**Bradon 课程计划和反馈**

主题: 游戏编程

目标: 帮助Bradon完成游戏编程的项目并学习相关的知识和技能。[[游戏项目安排](https://docs.google.com/document/d/1zVtWqDz4yxomFM0v1EehclM0uCZNa8JgKAnpYbQHj4E/edit#heading=h.8m6vb0n75osf)]

课程时间：2023年11月22日

课程内容: 这节课Bradon 主要继续完成个人网站GitHub的项目上传

课程反馈:

* 根据11月04日课程中所提供的指导步骤和链接，Bradon 已经基本完成了个人网站的创建

课程作业:

1. 给Nina老师发GitHub的链接

课程时间：2023年11月18日

课程内容: 这节课继续插件项目的开发，主要讲解了处理弹出页面点击事件以及处理文本转语音脚本的伪代码，和使用 Google Text-to-Speech API 的方法

课程反馈:

* 创建一个popup.js文件，处理弹出界面按钮的点击事件 [javascript [伪代码](https://docs.google.com/document/d/1wMJnodAqh7OPqLpVC5TAJDdD-Cs6jAwCISry1T3tTDU/edit#bookmark=id.juin376z0df7)]
* 创建一个content.js文件，作为内容脚本，用于在页面中执行屏幕阅读器的功能 [javascript [伪代码](https://docs.google.com/document/d/1wMJnodAqh7OPqLpVC5TAJDdD-Cs6jAwCISry1T3tTDU/edit#bookmark=id.e41667uf4vgc)]
  + 编写脚本读取页面文本内容并存为 .txt 文件
  + 使用Google Text-to-Speech API [[链接](https://cloud.google.com/text-to-speech?utm_source=google&utm_medium=cpc&utm_campaign=na-US-all-en-dr-skws-all-all-trial-b-dr-1605212&utm_content=text-ad-none-any-DEV_c-CRE_673766089147-ADGP_Hybrid+%7C+SKWS+-+MIX+%7C+Txt_Text-to-Speech-KWID_43700077224548055-kwd-10350000&utm_term=KW_text%20to%20speech-ST_text+to+speech&gad_source=5&gclid=Cj0KCQiA3uGqBhDdARIsAFeJ5r2_imFP_UAxdJK3Yghk7JdElYIwkzj24TW9ndfr8BUeVdwQzMqSbYUaAvg7EALw_wcB&gclsrc=aw.ds)] 将文本转成语音，使用该API 的步骤如下
* 创建一个Google Cloud 试用账户 [[链接](https://console.cloud.google.com/freetrial/signup/)]
* 开启 Cloud text-to-speech API
  + 在Goolge Cloud里，选择 ‘APIs & Services’ 然后选择 ‘Library’
* 创建一个Google Service 账户 [[链接](https://cloud.google.com/iam/docs/service-accounts-create)]
* 访问Google Cloud shell
  + 在Goolge Cloud里，点击左上角的汉堡菜单，选择 Cloud Shell
  + 在设置里，勾选 ‘Enable Cloud Shell’
  + 点击控制栏上方的 ‘Activate Cloud Shell’, 这样弹出一个新的 ‘Cloud Shell’
* 在Google Cloud shell 里测试Google test-to-speech REST 服务
  + 将项目ID存为一个环境变量
  + 对该项目开启 test-to-speech api：运行如下指令 ‘gcloud services enable texttospeech.googleapis.com’
* 将文本编码为语音文件
  + 在与 synthesize-text.txt 文件相同的目录中运行以下命令，将来自文本转语音 API 的结果解码为 MP3 音频文件

课程作业:

1. 完成读取页面文本内容的脚本编写

课程时间：2023年11月04日

课程内容:

* 这节课包含两个内容
  + 环境设置和一部分的插件代码编写
  + 注册GitHub账户并上传本地的项目

课程反馈:

* 开发屏幕阅读器浏览器插件 [[开发流程](https://docs.google.com/document/d/1wMJnodAqh7OPqLpVC5TAJDdD-Cs6jAwCISry1T3tTDU/edit?usp=sharing)]
  + 环境设置：Chrome 浏览器和VS Code 代码编辑器
  + 编写插件代码
    - 创建一个新文件夹，命名为"ScreenReaderExtension"
    - 在文件夹中创建一个manifest.json文件 (伪代码见[开发流程](https://docs.google.com/document/d/1wMJnodAqh7OPqLpVC5TAJDdD-Cs6jAwCISry1T3tTDU/edit?usp=sharing))，其中包含插件的基本信息和权限声明
    - 在同一文件夹中创建一个popup.html文件 (伪代码见[开发流程](https://docs.google.com/document/d/1wMJnodAqh7OPqLpVC5TAJDdD-Cs6jAwCISry1T3tTDU/edit?usp=sharing))，用于插件的弹出界面，内容可以是一个简单的按钮
* 注册GitHub 账户并学习如何上传/同步本地项目文件
  + 注册GitHub 账户 [[链接](https://docs.github.com/en/get-started/signing-up-for-github/signing-up-for-a-new-github-account)]
  + 下载GitHub 电脑客户端 [[链接](https://docs.github.com/en/desktop/installing-and-authenticating-to-github-desktop/setting-up-github-desktop)]
  + 用GitHub电脑客户端创建代码库 (repository) [[链接](https://docs.github.com/en/desktop/overview/creating-your-first-repository-using-github-desktop)]
  + 向代码库中添加项目文件 [[链接](https://docs.github.com/en/issues/planning-and-tracking-with-projects/managing-your-project/adding-your-project-to-a-repository)]

