mindX
Technology & Startup School

Session 13

Game

mindX
Technology & Startup School

Pygame module and Pong game

Content

1. Installing Pygame
2. How Pygame works
3. Drawing
4. Animations
5. Input
6. Opponent ‘AI’

Pygame?

- Pygame is a python package
- Pygame is a collection of different modules. Some of the modules are written in C, and some are written in Python.

Install Pygame

1. Check if you already have pip.exe and python.exe in PATH
- Add to path: System environment variables
- Check pip in cmd: pip --version
- Check python in cmd: python -version

2. Install Pygame module
- Type in cmd: pip install pygame
- Check pygame: import pygame

```
pygame 2.0.3 (SDL 2.0.16, Python 3.9.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

How pygame works

Begin: pygame.init()

Setup

Game logic
Data

Loop

Drawing
Updating
(All game animations run in here)

End: pygame.quit()

Initialize some basic stuff

Setup

- Set display screen
- Set caption

Loop

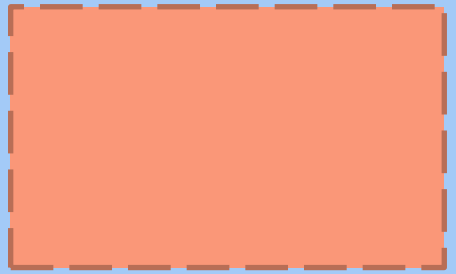
Dealing with user input: using module *pygame.event*
- Get all user's actions as input
- Check for type of event (type of input)

Updating the window
- *.flip()* in module *pygame.display*
- *.tick()* in module *pygame.time*

Drawing in pygame

mindX Technology & Startup School

Some basic elements:



Rectangle
Additional surface rounded by a rectangle that is able to move

(regular) Surfaces
Additional layers that can be attached later on

Display Surface object
