

中国科学技术大学



计算机程序设计课程实践项目报告 USTC 求生——Survive in USTC

作者姓名：闫泽轩(PB21111696) 郑逸飞(PB21111675)

导师姓名：李卫海老师 孙广中老师

完成时间：二零二一年十二月二十一日

摘 要

《USTC 求生》（英文名 Survive in USTC）是为 2021 年秋季学期计算机程序设计 A 课程的大作业要求而创作的一款双人联机跑酷游戏。本文是该游戏项目的实验报告。

本项目是一款基于 Python 和其扩展库 Pygame，以科大学习生活为背景的跑酷游戏。该项目的一大特点在于支持线上双人联机，通过利用 TCP 协议进行 Socket 编程实现。同时，本项目中除了人物的所有美术素材均为原创，使用 Pixel Studio 进行绘制。该游戏操作易于上手，同时具有适中的难度，并融入了许多科大元素。在锻炼编程能力的同时，也展现出我们对科大的优良传统和校园文化的继承与发扬。

目录

一、项目介绍

1. 编程语言
2. 游戏背景
3. 美术特色
4. 联机特色
5. 玩法特色

二、环境依赖

1. 硬件要求
2. 软件要求
3. 游戏实力要求

三、功能和展示

1. 联网功能
2. 游戏操作
3. 其他核心功能

四、总结

一、项目介绍

1.编程语言

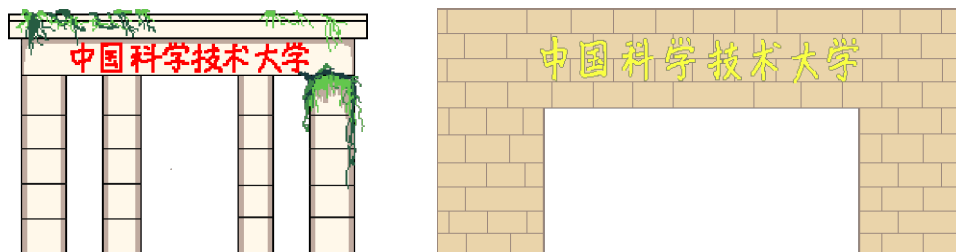
本项目使用了 Python 和其扩展库 Pygame 进行编程。作为时下最热门的编程语言之一，Python 以其简洁性、易读性以及可拓展性而深受喜爱；近年使用 Python 编写的游戏也日益增加。同时，Pygame 作为专门面向游戏编程的扩展库，在我们的工作过程中提供了许多游戏逻辑和关卡编辑上的便利。本游戏的动态也主要依赖 Pygame 中精灵组(sprite)的层(layer)进行编写。在实践中，我们不仅收获了 Python 语言的相关知识，更锻炼了将来学习与工作所需的调试能力和项目规划能力，不可谓收获不大。

2. 游戏背景

《USTC 求生》以科大学生为了获得更高的 GPA 而不断奔跑为背景，展现了科大学生在学习生活中的拼搏奋进。同时，双人联机对战更突出了学子们的良性竞争，共同奋进。在本项目中，我们融入了诸如“理实交融”、“GPA4.3”、“淑芬”等等能引起科大学生强烈共鸣的元素。我想，每个科大学子都能在这个项目中找到自己青春靓丽的身姿。

3. 美术特色

本项目中除了人物的所有美术素材均为原创，使用 Pixel Studio 进行绘制。一系列素材均保持了统一的像素风格，力求画面明亮和谐。同时，我们在绘制素材的过程中，在融入科大元素上注入了许多心血，如下列素材：



科大的两个大门

同时，我们也从校园文化中汲取了灵感，在背景中加入了许多科大人耳熟能详的语句和词汇：



在一些小元素的设计上，我们也下了不少功夫，让画面俏皮活泼：



特别的，我们还学习了逐帧动画的制作，这是一个 10 帧的死神：



本作在美术上找到了自己的风格，最终的成品也达到了我们的预期。

4. 联机特色

前文曾述，该项目的一大特点在于支持线上双人联机，通过利用 TCP 协议进行 Socket 编程实现。通过本功能，两位玩家可以实现在不同网络环境下的实时竞技，互相比拼谁能拿到更高的 GPA。该功能使得本游戏更富趣味性与刺激性，也展现出我们在 Python 网络编程上的精进。