课程作业:

1. 完成GitHub账户注册，并上传之前完成的游戏项目

课程时间：2023年10月28日

课程内容:

* 课上Bradon完成了第一人称射击教程【[6](https://youtu.be/BLWWw49c7kU?si=CpjD5NYIkKuOCfpd)】搜索状态的开发。课程后半部分我介绍了开发屏幕阅读浏览器插件的总体流程，着重展开了第一部分‘需求分析’

课程反馈:

* 截止目前，第一人称射击教程只更新到系列6，完成教程【[6](https://youtu.be/BLWWw49c7kU?si=CpjD5NYIkKuOCfpd)】后，该游戏项目将暂告一段落直到新的系列更新
* 开发屏幕阅读浏览器插件的主要步骤 [[思维导图](https://excalidraw.com/#json=Um8T1xmcAiBnqUAjH_Iif,yn9bQ5Vw6DejYo0gn0XuiQ)]
  + 需求分析
    - 目标受众：视障患者
    - 插件的功能：协助视障患者浏览网页，将网页上的文本内容转化成语音播放
    - 用户的辅助技术要求：语音指令
  + 环境设置
  + 编写插件代码
  + 文本转语音
  + 图标设计
  + 调试和测试
  + 发布插件

课程作业:

1. 下节课将开始做‘环境设置’和‘编写插件代码’。由于插件使用JavaScript编写并且适配于Chrome，请确保电脑已经安装好Chrome浏览器和VS Code 代码编辑器

课程时间：2023年10月20日

课程内容:

* 课上Bradon观看了第一人称射击教程【[6](https://youtu.be/BLWWw49c7kU?si=CpjD5NYIkKuOCfpd)】有关用有限状态机编辑敌人的搜索状态（Game AI – search state），并完成了一部分。继上节课探讨了结合公益和软件开发的项目，初步确定了要做的项目

课程反馈:

* 射击教程【[6](https://youtu.be/BLWWw49c7kU?si=CpjD5NYIkKuOCfpd)】关于编辑敌人搜索状态的要点
  + “搜索状态“的目标：当敌人失去玩家的视线时，敌人会前往玩家最后被发现的位置，在短暂的搜索后切换为其他状态
  + 创建变量，包括“最后已知位置“和”搜索计时器“，用于跟踪玩家的最后已知位置和搜索持续时间
  + 在“搜索状态“中实施逻辑，包括根据条件设置目的地以及基于条件切换到”攻击状态“或”巡逻状态“
  + 创建“搜索状态“脚本，并从”基础状态“继承
* 和Bradon继续探讨了结合公益和软件开发的项目
  + 开发在线辅助工具以帮助视障或听障人士更容易使用互联网，如文本到语音转换的浏览器插件等，以下是一些相关的插件主题以供参考
    - 屏幕阅读器
    - 文本转语音（text-to-speech, TTS）插件
    - 字幕和转录工具
    - 手语浏览器扩展
    - 放大和对比度工具
    - 语音命令和语音搜索

课程作业:

1. 如有时间完成射击教程【[6](https://youtu.be/BLWWw49c7kU?si=CpjD5NYIkKuOCfpd)】剩余的内容
2. 在提供的插件/扩展主题内选择感兴趣的项目，下节课制定项目计划

课程时间：2023年10月14日

课程内容:

* 课上Bradon完成第一人称射击教程【[5](https://youtu.be/wNdi5hc0anY?si=A-FIXFKLIB4QDyR0)】有关战斗/攻击有限状态机的复现，并且对接下来开发辅助视障患者读谱的软件进行了讨论

课程反馈:

* 这节课Bradon根据教程完成了第二个有限状态机 -敌人NPC的战斗和攻击状态 - 的编写。
* 课程后半段和Bradon讨论了开发辅助视障患者读谱软件的想法
  + 出发点：有一些视障患者有学习乐器的需求，但由于他们视力受限，目前需要依赖志愿者来为他们读谱。然而这一过程可以通过开发软件来自动化
  + 益处：因为该项目将公益和软件开发结合在一起，有可能为留学申请提供一个有益的项目经验
  + Bradon的想法：似乎对这个项目没有浓厚的兴趣
* 一些结合公益和软件开发的项目想法，以供参考
  + 在线辅助工具：开发在线工具，如文本到语音转换或可访问性插件，以帮助视障或听障人士更容易使用互联网
  + 志愿者协调平台：建立一个平台，使志愿者更容易找到并参与支持社区，慈善机构和非盈利组织的活动

课程作业:

1. 可以再仔细考虑下是否想做该项目，如果不想的话，是否对以上列出的项目或其他项目感兴趣

课程时间：2023年10月06日

课程内容:

* 课上Bradon观看了第一人称射击教程【[5](https://youtu.be/wNdi5hc0anY?si=A-FIXFKLIB4QDyR0)】，其内容是用有限状态机编辑游戏人工智能中的战斗/攻击行为（Game AI – Combat/Attack）

课程反馈:

这节课的内容是开始编写第二个有限状态机，用于控制敌人NPC的战斗和攻击状态。Bradon在课上了观看了相关教程【[5](https://youtu.be/wNdi5hc0anY?si=A-FIXFKLIB4QDyR0)】，以下是该教程的要点：

* 处理状态转换 (handling state transitions)：
  + 检查玩家可见性 (checking player visibility)：通过代码确定玩家是否在敌人的范围内并在其视野内。这涉及计算到玩家的距离，敌人正前方与指向玩家的向量间的角度等
  + 使用射线投射 (using racycasting)：从敌人的眼睛高度向玩家投射一条射线，如没有受阻地命中玩家，即确认玩家可见
  + 状态转换逻辑 (state transition logic)：基于可见性检查来确定敌人是否应该从当前状态（如巡逻）转化到攻击状态
* 组织敌人游戏对象 (organizing enemy game object)
  + 层次结果 (hierarchy structure)：敌人的根游戏对象应主要包含逻辑组件（脚本），而不是可视元素。所有可视元素应作为该根对象的子对象进行组织
  + 正确的继承关系 (correct parenting)：演示了如何通过创建一个新的游戏对象（如”Cuba”）来正确构建层次结构，并将逻辑组件和可视元素至于其下
  + 一致的缩放 (consistent scaling)：确保根游戏对象具有统一的缩放

课程作业:

1. 复现一部分的教程【[5](https://youtu.be/wNdi5hc0anY?si=A-FIXFKLIB4QDyR0)】，其余部分将在下节课上完成

课程时间：2023年09月22日

课程内容:

* 课上Bradon复现了第一人称射击教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】的内容，即编辑游戏人工智能中的 巡逻（Game AI - Patrolling）

课程反馈:

这节课Bradon复现了教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】中运用有限状态机编辑敌人NPC的巡逻状态。以下是关于敌人有限状态机的要点：

* 敌人有限状态机 (enemy FSM)：
  + 一个基础状态 (base state) 和5个子状态
    - 基础状态: 用来中心化处理子状态间的转换
      * 2个类: 敌人 (Enemy) 和 状态机 (StateMachine)
      * 3个函数: 进入 (Enter())，操作 (Perform())，退出 (Exit())
    - 子状态：**巡逻 (patrol)**，攻击(attack)，逃跑 (flee)，躲藏 (heal), 搜寻 (search)
      * 巡逻状态
        + 包含基础状态的3个函数以及独有的巡逻循环函数 (PatrolCycle())
        + 2个变量: 航点索引 (waypointindex) 和 等待时间 (waitTime)
  + 每个子状态有各自独立的脚本编辑其特征

课程作业:

1. 下节课学习编辑Game AI的另一个状态 – 攻击，有时间可以课前预览下教程【[5](https://youtu.be/wNdi5hc0anY?si=aaLbt6-OMjRmHpMj)】

课程时间：2023年09月15日

课程内容:

* 课上Bradon 观看并学习了第一人称射击教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】，其内容涵盖游戏人工智能中的 巡逻（Game AI - Patrolling）

课程反馈:

这节课学习游戏人工智能，即为游戏中非玩家角色（NPCs）编辑移动行为。视频介绍了游戏编程中一种常用的计算模型 – 有限状态机 ([Finite State Machine](https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c#:~:text=A%20Finite%20State%20Machine%20is%20any%20device%20storing%20the%20state,Science%20called%20%E2%80%9Cautomata%20theory%E2%80%9D.), FSM)，通过管理状态转换和决策过程来设计和实现游戏对象行为。有限状态机包括三个组成部分：

