一 环境准备

（一） 第三方包pygame安装

1 在命令行窗口中，安装命令: `pip install 模块名/包名`

```
pip install pygame
```

2 安装过程中出现TimeoutError 解决方法：更换国内镜像源

- **临时更改**：`pip install 模块名/包名 -i 镜像源地址`

```
pip install pygame -i https://pypi.tuna.tsinghua.edu.cn/simple
```

- **永久更改**

  在user目录中创建一个pip目录，如：`C:\Users\Administrator\pip`，新建文件 `pip.ini` 内容

  ```
  [global]
  index-url = https://pypi.tuna.tsinghua.edu.cn/simple
  ```

  或是

  ```
  pip install pip -U   # 升级pip工具
  pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
  ```

（二）开发环境：pycharm

二 pygame基本使用

（一） pygame中常用模块（了解）

模块名	功能
pygame.display	访问显示设备
pygame.draw	绘制形状、线和点
pygame.event	管理事件
pygame.font	使用字体
pygame.image	加载和存储图片
pygame.key	读取键盘按键
pygame.mixer	声音
pygame.mouse	鼠标
pygame.movie	播放视频
pygame.music	播放音频
pygame.rect	管理矩形区域
pygame.sprite	操作移动图像
pygame.surface	管理图像和屏幕
pygame.surfarray	管理点阵图像数据
pygame.time	管理时间和帧信息
pygame.transform	缩放和移动图像

窗口方法

方法	描述
pygame.display.set_mode((窗口宽，窗口高))	创建窗口
pygame.display.update()	刷新界面
pygame.display.set_caption("窗口标题")	设置窗口标题

图像相关操作

方法	描述
pygame.image.load(path)	加载图片
window.blit(image, (0, 0))	将图片贴到指定位置
pygame.Rect.colliderect(rect1, rect2)	判断两个矩形是否相交

事件相关操作

方法	描述
event.type	事件类型
event.key	按键类型
event.get()	获取事件

（二）demo演示

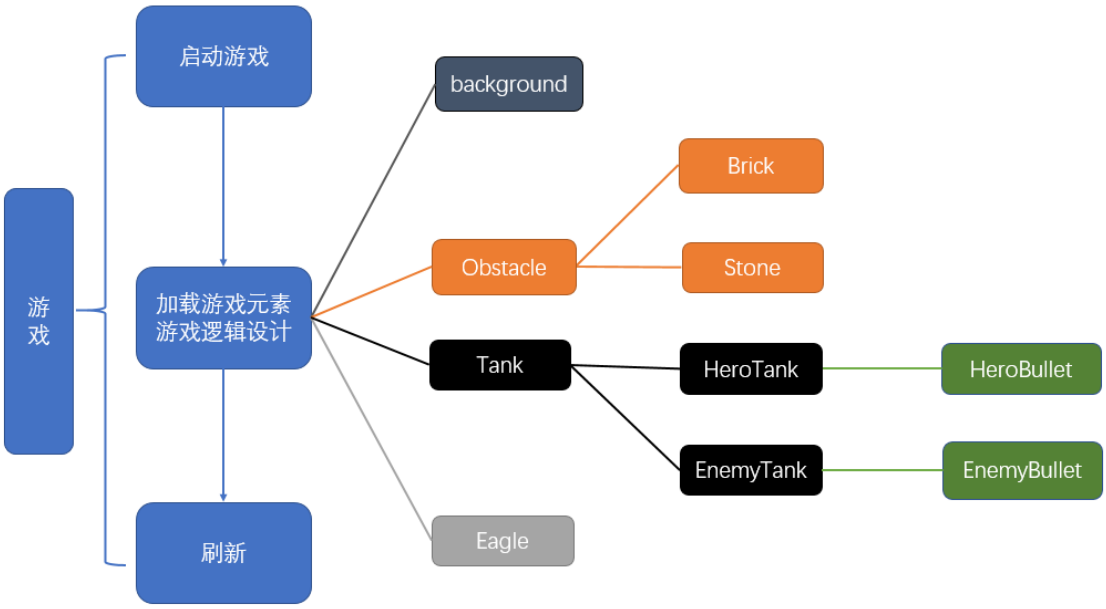
```
# 导入pygame库
import pygame


# 初始化pygame,为使用硬件做准备
pygame.init()

# 创建了一个窗口
screen = pygame.display.set_mode((640, 480), 0, 32)
# 设置窗口标题
pygame.display.set_caption("HELLO WORLD")

# 加载并转换图像
background = pygame.image.load('./images/bg.jpg')
tank = pygame.image.load('./images/tugai.net.20101117235923.gif')

# 定义坦克初始位置
x = 100
y = 200

while True:
    # 游戏主循环
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            # 接收到退出事件后退出程序
            exit()

    # 将背景图画上去
    screen.blit(background, (0, 0))

    # 鼠标控制坦克
    # 获得鼠标位置
    # x, y = pygame.mouse.get_pos()
    # 计算光标的左上角位置
    # x -= tank.get_width() / 2
    # y -= tank.get_height() / 2

    # 键盘控制坦克
    # 键盘长按事件
    pressed_keys = pygame.key.get_pressed()

    if pressed_keys[pygame.K_a]:
        x -= 4
    if pressed_keys[pygame.K_d]:
        x += 4
    if pressed_keys[pygame.K_w] or pressed_keys[pygame.K_UP]:
        y -= 4
```

```
    if pressed_keys[pygame.K_s] or pressed_keys[pygame.K_DOWN]:
        y += 4

    # 把坦克画上去
    screen.blit(tank, (x, y))

    # 刷新一下画面
    pygame.display.update()
```