5. 玩法特色

本游戏中，玩家所操作的角色会不断地向前奔跑来获取 GPA，一旦 GPA 达到了 4.3 则获胜，若 GPA 过低则为失败。如果碰撞到了身前的砖块，空中飞行的“数分”炸弹、“线代”火焰、“程设”导弹，或者被身后的死神追上，都会导致玩家减速，同时还会带来扣除 GPA 的惩罚。本游戏还设有能量槽，活用“躺平”和“冲刺”功能可以帮助玩家躲避障碍和攻击。一旦能量槽积攒满了，则可以开启“无双”模式，进入充满 GPA 的地图并可无限跳跃。地图上还有随机生成的跑鞋和医疗包道具，分别能够加速和回复能量槽。总体上讲，本游戏难度适中而又不失挑战性，在传统的跑酷玩法上实现了一定创新，丰富的功能和元素以及随机的地图也给每一次的体验带来了新鲜感和趣味性。

二、环境依赖

1.硬件要求

由于本项目采用了平面像素风格，且全程使用 Python 进行编程而未采用复杂的大型引擎，所以本项目对硬件性能的要求相当的低，近年的硬件无问题的电脑都可以十分流畅地运行。

2.软件要求

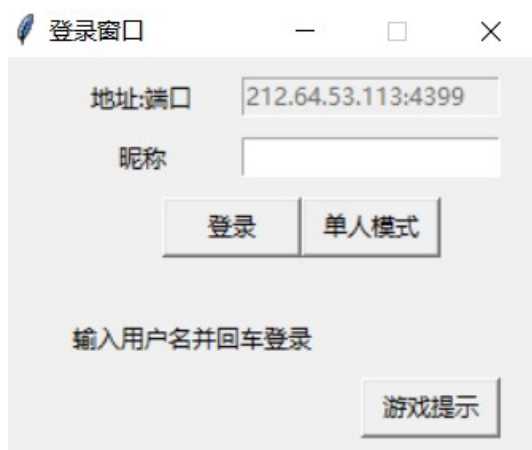
本项目暂时仅能在 Windows 系统上运行，但由于本项目的封装较为完善，在测试用的电脑上均无需安装额外的库。若运行出现问题，请尝试安装 Python3 环境及 Pygame 扩展库。同时，由于本游戏支持在线双人联机的特性，请确保进行联机对战游戏时电脑连接网络。

3. 游戏实力要求

本项目的小组成员之一在玩这个游戏的时候多次被其他测试人员碾压。由于人物的奔跑速度较快且道具较多较复杂，本游戏对于不擅长跑酷游戏的人还是具有相当的难度的，如果想要获得较好的游戏体验，可以多加练习，或者找本项目的小组成员中不大会玩游戏的那一个(郑逸飞)进行对战来获得满足。

三、功能和展示

1.联网功能



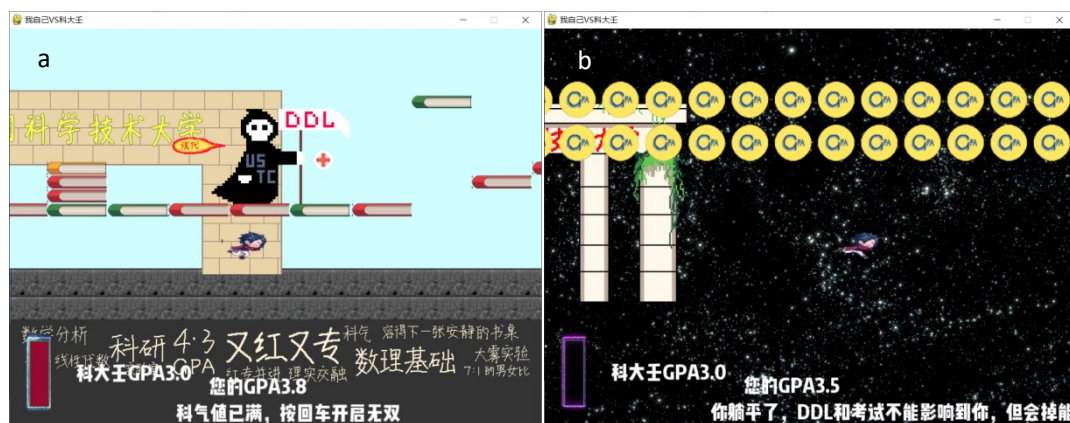
游戏双方输入自己的昵称并进行登录，即可在服务器上实现匹配。本功能通过 TCP 编程实现。在游戏过程，服务器每 1.0 秒进行一次数据的收发，从而实现匹配双方数据的同步和实时更新。

本游戏联网交换所使用的数据被封装在了类中，其定义如下：

```
class Client(object):
    def __init__(self, nickname="未命名", oppose="对手", gpa=3.0, game_over=False, youwin=False, availability=True):
        self.nickname = nickname
        self.oppose = oppose
        self.gpa = gpa
        self.game_over = game_over
        self.youwin = youwin
        self.available = availability
```