* 状态（states）：代表游戏中的实体或对象可能处于的具体条件或情况，如“空闲”，“巡逻”，“追逐”，“攻击”等
* 转换（transitions）: 定义导致从一个状态到另一个状态的条件或事件，如从“巡逻”到“追逐” 的转换可能是敌人NPC在一定范围内检测到玩家时发生
* 动作 (actions)：动作与状态相关联，决定AI在每个状态下应执行是么操作。如在“巡逻”状态下，AI会按照定义的路径或航点移动

视频讲解了如何加入逻辑来实现FSM并在游戏循环周期中更新。如对路径规划算法感兴趣，可以参考 ：[A\*算法](https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2)，[Dijkstra 算法](https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/)，[贪心算法](https://ai-master.gitbooks.io/heuristic-search/content/what-is-greedy-best-first-search.html)等

课程作业:

1. 根据教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】完成一部分FSM的编写

课程时间：2023年09月08日

课程内容:

* 课上Bradon 完成了第一人称射击教程【[3](https://youtu.be/LugpgsMdLWw?si=UBq2F2eVVo4EMo4n)】的内容

课程反馈:

上节课Bradon观看了教程【[3](https://youtu.be/LugpgsMdLWw?si=UBq2F2eVVo4EMo4n)】有关玩家生命值与伤害效果的内容，这节课Bradon完成了了本地复现。下节课将学习有关游戏人工智能 – 巡逻（Game AI - Patrolling）的教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】，指的是编辑游戏中非玩家角色（NPCs）或实体移动的行为。NPCs会沿着预定义的路径或路线在游戏世界内移动，种行为通常用于模拟游戏环境中需要覆盖特定区域或区域的守卫、怪物或其他实体的移动模式。关键组成部分有：

* 寻路 (pathfinding)：通常使用A\*或Dijkstra等寻路算法构建路径以使NPC穿越游戏世界中的特定路标或区域
* 路标 (waypoints)：这些是游戏世界中的预定义位置，AI会在巡逻时按照它们的路径前进。路标通过连接在一起以创建路径
* 行为逻辑 (behavior logic)：AI到达路标时，需要逻辑来确定要执行的操作。它可能会短暂停顿、掉头或执行其他预定义的行为

课程作业:

1. 观看教程【[4](https://youtu.be/Mp8yFUKxldk?si=JWol19nEPGYyAO5l)】

课程时间：2023年09月02日

课程内容:

* 课上Bradon 完成了第一人称射击【[2](https://youtu.be/gPPGnpV1Y1c?si=KAyy1TIynxrpvmJZ)】及【[2.5](https://youtu.be/_UIiwzfZoZA?si=XSar5bDetaQBWZ6t)】教程的复现并观看了教程【[3](https://youtu.be/LugpgsMdLWw?si=UBq2F2eVVo4EMo4n)】的内容

课程反馈:

这节课前一部分Bradon完成了上节课及作业剩余的部分——复现了教程【2】和【2.5】中有关射线投影及交互的内容。课程的后半部分Bradon观看了教程【3】有关玩家生命值与伤害效果的内容，包含以下要点

* 玩家生命值脚本 (player health script)：根据交互事件（是否被击中）更新玩家的生命值
* 添加叠加效果 (add in an overlay effect)：添加功能以支持在主视角页面叠加显示效果
* 淡化图像 (fade the image)：支持叠加显示效果的淡化
* 伤害叠加效果 (damage overlay)：结合前面编写好的功能设置玩家被击中后的伤害效果

课程作业:

1. 复现教程【[3](https://youtu.be/LugpgsMdLWw?si=UBq2F2eVVo4EMo4n)】的内容；下节课学习教程【4】有关游戏人工智能 – 巡逻 (Game AI - Patrolling) 的内容

课程时间：2023年08月18日

课程内容:

* 课上Bradon 完成了第一人称射击 (FPS) 前两期教程的学习，及移动方式和射线投影交互

课程反馈:

以下为这两部分内容的要点

* 移动方式 (movement)
  + 玩家设置 (player setup), 导入输入系统 (import input system), 移动和跳跃动作 (movement & jump actions), 输入管理器 (input manager), 移动 (movement), 输入调试器 (input debugger), 添加重力属性 (adding gravity), 跳跃 (jumping), 玩家视角动作 (player look actions)
* 射线投影交互 (raycast interactions)
  + 射线投影 (raycast), 射线投影函数 (raycast function), 层级遮罩 (layer mask), 覆盖交互功能 (override the interact function), 使用文本在屏幕上显示消息 (display message on the screen using text), 创建准星 (create a crosshair), 创建动画 (create the animation)

课程作业:

1. 观看视频 [2.5] 更深度了解交互事件的设置，下节课学习视频 [3] 设置玩家的生命值和伤害效果

课程时间：2023年08月12日

课程内容:

* 这节课前一部分Bradon 完成了3D太空射击游戏，剩余的时间开始学习开发第一人称射击游戏

课程反馈:

课上Bradon完成了3D太空射击游戏 (3D Space Shooter) 最后的优化。由于Bradon对射击类游戏感兴趣，我们选择进一步学习第一人称射击 (FPS: First- Person Shooter) 游戏的开发。教程 ([链接](https://youtu.be/rJqP5EesxLk))共包含7个视频，分别介绍以下内容：

[1] FPS 的移动方式 (movement)

[2] FPS的射线投影 (raycast interactions)

[2.5] 交互事件 (interaction events)

[3] 玩家的生命值和伤害效果 (player health & damage effects)

[4] NPCs 的 巡逻行为 (game AI - patrolling)

[5] NPCs 的战斗和攻击行为 (game AI – combat/attack)

[6] NPCs 的搜索状态 (search state)

课程作业:

1. 完成第一个视频的学习和开发

课程时间：2023年08月05日

课程内容:

* 这节课继续根据教程（[链接](https://www.youtube.com/watch?v=VW3PkEF1Fzk)）上节课的3D太空射击游戏项目。

课程反馈:

出于对该项目的兴趣，这节课Bradon继续开发太空射击游戏。课上已完成了如下部分：

1. 导入必要的Unity asset 和 packages
2. 创建主场景 (main scene) 并添加太空空间 (space skybox)
3. (选做) 安装Odin 验证器 (inspector) 用于在Unity编辑器中调整变量和属性
4. 添加玩家飞船

待完成的部分包括：

* 给飞船添加飞行属性
* 调整背景相机 (background camera) 的旋转 (rotation) 以匹配主相机 (main camera)
* 添加星际尘埃 (space dust) 的颗粒效应 (particle effects)
* 设置飞船移动输入控制 (input controls)
* 将光标限制在游戏窗口 (game window) 并隐藏它

课程作业:

1. 完成一部分待完成的开发，下节课继续完成剩余的部分

课程时间：2023年07月28日

课程内容:

* 这节课开始做新的3D游戏 – Space Shooter (太空射击)。课上简单介绍了该游戏涉及的内容，之后与Bradon一起观看了视频教程（[链接](https://www.youtube.com/watch?v=VW3PkEF1Fzk)）并着手实现项目。

课程反馈:

Bradon提到暑校的游戏项目开发使他感受到UI设计是当前的难点。因此这节课我们选择做一款注重游戏画面设计的游戏，3D 太空射击。该游戏包含以下的内容：

1. 设置项目 (set up the project): 导入必要的Unity asset 和 packages, 包括2D或3D模型，结构 (textures) 和音频文件
2. 创建玩家角色 (create the player character): 实现玩家移动控制（例如键盘，鼠标或游戏手柄），以及添加射击机制并设置玩家的武器系统
3. 设计敌人AI (design enemy AI): 创建敌人角色和行为，实现敌人的移动模式和射击模式，让敌人在特定间隔或位置出现
4. 碰撞检测和伤害系统 (collision detection and damage system): 实现碰撞检测，以检测玩家的子弹是否击中敌人，反之亦然。创建伤害系统，以管理玩家和敌人的生命值和伤害
5. 其他可以添加的功能如关卡设计，用户界面，和音效等

课程作业:

1. 根据教程（[链接](https://www.youtube.com/watch?v=VW3PkEF1Fzk)），完成该游戏一部分的开发，下节课可以继续完成后面的部分

课程时间：2023年06月16日

课程内容:

* 讲解简易版3D Endless Runner (神庙逃亡) 游戏的要点
* 课上Bradon观看了视频教程（[链接](https://www.youtube.com/watch?v=UKpTUkD2z8Q)）并开始该游戏项目

课程反馈:

Bradon完成了上节课的作业（俄罗斯方块游戏）。这节课开始做3D游戏的开发 – 神庙逃亡，这是一款开发比较简单而又流行的3D游戏（[游戏介绍](https://zh.wikipedia.org/wiki/%E7%A5%9E%E5%BA%99%E9%80%83%E4%BA%A1)）。首先我讲解了下开发这款游戏的要点：