三 项目分析和设计

（一）项目核心功能分析，抽象类

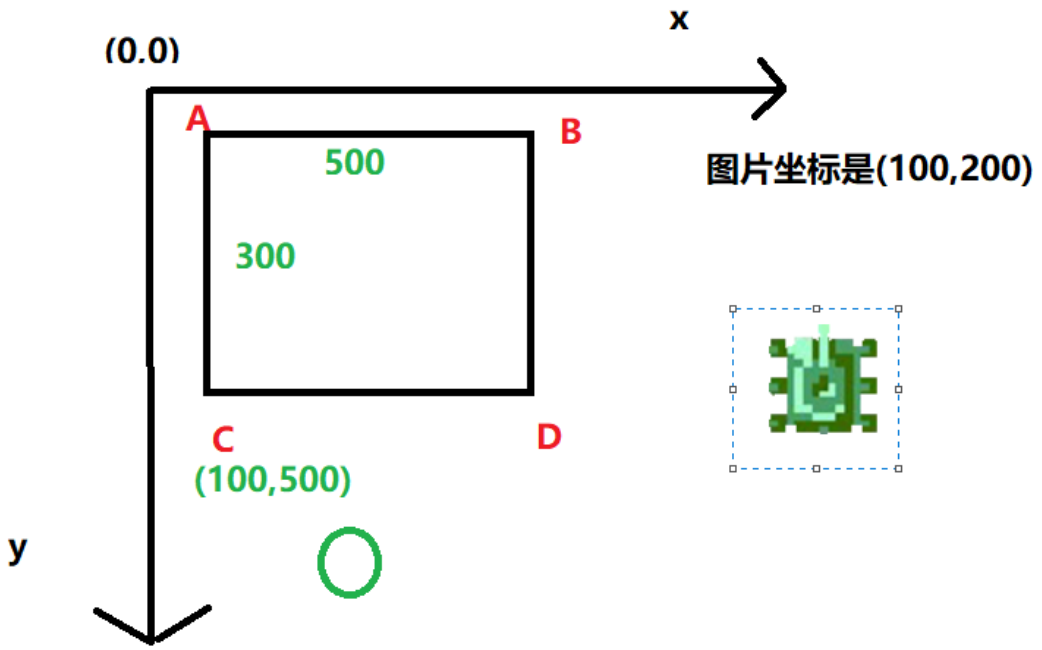
- **游戏类(Game)----游戏的整体逻辑**
  - 加载各种元素
  - 刷新
- **游戏中的元素**
  - 背景（background）
  - **坦克(BaseTank)**
    - 我方坦克(HeroTank)
    - 敌方坦克(EnemyTank)
  - **子弹(BaseBullet)**
    - 我方子弹(HeroBullet)
    - 敌方子弹(EnemyBullet)
  - **障碍物(Obstacle)**
    - 砖块墙(Brick)
    - 石头墙(Stone)
  - **老鹰(Eagle)**



四 功能开发

1 游戏界面和背景的加载

```
import pygame

pygame.init()

WINDOW_W = 1200
WINDOW_H = 720
WINDOW = pygame.display.set_mode((WINDOW_W,WINDOW_H))

class Game:
    """游戏类：负责总体逻辑"""
    def __init__(self):
        # 加载背景图
        self.bg = pygame.image.load('./images/bg.jpg')
        # 设置窗口标题
        pygame.display.set_caption("坦克大战")

        # 设置界面图标
            #1加载图片
        image = pygame.image.load('images/tugai.net.20101118000029.gif')
            #2设置图标
        pygame.display.set_icon(image)


    def draw(self):
        # 贴背景图
        WINDOW.blit(self.bg,(0,0))

    def update(self):
        pygame.display.update()

    def run(self):
        """游戏的启动方法"""
        while True:
            # 贴图
            self.draw()
            # 刷新
            self.update()


if __name__ == '__main__':
    game = Game()
    game.run()
```



2 坦克基本设计

1 坦克基类设计

定义坦克的属性和方法

- 属性：坐标，图片，移动速度，初始方向,血量
- 方法：移动和发射子弹

```
class BaseTank:
    def __init__(self,x,y):
        """坦克初始配置"""
        self.speed = 3  # 默认坦克移动速度
        self.hp = 2  # 默认血量

        # 不同方向的坦克图片
        self.images = {
            'U': pygame.image.load('images/p1tankU.gif'),
            'D': pygame.image.load('images/p1tankD.gif'),
            'L': pygame.image.load('images/p1tankL.gif'),
            'R': pygame.image.load('images/p1tankR.gif'),
        }

        self.direction = 'D'  # 初始方向
        self.image = self.images[self.direction]  # 获取对应图片
        self.rect = self.image.get_rect()  # 获取rectangle属性（x,y,w,h）
        self.rect.x = x  # 更新图片的坐标
        self.rect.y = y  # 更新图片的坐标

    def move(self):
        """移动"""
        if self.direction == 'U':
            self.rect.y -= self.speed if self.rect.y > 0 else 0
        if self.direction == 'D':
            self.rect.y += self.speed if self.rect.y < WINDOW_H -
self.rect.height else 0
        if self.direction == 'L':
```

```
            self.rect.x -= self.speed if self.rect.x > 0 else 0
        if self.direction == 'R':
            self.rect.x += self.speed if self.rect.x < WINDOW_W -
self.rect.width else 0

    def fire(self):
        raise NotImplementedError("子类必须重写该方法")
```