

# Left and right inter-fighting!

Team: 509 (three roommates in 509)  
张佳诚 谭智壕 王逸其





Background

Simplify/  
Change

Programming

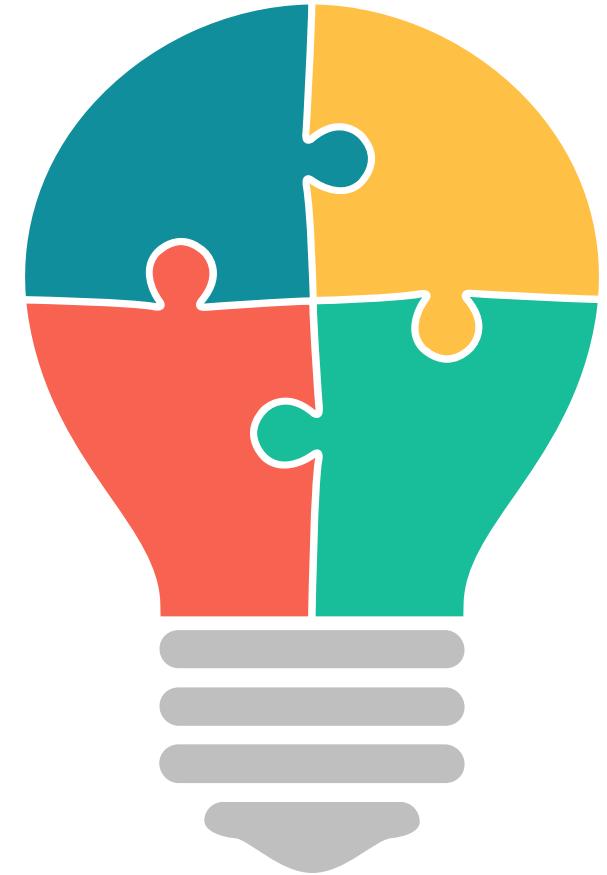
Demo time

# Background

PART 01

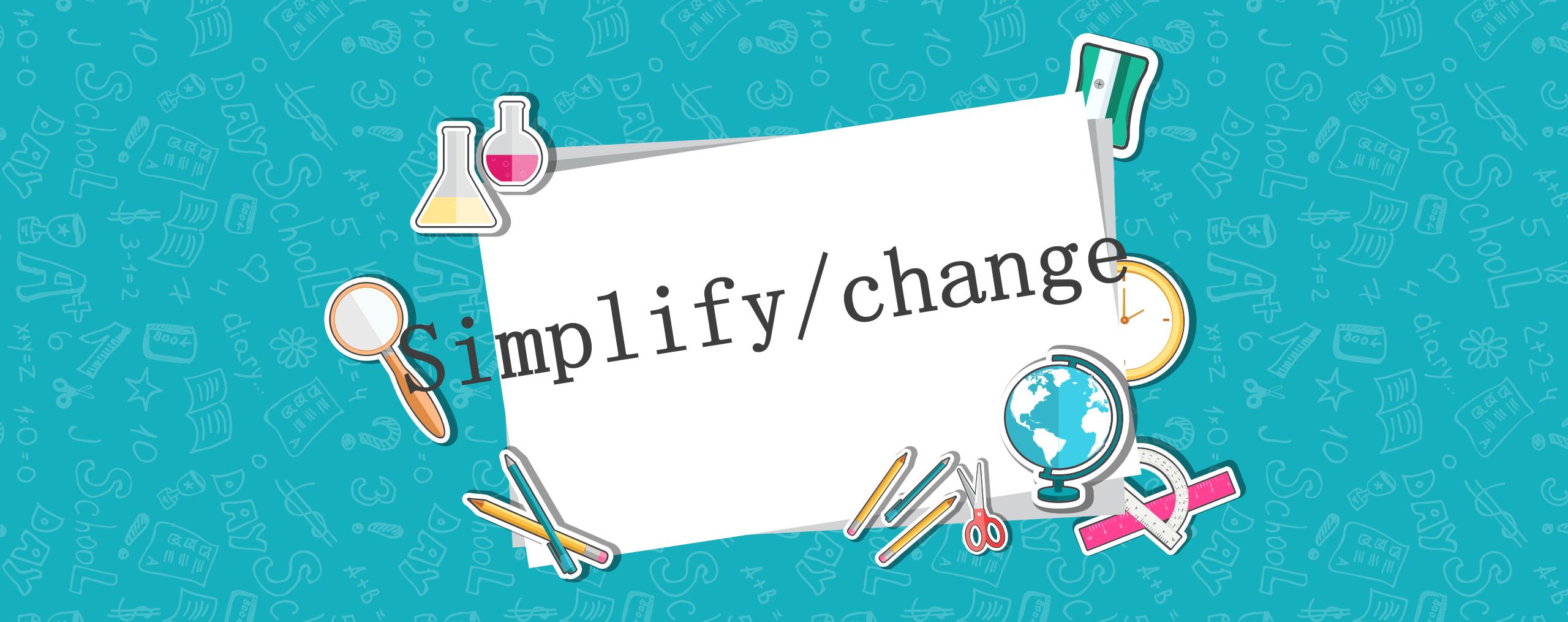
# background

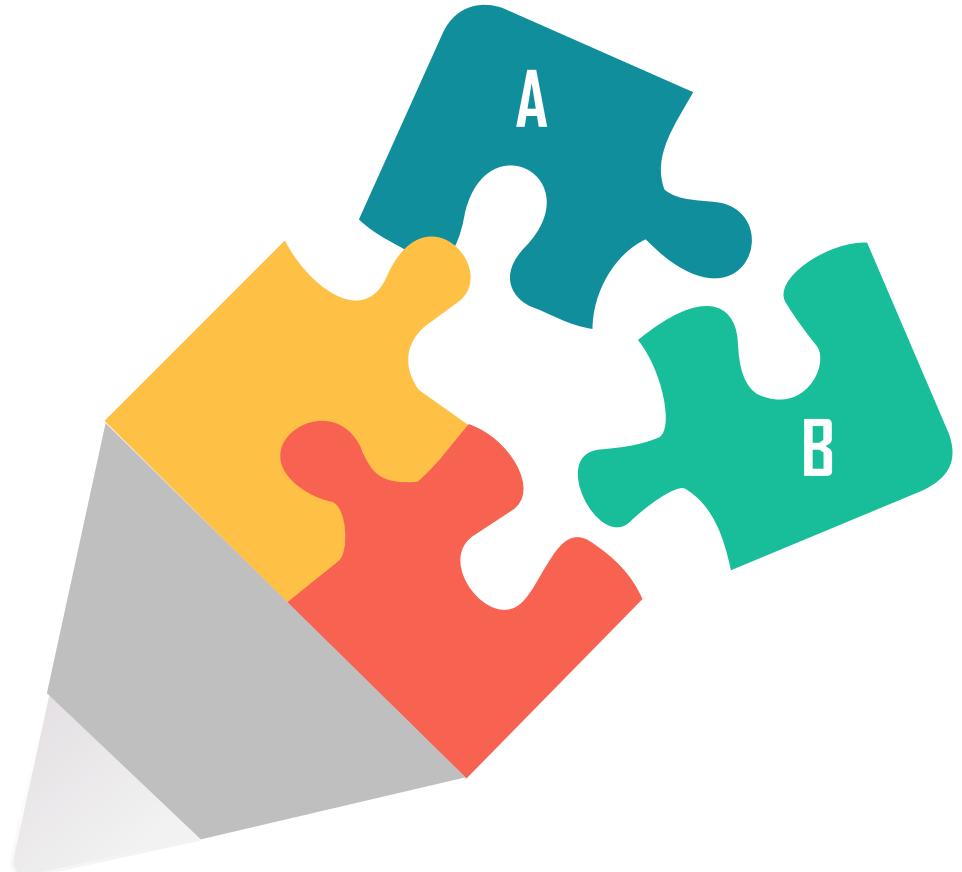
- 1.Game lover(by using pygame)
- 2.Inspiration from an amusing game  
(a game about running and chasing)



## PART 02

Simplify / change





The rule of the game:

- \*Two players control the same chess
- Attacker aims at pushing it out of the court
- Defender aims at keeping it in the court
- \*The power of the attacker will increase
- \*20 seconds for each round

