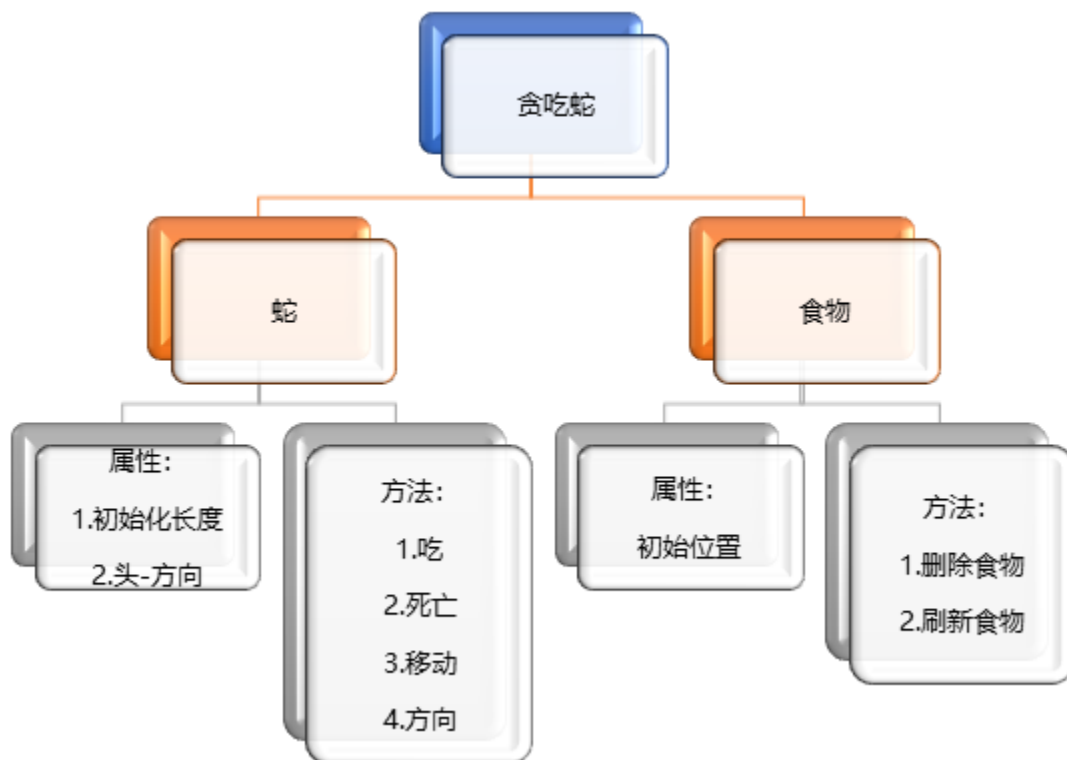


一、项目涉及的对象划分：



Pygame 的坐标系是以左上角为 (0, 0) 点，往右方向为 X 轴，往下方向为 Y 轴，单位为像素。

导入库和全局变量：

```
import sys
import random
import pygame
from pygame.locals import *

#全局变量
SCREEN_X = 600
SCREEN_Y = 600
BLACK = pygame.Color(0,0,0)
WHITE = pygame.Color(255,255,255)
RED = pygame.Color(255,0,0)
GREY = pygame.Color(150,150,150)
LIMEGREEN = pygame.Color(50,205,50)
class Snake(object):
```

蛇类：

```

class Snake(object):
    # 属性:
    # 1. 初始化长度
    # 2. 头-方向
    def __init__(self):
        self.direction = pygame.K_RIGHT
        self.body = []
        for i in range(5):
            self.addnode()

    def addnode(self):
        left,top = (0,0)
        if self.body:
            left,top = (self.body[0].left,self.body[0].top)
        #
        pygame.Rect(left, top, width, height)
        node = pygame.Rect(left,top,25,25)
        if self.direction == pygame.K_LEFT:
            node.left -= 25
        elif self.direction == pygame.K_RIGHT:
            node.left += 25
        elif self.direction == pygame.K_UP:
            node.top -= 25
        elif self.direction == pygame.K_DOWN:
            node.top += 25
        self.body.insert(0, node)

    def delnode(self):
        self.body.pop()

#死亡判断
    def isdead(self):
        #撞边界
        if self.body[0].x not in range(SCREEN_X):
            return True
        if self.body[0].y not in range(SCREEN_Y):
            return True
        #撞自己
        if self.body[0] in self.body[1:]:
            return True
        return False

#移动
    def move(self):
        self.addnode()
        self.delnode()

#改变方向
    def changedirection(self,curkey):
        LR = [pygame.K_LEFT,pygame.K_RIGHT]
        UD = [pygame.K_UP,pygame.K_DOWN]
        if curkey in LR+UD:
            #蛇是横向的, 方向也是左右就不需要改变
            if (curkey in LR) and (self.direction in LR):
                return
            #蛇是纵向的, 方向也是上下就不需要改变
            if (curkey in UD) and (self.direction in UD):
                return
            #不满足以上情况就根据键盘方向进行改变
            self.direction = curkey

```

食物类:

```

class Food(object):
    def __init__(self):
        self.rect = pygame.Rect(-25,0,25,25)
        #删除食物
    def remove(self):
        self.rect.x = -25
        #刷新食物
    def set(self):
        if self.rect.x == -25:
            allpos = []
            #产生食物的范围
            for pos in range(25,SCREEN_X-25,25):
                allpos.append(pos)
            self.rect.left = random.choice(allpos)
            self.rect.top = random.choice(allpos)
            print(self.rect)

```

格式函数:

```

#定义一个函数控制显示内容
def show_text(screen,pos,text,color,font_bold=False,font_size=50,font_italic=False):
    cur_font = pygame.font.SysFont("微软雅黑",font_size)
    cur_font.set_bold(font_bold)
    cur_font.set_italic(font_italic)
    text_fmt = cur_font.render(text,1,color)
    #绘制文字
    screen.blit(text_fmt,pos)

```

主函数:

```

def main():
    pygame.init()
    screen_size = (SCREEN_X,SCREEN_Y)
    screen = pygame.display.set_mode(screen_size)
    pygame.display.set_caption("贪吃蛇")
    clock = pygame.time.Clock()
    scores = 0
    isdead = False

    snake = Snake()
    food = Food()

    while True:
        for event in pygame.event.get():
            # 监听事件, 如果点击关闭按钮就退出
            if event.type == pygame.QUIT:
                sys.exit()
            # 如果点击键盘, 就监听键盘事件进行操作
            if event.type == pygame.KEYDOWN:
                snake.changedirection(event.key)
                if event.key == pygame.K_SPACE and isdead:
                    return main()
        screen.fill(WHITE)

```

```
#画蛇身体
if not isdead:
    snake.move()
for rect in snake.body:
    pygame.draw.rect(screen,LIMEGREEN,rect,0)
#蛇死亡
isdead = snake.isdead()

if isdead:
    show_text(screen,(100,200),"YOU LOSE!",RED,False,100)
    show_text(screen,(150,260),"press space ,try again!",BLACK,False,30)
#回到食物
if food.rect == snake.body[0]:
    scores += 10
    food.remove()
    snake.addnode()

food.set()
pygame.draw.rect(screen,(136,0,24),food.rect,0)

show_text(screen,(50,500),"Scores:"+str(scores),GREY)

pygame.display.update()
clock.tick(10)
```

```
main()
```