The main layer that is displayed to the user

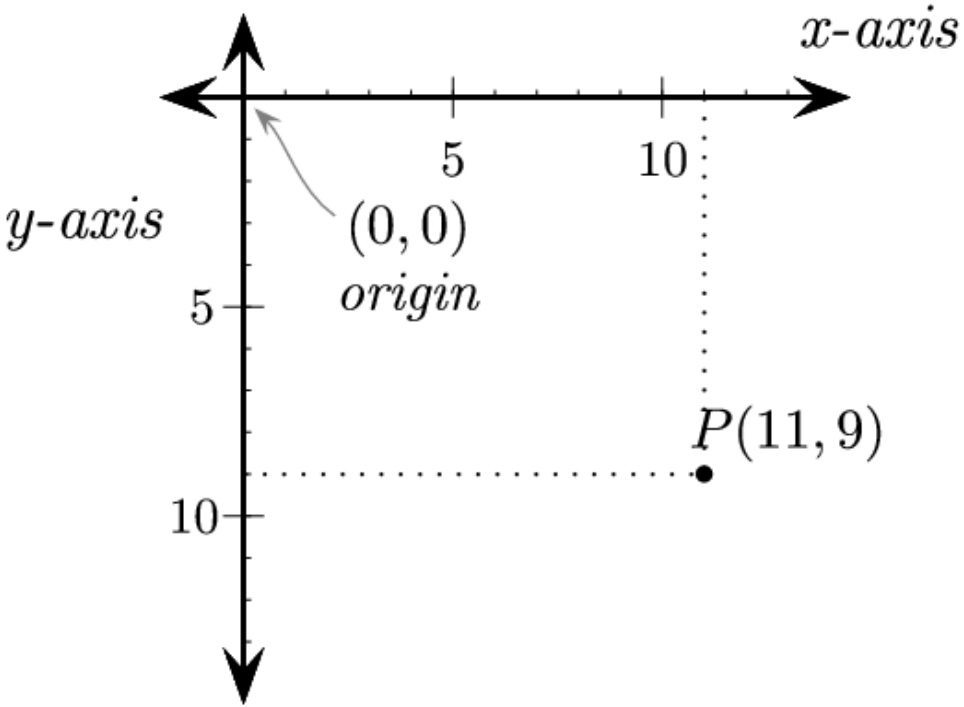
Drawing in pygame

We need **Rectangle** to carry images because an image cannot move by itself

Let’s take a quick example of Cheem story

Drawing in pygame

Coordinate



Drawing in pygame

Now let's create some Rects for carrying stuff: *pygame.Rect()*

- Ball
- Player
- Opponent

Drawing in pygame

Draw colors

1. **Using RGB color = a tuple (r, g, b), each value is in range(256)**
2. **Create a color object: pygame.Color('name')** // search: "color table"

Drawing in pygame

Draw shape

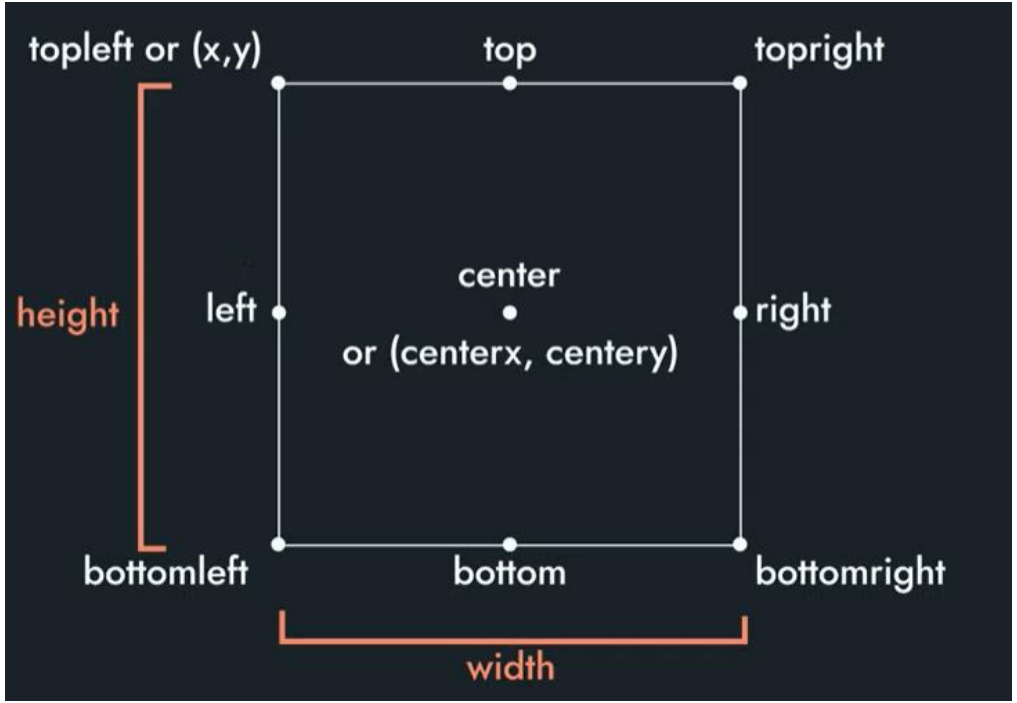
[pygame.draw — pygame v2.1.1 documentation]

Using functions below to draw on a Rect:

- **pygame.draw.rect(surface, color, RecValue)**
- **pygame.draw.ellipse(surface, color, RecValue)**
- **pygame.draw.aaline(surface, color, start_pos, end_pos)**

Draw order:
- *First - bottom*
- *Last - top*

Animations



Rectangle attributes

Move object by adding incremental changes of these attributes over loop

Let's try to move the **ball** by changing its rect's attribute over the loop

Animations

Ball animation

How to keep the ball stay inside the screen?

Just use these Rect attributes to make a bouncing ball

Animations

Ball animation

How to make the ball bounces against **Player** and **Opponent**?

rect1.colliderect(**rect2**)

Animations

Ball animation

Resetting the ball

If **Ball** hits left or right wall *(ball.left* or *ball.right):*