simplify the game

Use some simple methods in pygame



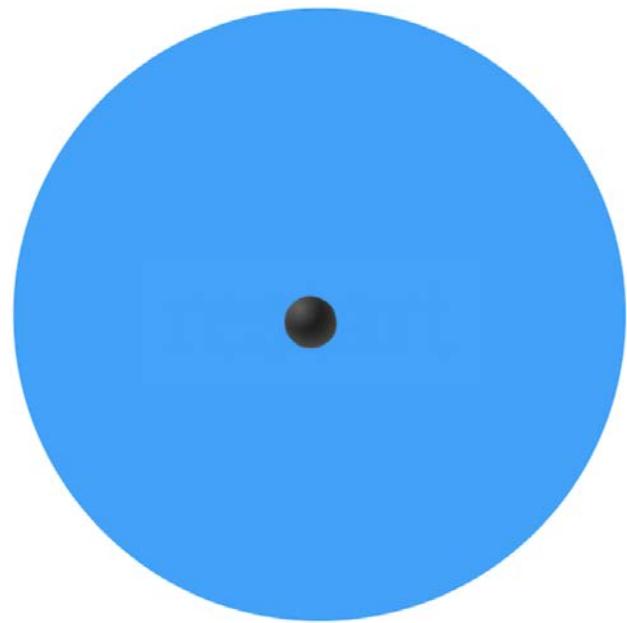
# programming

PART 03



# preparation

pygame window



```
pip install pygame
```

```
screen.blit(im_court, (0, 0))  
screen.blit(im_chess, (327, 327))  
screen.blit(im_start, (200, 289))  
pygame.display.set_icon(logo)  
pygame.display.flip()  
pygame.mixer.Channel(1).play(bgm)
```



# programing

## preparation

These two functions combined together can serve as a timer.

```
| def timer(time, timerevent):  
|     while time[1] == True:  
|         time[1] = False  
|         print(time[0])  
|         pygame.time.set_timer(timerevent, 1000)  
  
| def countdown(time):  
|     if time[0] > 0:  
|         time[0] -= 1  
|         time[1] = True
```



# programing

## preparation

Precise decision on winning and losing

```
|def get_action(action):  
|    pygame.time.set_timer(action, 40)  
  
|def checkwin(pos):  
|    if (pos[0] + 22.5 - 358) ** 2 + (pos[1] + 22.5 - 350) ** 2 > 265 ** 2:  
|        return True  
|    else:  
|        return False
```



# programing

## preparation

### Some variables

```
running = True  
playing = False  
moving1 = [0, 0]  
moving2 = [0, 0]  
pos = [327, 327]  
time = [20, True]  
timerevent = pygame.USEREVENT + 1  
action = pygame.USEREVENT  
winstatus = 0  
difficulty = 2.5
```



# Game loop

—about time and difficulty

```
while running:  
    for event in pygame.event.get():  
        if event.type == pygame.MOUSEBUTTONDOWN:  
            if not playing:  
                playing = True  
                get_action(action)  
                timer(time, timerevent)  
                screen.blit(im_court, (0, 0))  
                screen.blit(im_chess, (327, 327))  
                winstatus = 0  
                difficulty = 2.5  
                pos = [327, 327]  
                pygame.display.update()  
                time[0] = 20
```

```
if event.type == timerevent:  
    countdown(time)  
    timer(time, timerevent)  
  
if time[0] == 20 and playing:  
    difficulty = 2.5  
if time[0] == 15 and playing:  
    difficulty = 3.5  
if time[0] == 10 and playing:  
    difficulty = 4.5  
if time[0] == 5 and playing:  
    difficulty = 5
```



# programing

## Let's start the game!

### Tips:

\*During the game process, if players do some actions like pressing 'wasd' or 'direction keys', the movement of the chess will be reflected on the screen.

\*Of course, every time an event is checked, the computer will check if any player has win the game.

```
if event.type == action:  
    if playing:  
        get_action(action)  
    if pygame.key.get_focused():  
        keys = pygame.key.get_pressed()  
        if keys[pygame.K_d] != 0 and keys[pygame.K_a] == 0:  
            moving1[0] = 1  
        elif keys[pygame.K_a] != 0 and keys[pygame.K_d] == 0:  
            moving1[0] = -1  
        else:  
            moving1[0] = 0  
        if keys[pygame.K_w] != 0 and keys[pygame.K_s] == 0:  
            moving1[1] = 1  
        elif keys[pygame.K_s] != 0 and keys[pygame.K_w] == 0:  
            moving1[1] = -1  
        else:  
            moving1[1] = 0  
        if keys[pygame.K_RIGHT] != 0 and keys[pygame.K_LEFT] == 0:  
            moving2[0] = 1  
        elif keys[pygame.K_LEFT] != 0 and keys[pygame.K_RIGHT] == 0:  
            moving2[0] = -1  
        else:  
            moving2[0] = 0  
        if keys[pygame.K_UP] != 0 and keys[pygame.K_DOWN] == 0:  
            moving2[1] = 1  
        elif keys[pygame.K_DOWN] != 0 and keys[pygame.K_UP] == 0:  
            moving2[1] = -1  
        else:  
            moving2[1] = 0  
    pos[0] += 10 * moving1[0]  
    pos[0] += moving2[0] * 2 * difficulty  
    pos[1] -= 10 * moving1[1]  
    pos[1] -= moving2[1] * 2 * difficulty
```



# Demo time!!

PART 04



# Demo time !!



# Shortage

- (1) The balance of the game
- (2) The conciseness of the code
- (3) The complexity of our project



thanks  
for your watching

speaker 谭智壕