* 无尽关卡生成 (infinite level generation)：创建一个能够生成无尽且动态变化的游戏关卡的系统。这涉及在玩家游戏过程中随机生成障碍物，平台和道具。关卡生成应该提供流程而挑战性的游戏体验
* 障碍物和道具逻辑（obstacle and power-up logic）: 定义障碍物和道具的行为。障碍物应该以适当的距离和高度放置，道具应该给玩家提供暂时的优势
* 计分和进程（scoring and progression）：设计一个计分系统，用于追踪玩家在游戏中的表现和进程。根据行进的距离，躲避的障碍物和收集的道具来增加分数（注：该[视频教程](https://www.youtube.com/watch?v=UKpTUkD2z8Q)不包含此功能，可参考其他教程来添加）
* 游戏结束和重新开始（game over and restart）： 实现逻辑来处理游戏结束时玩家与障碍物碰撞或掉落平台的情况。显示最终得分，并提供选项让玩家重新开始游戏或返回主菜单
* 音频和视觉效果（audio and visual effects）：通过合适的音效和背景音乐增强游戏体验。添加视觉效果，如粒子系统，屏幕抖动和相机效果等，使游戏更具吸引力

课程作业:

1. 根据教程（[链接](https://www.youtube.com/watch?v=UKpTUkD2z8Q)），完成简易版神庙逃亡的开发

课程时间：2023年06月02日

课程内容:

* 讲解了Tetris (俄罗斯方块) 游戏的要点
* 课上Bradon根据教程（[链接](https://noobtuts.com/unity/2d-tetris-game)）完成了大部分的开发

课程反馈:

Bradon完成了上节课的作业（扫雷游戏）。这节课开始做俄罗斯方块（一款经典的图块匹配游戏- [游戏介绍](https://zh.wikipedia.org/wiki/%E4%BF%84%E7%BE%85%E6%96%AF%E6%96%B9%E5%A1%8A)）的游戏开发。首先我讲解了下开发这款游戏的要点：

* 四连方块 (tetrominoes)：由四个方格 (block)组成的多连方块 (group)。四连方块共有七种，分别为I, O, T, S, Z, J 和 L
* 游戏区间 (playfield)：游戏区间是由网格 (grid) 的数据结构 (data structure) 构成的 （二维的数组，也叫作矩阵），每个数组元素记录该空间网格是否被填充的状态。这一部分开发涉及到编写若干辅助函数：
  + roundVec2: 将四连方块的移动向量四舍五入为整数
  + insideBorder: 判断方格是否在游戏区间内。若判断游戏指令会使四连方块超出游戏边界，则指令不会被执行
  + deleteFullRows:
    - isRowFull: 判断游戏区间内各行并判断改行是否被方格填充满
    - deleteRow: 该行的方格游戏对象删除 (Destroy)，对应网格状态由填充转变为空 (null)
    - decreaseRow & decreaseRowsAbove: 将被删行之上的各行依次向下移动一行，以此来模拟组块被消后下落的行为
* 编写脚本定义组块的运动逻辑：
  + 判断组块移动的目标网格是否有效：此处会用到在游戏区间部分编写的辅助函数
  + 移动 (move) 和下落 (fall)： 移动包括左，右，和旋转（每次顺时针旋转90度）

课程作业:

1. 根据教程（[链接](https://noobtuts.com/unity/2d-tetris-game)），完成剩余的俄罗斯方块开发
2. 之前我们做了若干2D的小游戏项目，下节课可以开始接触更复杂的3D游戏开发，若有时间可以课前预览下内容（神庙逃亡 – [链接](https://assetstore.unity.com/packages/templates/tutorials/endless-runner-sample-game-87901)）

课程时间：2023年05月26日

课程内容:

* 讲解了Minesweeper (扫雷) 游戏的要点
* 课上Bradon根据新的视频教程优化了吃豆人游戏

课程反馈:

在完成上节课的作业（剩余的吃豆人游戏开发）过程中， Bradon发现原教程有的地方不能复现出预期的效果，但他在Youtube上找到了能解决这一问题的更详细的视频教程。本节课的游戏编程部分Bradon根据新的教程继续完善了吃豆人游戏的开发。

本节课游戏讲解的部分我介绍了一个新的游戏—— 扫雷（一款单机解谜类电脑游戏 – [游戏介绍](https://zh.wikipedia.org/zh-hans/%E8%B8%A9%E5%9C%B0%E9%9B%B7)），作为课后作业。这款游戏的开发用到了之前小游戏项目所学的知识，开发过程包含以下几个模块：

* 设置界面 (Set up the game grid)：二维的网格化界面，每个网格由一个游戏对象来存储该单元的状态，如是否含雷或是否被点开
* 埋雷 (Place mines)：在网格界面上随机放置雷点，可以通过编写一个脚本随机生成坐标集并标记为含雷区
* 计算周围的雷区数量 (Calculate numbers)：对于网格界面上的每个非雷单元，计算周围8个网格的含雷数量， 并将该信息存储在响应的网格中
* 用户输入 (User input)：编译用户与游戏交互的接口，比如鼠标点击单元格来揭示该单元是否含雷
* 揭示单元格 (Reveal cells)：当玩家鼠标点击某单元格，若该单元格含雷，游戏结束，否则递归式揭示周围的单元格直至单元格与雷区相邻
* 游戏结束的条件 (Game over conditions)：玩家每步操作后查看是否满足游戏结束的条件。如果玩家点到雷区，结束游戏并揭示界面内所有未显示的雷区。若玩家成功揭示所有的非雷区，则玩家获胜
* 游戏循环 (Game loop)：设置一个游戏循环从而能反复地接受玩家的输入指令并相应地更新游戏状态。这一循环会持续到游戏直到输或赢