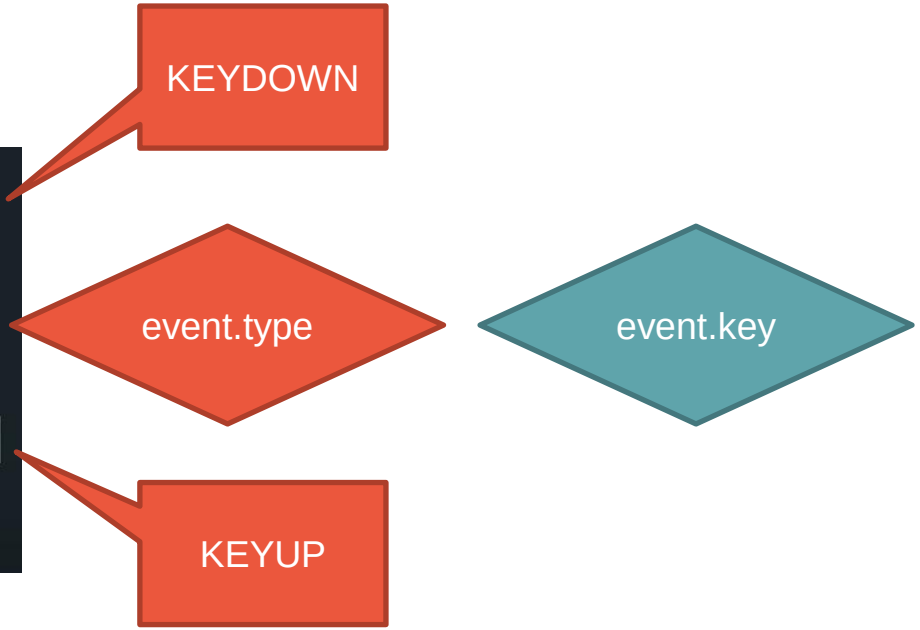
- Teleport it to the center: *ball.center*
- Restart in random direction: *ball_speed *= random.choice()*

Begin in random direction

Animations

Players' animations

1. A button has been pressed
   - ▷ start movement
2. A button has been released
   - ▷ stop movement



Movement of player depends on user input

Animations

Player animation

How should we make the **player** animation?

1. Declare player speed variable
2. Add this speed to the player on every frame
3. No **button** pressed: speed = 0
4. **Button** pressed: increase/decrease **player speed** becomes positive or negative

Animations

Player animation

How to keep the **player** stay inside the screen?

Same with the ball!

Animations

Opponent animation

Movement logic

- If **Opponent**'s top above ball => Move down
- If **Opponent**'s bottom below ball => Move up

- Also prevent the **Opponent** from leaving the screen

mindX
Technology & Startup School

Pong game upgrade

Content

1. Basic Scoring (how to add text)
2. Fixing collision error between ball and player/opponent
3. Timer

Score Counter

1. Variables to store scores

2. Text on screen to display it

- Create a font & font size: (eg: `freesansbold.ttf`)
  Any font: *pygame.font.Font('font name', size)*
  System font: *pygame.font.SysFont('font name', size)*

- Write text on a new surface (create an image (surface) of text)
  Create surface: *text = font.render('text', aliasing, color)*

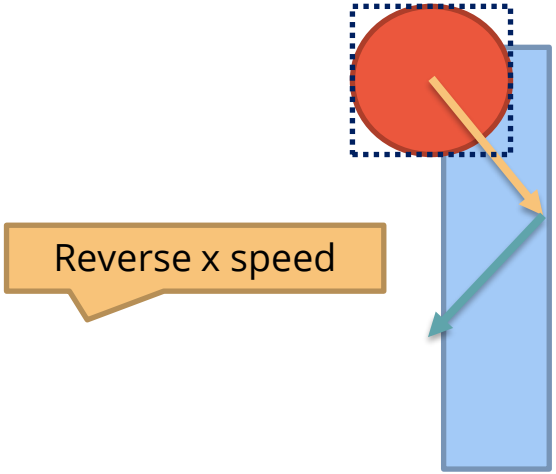
