The slide features several decorative colored dots of various sizes scattered across the background. There are dots in shades of green, blue, red, orange, yellow, and grey.

# 第一小组成果汇报

## ——打 砖 块

A yellow dot is positioned to the left of the text.

汇报人：余玉非

A grey dot is positioned to the right of the text.

2017年12月12日

# 目录

01

成员介绍

02

功能演示

03

总体设计

04

特色和创新点



# 成员介绍

余玉非：15级软件工程 组长 程序

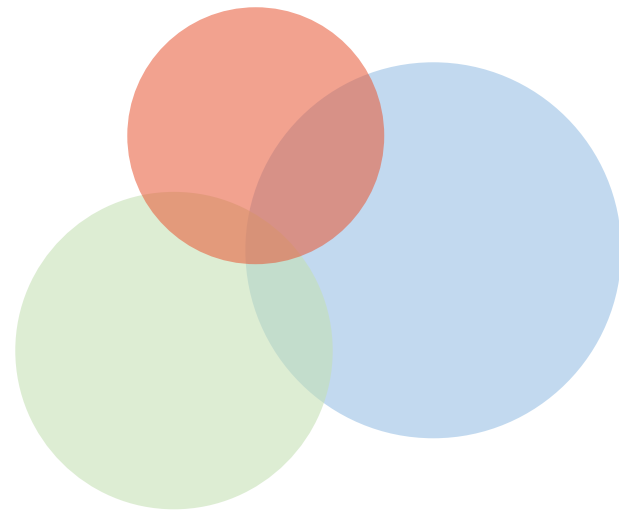
吴 超：16级软件工程 程序

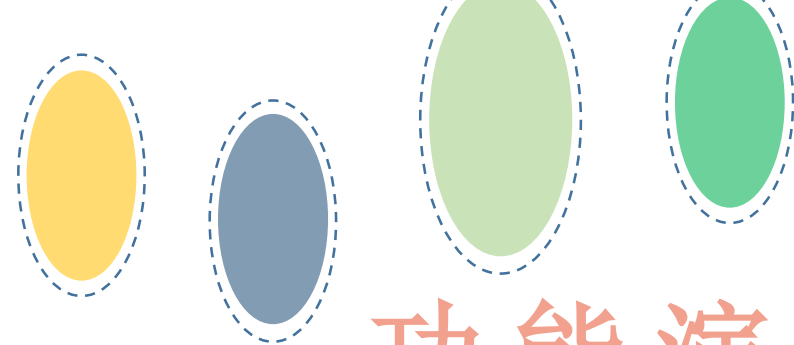
连思源：16级通信工程 程序

吴 雄：16级计算机工程与技术 程序

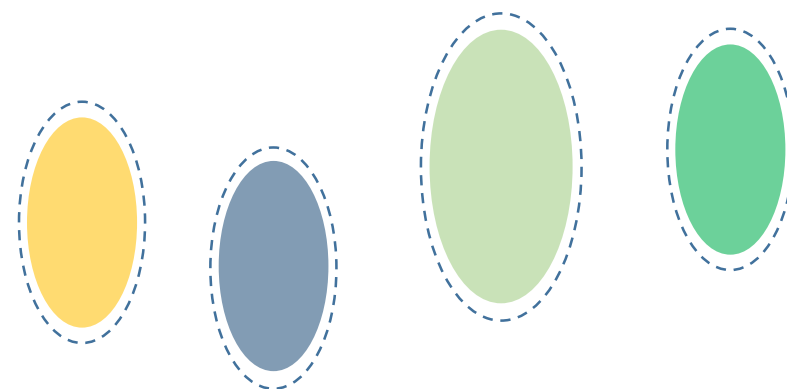
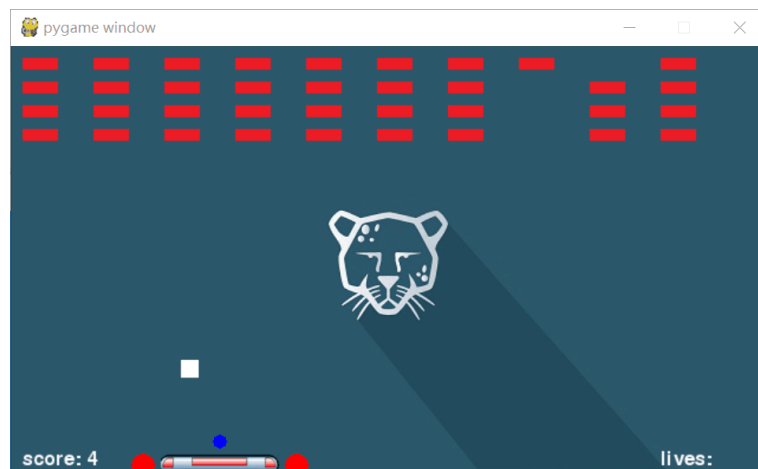
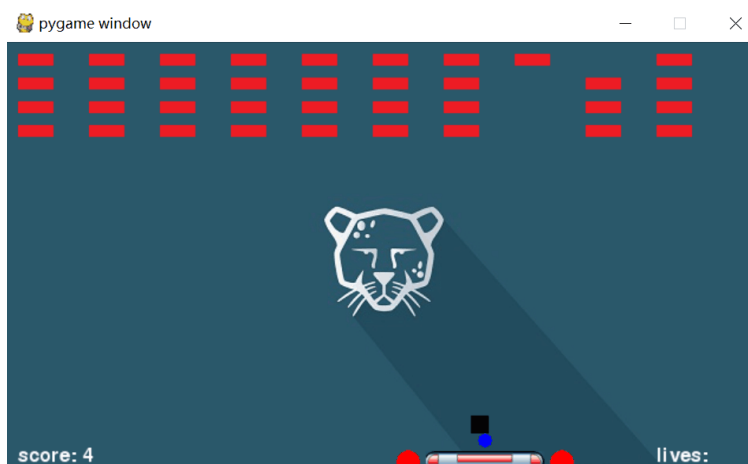
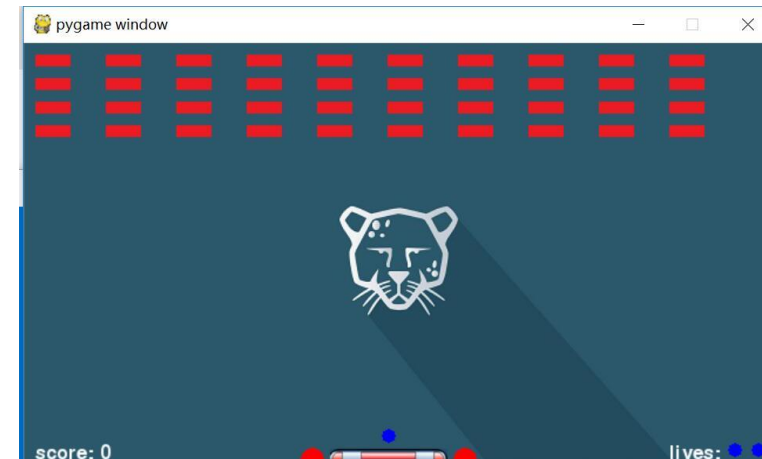
王智男：16级土木工程 文档