课程作业:

1. 根据教程（[链接](https://noobtuts.com/unity/2d-minesweeper-game)），完成扫雷游戏的开发
2. 根据Bradon的提议，下节课开始编写游戏俄罗斯方块，若有时间可以课前预览下内容（简易版 – [教程](https://noobtuts.com/unity/2d-tetris-game)，进阶版 – [教程](https://www.youtube.com/watch?v=ODLzYI4d-J8)）

课程时间：2023年05月19日

课程内容:

* 讲解Pac-Man (吃豆人) 游戏的要点
* 课上Bradon完成了该游戏一部分的内容

课程反馈:

Bradon完成了上节课的作业（剩余的乒乓游戏开发）。这节课开始做一个更复杂的游戏—— 吃豆人（一款街机游戏 – [游戏介绍](https://zh.wikipedia.org/zh-hans/%E5%90%83%E8%B1%86%E4%BA%BA)）。这个游戏的开发用到了之前小游戏项目所学的知识，开发过程会拆解为几个任务模块并最终整合起来。这款游戏的新知识如下：

* 状态机 (state machine)：状态机是游戏开发中常用的一项技术，用来定义和模拟游戏实体或系统的不同状态及行为。状态机由三个组分构成
  + 状态 (states)：游戏中实体或系统的状态。比如 “空闲”, “走”, “跳”, “攻击” 等
  + 过渡 (transitions)：条件或事件导致状态的转换。可以是玩家输入，基于时间的事件，碰撞检测等
  + 行为和逻辑 (actions and logic)：每个状态下包含的行为和逻辑。比如更新动画效果，处理输入，与其他游戏对象交互等
* 游戏人工智能 (game AI)：游戏AI是游戏开发中定义智能行为及决策过程的重要技术，涉及到编写算法和系统来模拟类人的智能或增加游戏的挑战性。常见的游戏AI应用场景如：非玩家角色的行为 (NPC behavior)，路径规划 (pathfinding) ，决策 (decision making)，学习和适应 (learning and adaptation)，程序化内容生成 (procedural content generation)

开发过程中需要注意的地方：

* 需要分别设置迷宫 (maze) 中各墙壁的碰撞器 (collider)
* 将吃豆人和鬼魂的单张动画 (animation) 分割成单独的状态 (sprite slicing)
* 用状态机 (state machine) 编写吃豆人 (Pac-Man) 的动画效果（上，下，左，右）
* 通过路径点 (waypoints) 来定义鬼魂 (Ghosts) 的AI运动路径
* 用之前学到的碰撞检测来处理吃豆人吃豆及鬼魂吃吃豆人

课堂上Bradon根据教程（[链接](https://noobtuts.com/unity/2d-pacman-game)）复现了该游戏一部分的内容

课程作业:

1. 根据教程（[链接](https://noobtuts.com/unity/2d-pacman-game)），完成吃豆人吃豆的内容，下节课编写鬼魂的运动

课程时间：2023年05月13日

课程内容:

* 讲解Pong (乒乓球) 游戏的要点
* 课上Bradon完成了游戏编写的大部分内容

课程反馈:

Bradon完成了上节课剩余的贪吃蛇游戏开发，这节课做了个新的游戏 – 乒乓球（一个益智类游戏 - [游戏介绍](https://zh.wikipedia.org/wiki/%E4%B9%93)）。课程开始我先介绍了乒乓游戏的要点：

* 球拍运动 (paddle movement)：两侧的球拍只能纵向移动，左右两侧分别由不同的按键控制上下
* 球体运动 (ball movement)：球体在设定的墙壁内运动，在碰撞球拍和墙壁后会反弹（遵循物理定律，涉及到贪吃蛇中设计的碰撞检测），需要编辑运动的速度和方向并更新其在屏幕中的位置

以及开发过程中需要注意的地方：

* 两侧的球拍要定义不同的轴（axis）来分别设置键盘上的不同按键来控制移动
* 球体的刚体 (Rigidbody 2D) 组分要将重力尺度 (Gravity Scale) 设置为0，否则球体的运动会受到重力的影响

课程的后半部分Bradon根据教程（[链接](https://noobtuts.com/unity/2d-pong-game)）复现了该游戏大部分的内容，我解答他所遇到问题。