当然，本游戏也支持单人离线游玩。

2.游戏操作



a)实际画面；b)无双模式

本游戏的玩法此处不再详述，其具体操作如下：

Space 键进行跳跃，

通过玩家类进行实现，其代码如下

```
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_SPACE:
        self.player.frame = 10
        self.player.first_frame = 10
        self.player.last_frame = 15
        if not self.player_jumping:

            self.player_jumping = True
            self.player.Y -= 5
            self.jump_vel = -10.0
            self.play_sound(self.jump_sound)
```

判断玩家按键时的操作

```
# 判断玩家跳跃
if self.player_jumping and not self.game_over:
    self.player.Y += self.jump_vel
    self.jump_vel += 0.5
# 玩家落地
if self.player.Y > player_start_y:
    # 判断滑步
    if not self.slide:
        self.player.first_frame = 1
        self.player.last_frame = 8
        self.player.Y = player_start_y
        if self.player_jumping:
            self.erduan += 1
        self.player_jumping = False
# 更新精灵组并绘图
if not self.game_over:
    if not self.isinsane:
        self.plat.block_group.update(ticks, 1)
    else:
        self.plat.gpa_group.update(ticks, 1)
    self.group.update(ticks, 100)
```

对玩家跳跃操作的处理

A 键进行“躺平”操作，D 键进行“冲刺”操作，当能量槽充满后，按 Enter

键进入无双模式。具体实现与跳跃类似。

3.其他的核心功能代码举例

```
# 判断火球与玩家碰撞,碰撞会加快DDL,并减少能量
if not self.slide and pygame.sprite.collide_rect_ratio(0.6)(self.arrow_linear, self.player):
    self.reset_arrow_linear()
    self.GPA -= 0.1
    ddlvel += 2
    self.player.X -= self.playerspeed
    self.miles -= self.playerspeed
    self.energy -= 3
if not self.slide and pygame.sprite.collide_rect(self.arrow_coding, self.player):
    self.reset_arrow_coding()
    self.GPA -= 0.1
    ddlvel += 2
    self.player.X -= self.playerspeed
    self.miles -= self.playerspeed
    self.energy -= 5
```

判断玩家与物体的碰撞及其后续操作

```
# 创建DDL
self.DDL = MySprite()
self.DDL.load("pic/DDL.png", 256, 256, 4)
self.DDL.position = 100, 230
self.DDL.first_frame_frame = 1
self.DDL.last_frame = 9
ddlvel = 3
self.group.add(self.DDL)
```

创建追随玩家的 DDL 死神

```
# 判断玩家踩踏砖块
collided_blocks = pygame.sprite.spritecollide(self.player, self.plat.block_group, False,
                                                collided=self.isdowncollided)
for collided_block in collided_blocks:
    if self.player_jumping and not self.slide:
        self.player.first_frame = 1
        self.player.last_frame = 8

self.erduan = 1
self.jump_vel = 0
self.player.Y -= abs(collided_block.rect.top - self.player.rect.bottom)
```

判断玩家踩踏砖块后的操作

四、总结

本项目由双人小组合作开发，基于 Python3 进行编程。本小组的两位成员都没有什么竞赛经验，对此类大型项目的开发也比较陌生。完成这样一个项目是对我们的自学能力和项目规划能力的一次极大的挑战。在工作期间，种种设计上的问题与编程过程中无数的 BUG 都让人倍感崩溃。对代码的反复调试早已成了家常便饭，但当每个 BUG 被修复后，亦或是看着游戏的功能一点一点添加，这种满足感是难以言说的。经过不懈努力后，最后的成品也达到了我们的心理预期。

这次实践中，我们收获了许多知识与经验。其中令我们感受最深的还是项目规划的重要性。如果事前没有一个完备的项目规划与框架，而是全凭感觉一步步搭建功能，将会带来一次次的重构和低下的效率。同时，我们也感受到一个统一的代码风格和代码习惯的重要性，让我们不必在自己都看不懂的代码中苦苦挣扎。

在此特别感谢两位老师和助教们的指导。老师们好厉害讲课好好，助教们好帅好漂亮。虽然上面这句话带有点调侃的口吻，但如果没有你们，我们一定不能顺利完成本项目。在计算机程序设计这门课程中收货的宝贵的编程能力和编程思想，一定会成为我们在将来的大学生活、乃至人生道路上的护航。

本项目参考资料：

[1][Python 语言参考手册 — Python 3.10.1 文档](#).

[2][Pygame 官方网站](#).