周家会：16级地理信息科学 ppt





# 功能演示





# 总体设计

一

总体思路

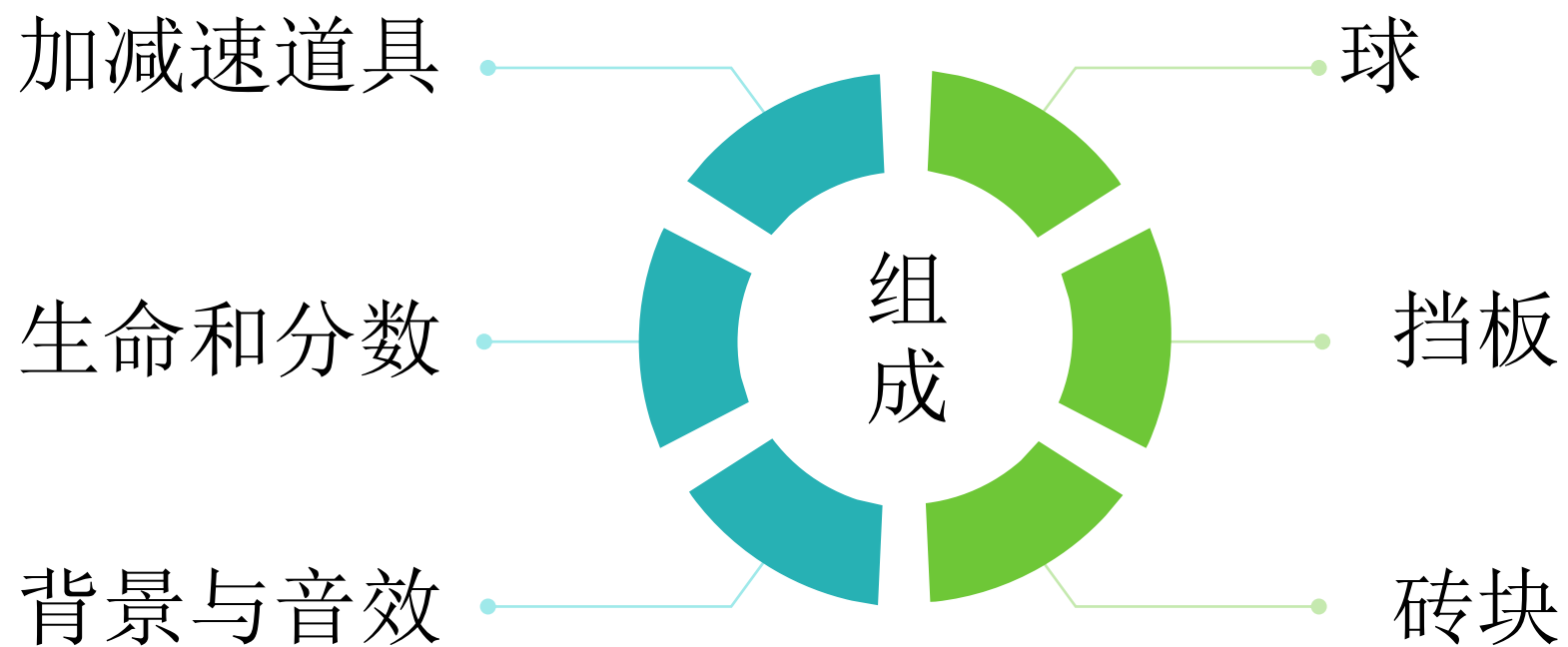
二

主要技术难点

三

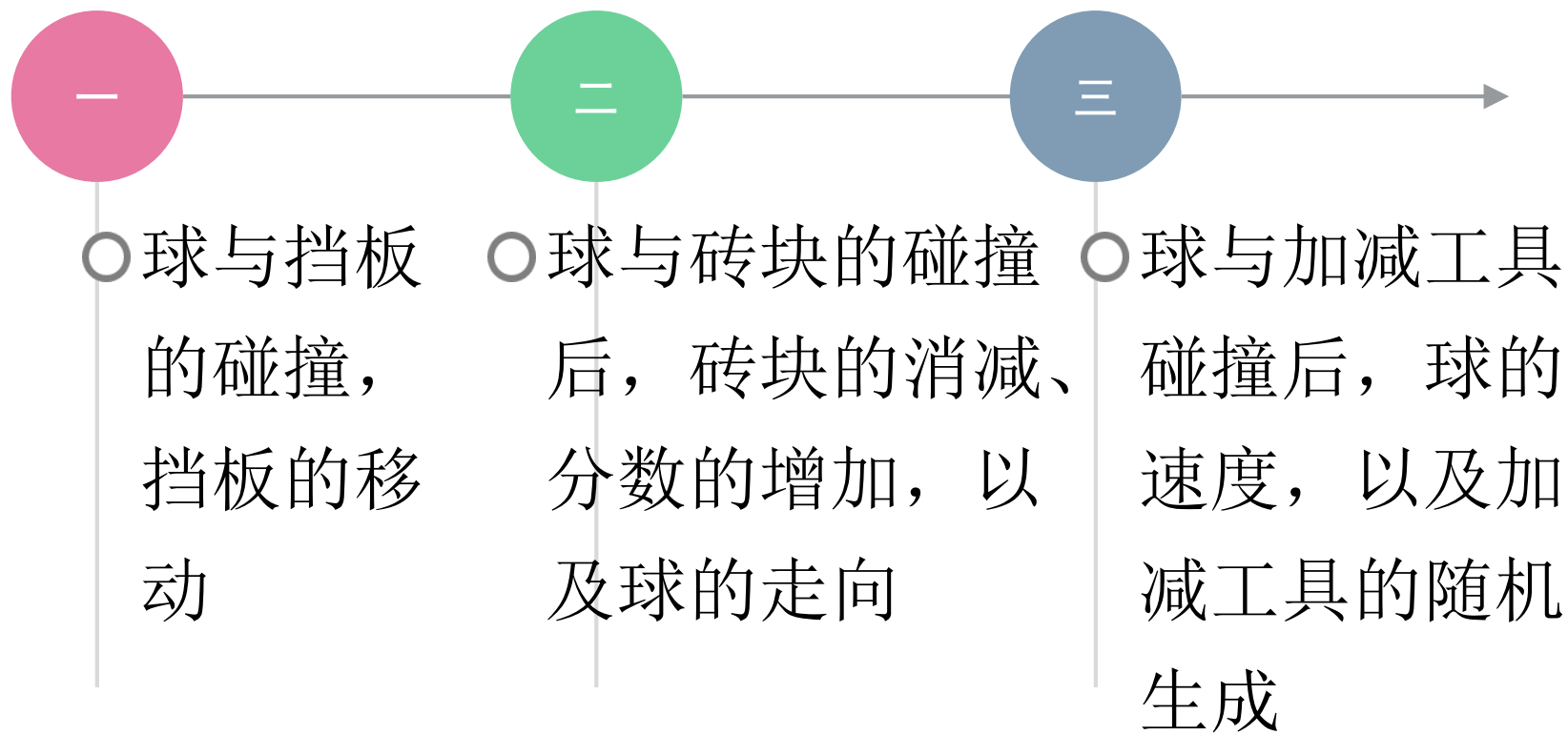
解决方案

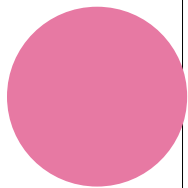
# 总体思路



# 总体思路

## 动画及相关关系





# 主要技术难点

一



碰撞检测：物体以什么样的时速和方向碰到另一个物体以及碰撞后方位角的改变

二



道具随机生成：道具出现的位置和出现的种类的随机数思想



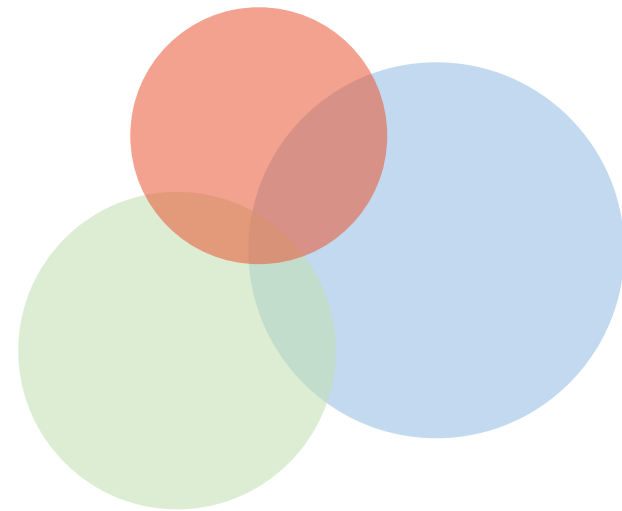


# 解决方案

```
# check hit
if pygame.sprite.spritecollide(my_ball, bricks, True):
    score += 1
