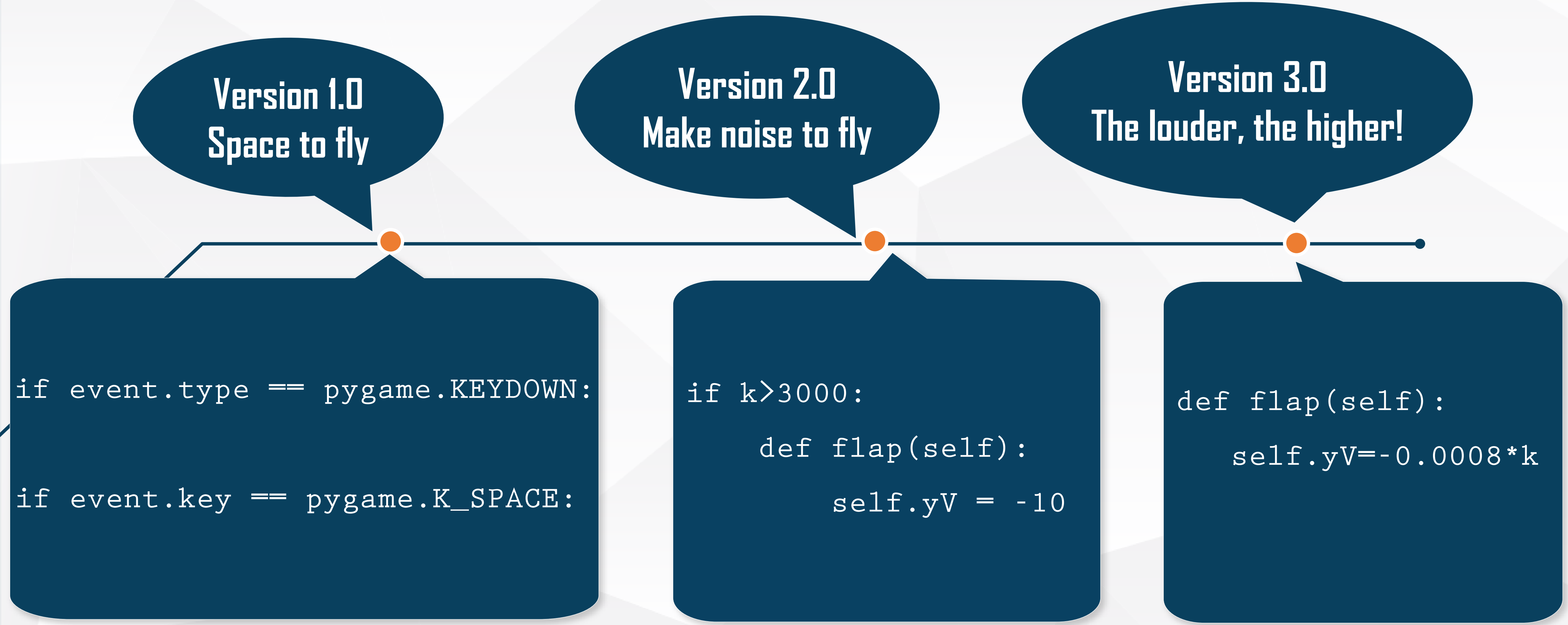
Finishing the task accordingly

Event Loop

Determining the gameState

```
if k>3000
    if gameState == 1:
        gameState = 2
    elif gameState == 3:
        bird.reset()
        pipes = []
        pipes.append(Pipe())
        gameState = 2
        score = 0
    else:
          bird.flap()
```

Part 4 Audio Control



Version 1.0
Space to fly

```
if event.type == pygame.KEYDOWN:

if event.key == pygame.K_SPACE:
```

Version 2.0
Make noise to fly

```
if k>3000:
    def flap(self):
        self.yV = -10
```

Version 3.0
The louder, the higher!

```
def flap(self):
   self.yV=-0.0008*k
```