- Put the text surface on the main surface
  "surface".blit(source, dest)

3. Increase score when the ball hits walls

Collision

Collide with player *if ball.colliderect(player) and ball_speed_x > 0:*

|ball.right – player.left| < 10

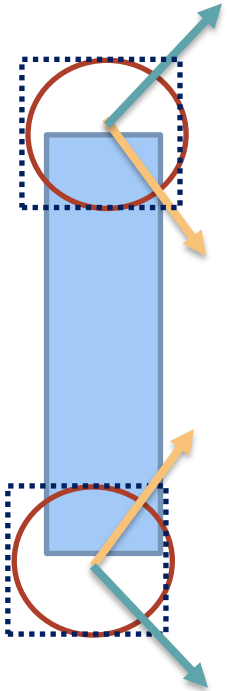


else: ***|ball.right – player.left| ≥ 10***

|ball.bottom – player.top| < 10
ball_speed_y > 0

Reverse y speed

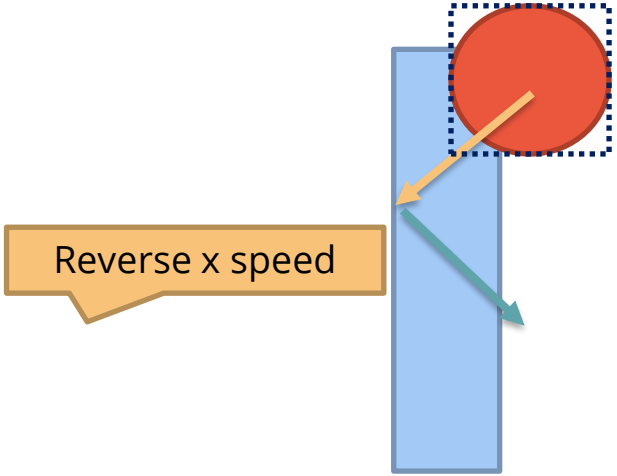
|ball.top – player.bottom| < 10
ball_speed_y < 0



Collision

Collide with opponent *if ball.colliderect(opponent) and ball_speed_x < 0:*

|ball.left – opponent.right| < 10

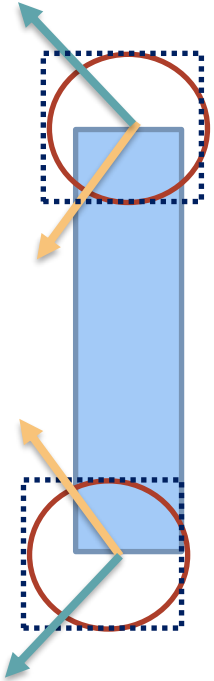


else: ***|ball.left - opponent.right| ≥ 10***

|ball.bottom – opponent.top| < 10
ball_speed_y > 0

Reverse y speed

|ball.top – opponent.bottom| < 10
ball_speed_y < 0



Timer

(waiting time before the ball starts moving)

1. Variables to store time: *score_time* (should be set *None* first)

2. Text on screen to display it

- When the ball hits (left/right) wall, get the moment:

```
score_time = pygame.time.get_ticks( )
```

- Only runs *ball_start( )* function when *score_time != None* in *while* loop:

```
if score_time:
    ball_start( )
```

3. Redesign the *ball_start( )* function so it can count as a timer:

- Using *render* to display time left on screen (3, 2, 1, go....)
- The ball keeps staying at the middle of the screen until done counting

mindX
Technology & Startup School

THANKS!

See you in the next lesson!