    # my_ball.speed[1] = -my_ball.speed[1]
    my_ball.angle = math.pi * 2 - my_ball.angle
    hit_sound.play()

screen.blit(my_ball.image, my_ball.rect)
screen.blit(paddle.image, paddle.rect)
screen.blit(left_paddle_side.image, left_paddle_side.rect)
screen.blit(right_paddle_side.image, right_paddle_side.rect)
for brick in bricks:
    screen.blit(brick.image, brick.rect)
```

碰撞检测：  
利用公式，角度，和速率计算出反射速度

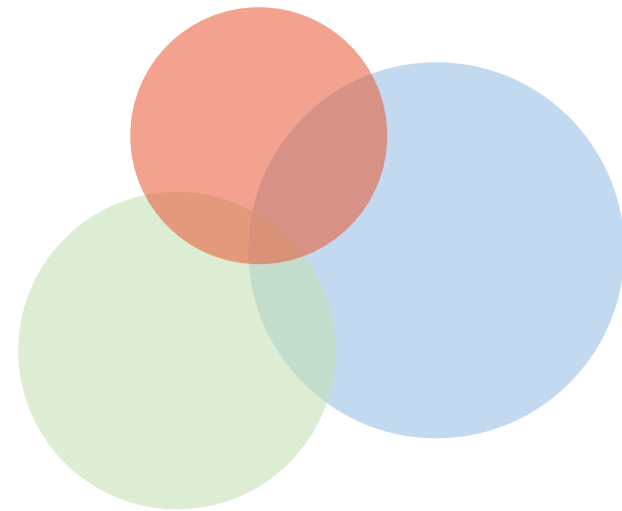




## 解决方案

```