Change the condition of the event loop

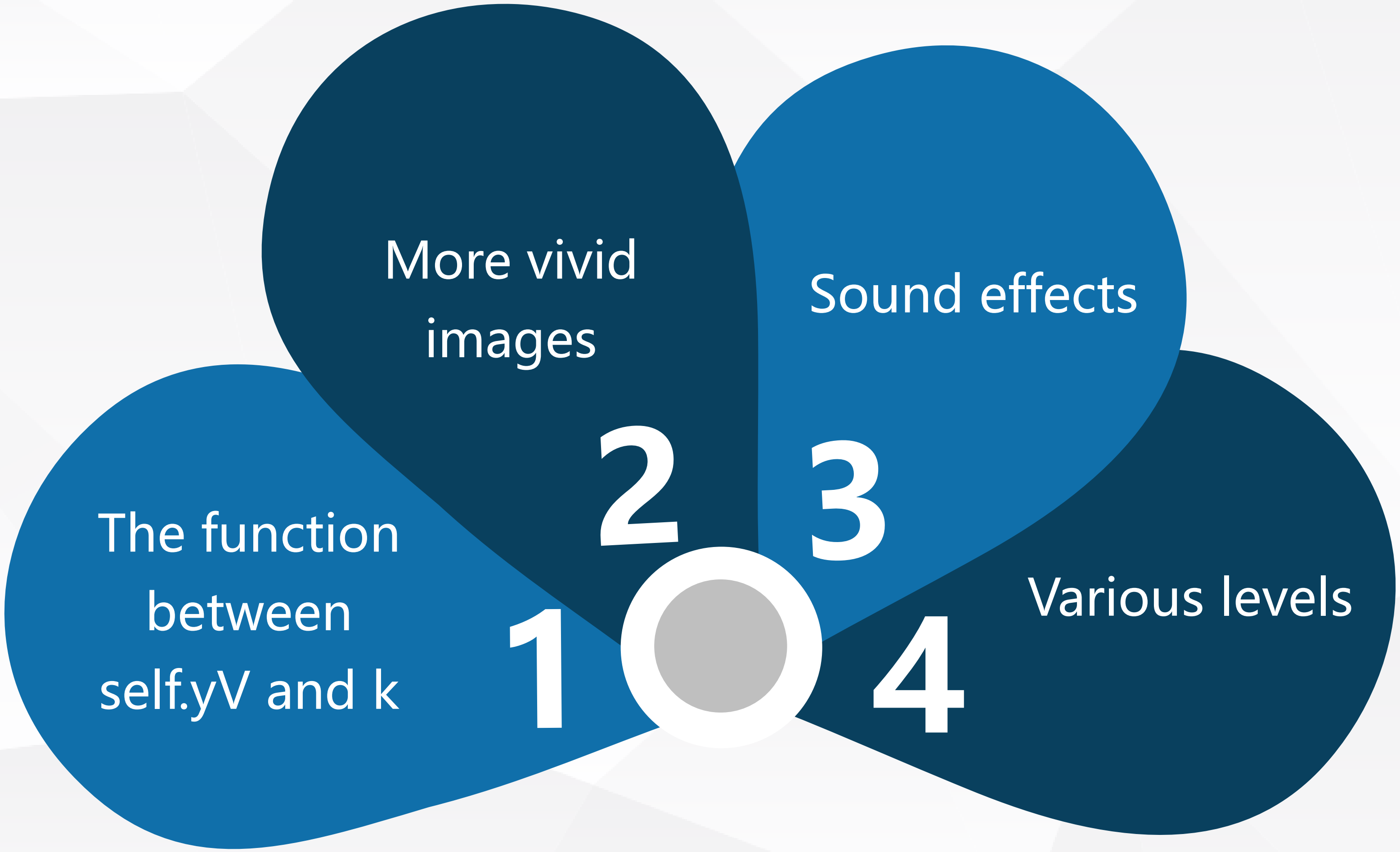
Change the function between self.yV and k

03

Further improvement

What can be further improved?

Further improvement



1 The function between self.yV and k

2 More vivid images

3 Sound effects

4 Various levels

Further improvement

01

The function between self.yV and k

```
# We have tried:
self.yV = -0.0005 * k
self.yV = -math.log(k)

# Currently applied:
self.yV = -0.0008 * k
```

self.yV: = △self.y

Negative → up

The larger abs(self.yV) is, the higher it can fly up each time.

k： loudness of sound

Further improvement

02 More vivid images

03 Sound effects

04 Various levels

…

THANK YOU FOR WATCHING!

Let’s Jointly Carry Highest GPA :P