def generate_item():  
    left = random.randint(0, 600)  
    choice = random.choice([1, 2])  
    if choice == 1:  
        item = SpeedUp((left, 0))  
        items.add(item)  
    else:  
        item = SlowDown((left, 0))  
        items.add(item)  
  
def slow_down():  
    my_ball.speed = my_ball.speed / 1.5
```

道具随机生成：用两个判断语句实现道具两个种类的随机生成





# 特色和创新点

一

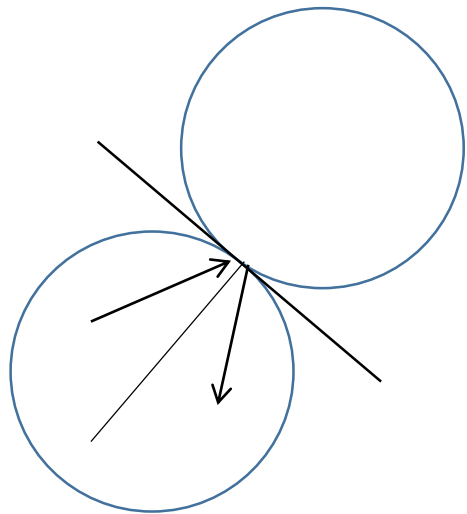
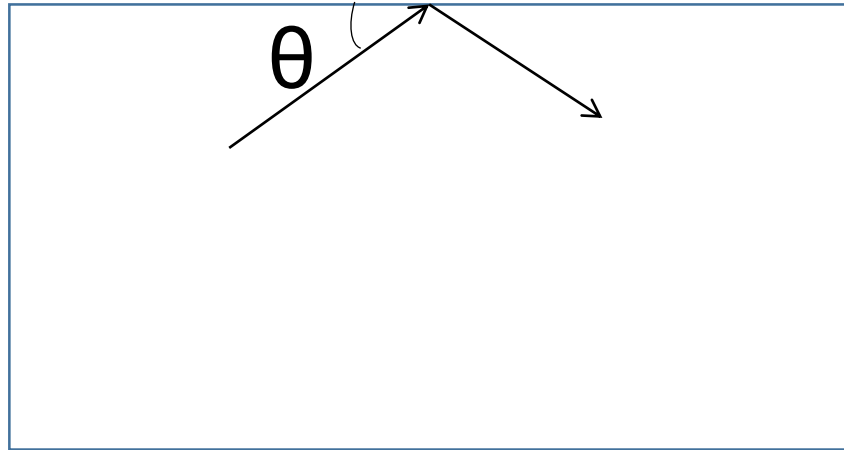
方位角

二

加减速道具



# 特色和创新点



方

位

角

- 原本程序用X和Y的值，我们改为用角的正弦值，不用考虑速度变化比起X和Y坐标更易于确定方向和编写，尤其便于确定圆碰撞后的方向

# 特色和创新点



```
# detect hit paddle_side
```

```
side = pygame.sprite.spritecollide(my_ball, paddle_sides, False, collided=pygame.sprite.collide_circle)
```

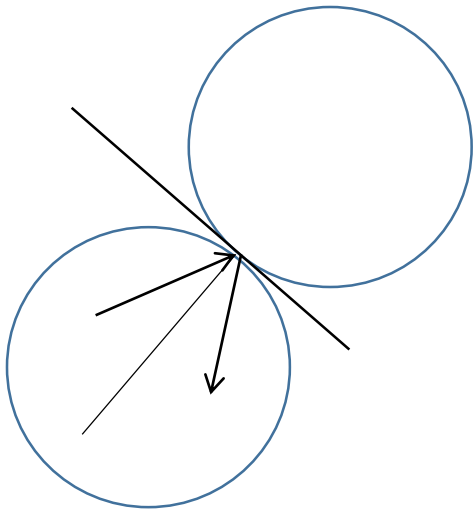
```
if side:
```

```
    side = side[0]
```

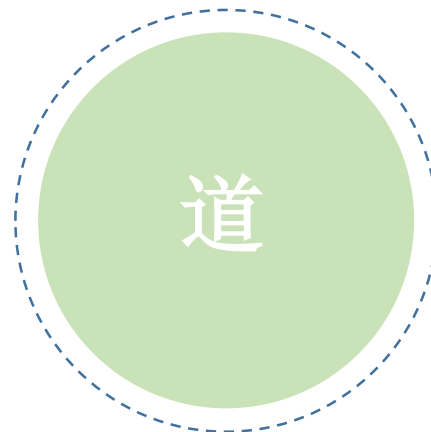
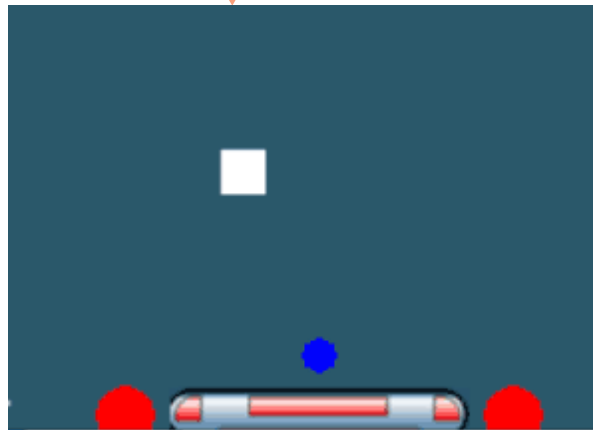
```
    collision_angle = math.acos((side.rect.centerx - my_ball.rect.centerx) / (side.radius + my_ball.radius))
```

```
    my_ball.angle = collision_angle * 2 - my_ball.angle + math.pi
```

```
    hit_sound.play()
```



# 特色与创新点



这个点是原本没有，我们小组想出来的

由黑白两种砖块随机生成，白色为减速，黑色为加速，增加玩法，使游戏形式更丰富

蟹蟹观赏！