课程作业:

1. 根据教程（[链接](https://noobtuts.com/unity/2d-pong-game)），完成乒乓游戏剩余的内容：
   1. 着重理解球碰撞球拍的三种模式以及它们是如何编译的
2. 如有余力可以预览下节课要做的吃豆子游戏 （[链接](https://noobtuts.com/unity/2d-pacman-game)）

课程时间：2023年05月05日

课程内容:

* 讲解贪吃蛇游戏项目中的要点
* 课上Bradon 独立完成了 Snake Game (贪吃蛇) 一半的内容

课程反馈:

Bradon完成了上节课布置的井字棋游戏作业，于是这节课开始做一个新的游戏 - 贪吃蛇（一个益智类游戏 - [游戏规则](https://baike.baidu.com/item/%E8%B4%AA%E5%90%83%E8%9B%87/9510203#:~:text=%E8%B4%AA%E5%90%83%E8%9B%87%E6%B8%B8%E6%88%8F%E7%8E%A9%E6%B3%95&text=%E7%94%A8%E6%B8%B8%E6%88%8F%E6%8A%8A%E5%AD%90%E4%B8%8A%E4%B8%8B,%E7%BB%A7%E7%BB%AD%E7%8E%A9%E4%B8%8B%E4%B8%80%E5%85%B3%E3%80%82)）。由于Bradon自学能力较强，课程的前半段主要由他自主学习教程（[链接](https://noobtuts.com/unity/2d-snake-game)）并在本机同步复现，我负责解答这一过程中碰到的问题。课程后半段我大体介绍了下贪吃蛇游戏考察的要点：

* 游戏循环 (game loop): 用于随机生成屏幕中的食物
* 碰撞检测 (collision detection): 判断蛇知否与边框碰撞以及是否与食物接触
* 用户输入(user input): 需要读取玩家的输入指令来控制蛇的移动

以及开发过程中需要注意的地方：

* 虽然墙壁和食物都有 Box Collider 2D 模块，两者的区别在于是否选择了 Is Trigger – 墙壁选择了因而有碰撞属性，食物没有选择所以可以接触但不碰撞

课程作业:

1. 根据教程（[链接](https://noobtuts.com/unity/2d-snake-game)），完成贪吃蛇游戏后半部分内容：
   1. 编辑贪吃蛇对象， 涉及蛇头和蛇尾元素不同的处理方法
   2. 编辑蛇吃食物的运动逻辑

课程时间：2023年04月29日

课程内容:

* 介绍了游戏项目所需要的工具和软件
* 演示了一遍如何在Unity Editor上实现TicTacToe (井字棋) 的游戏开发

课程反馈:

这节课向Bradon具体讲解了井字棋（一个益智类游戏 - [游戏规则](https://zh.wikipedia.org/zh-hans/%E4%BA%95%E5%AD%97%E6%A3%8B)）在Unity平台上的开发流程。首先介绍了下开发游戏项目所需的工具:

* Unity Hub: 用于安装Unity软件及管理游戏项目的平台 [[链接](https://store.unity.com/download?ref=personal)]
* Unity Editor: 用于开发游戏的集成开发平台 [[链接](https://unity.com/releases/editor/archive)]
* Visual Studio Code: [[链接](https://code.visualstudio.com/)]

按照教程（[链接](https://www.red-gate.com/simple-talk/development/dotnet-development/using-unity-ui-and-c-to-create-a-tic-tac-toe-game/)）, 从头到尾演示了一遍如何完成井字棋的游戏开发，包括如下几部分:

* 介绍了Unity Editor 界面下各窗口的功能
* 在游戏界面中添加不同种类的物体 (panel, button, text, GameObject) 并更改其属性
* 编辑C#脚本，定义物体可实现的功能
  + 如何导入包，设置变量，编写功能函数
* 设置界面下的物体与编辑脚本间的映射关系

课程作业:

1. 安装软件：
   1. 安装Unity Hub（[链接](https://store.unity.com/download?ref=personal)）及Unity Editor（[链接](https://unity.com/releases/editor/archive)）
   2. 安装代码编辑器Visual Studio Code （[链接](https://code.visualstudio.com/)），用于编辑游戏中的C#脚本
2. 井字棋游戏开发：
   1. 根据课上所演示的步骤以及教程（[链接](https://www.red-gate.com/simple-talk/development/dotnet-development/using-unity-ui-and-c-to-create-a-tic-tac-toe-game/)），在本机的Unity平台上实现井字棋项目（如果感觉任务量大，可分两次作业完成）

课程时间：2023年04月21日

课程内容:

* 了解Bradon的编程基础以及想做的项目
* 探讨了下大体的游戏项目规划

课程反馈:

了解到Bradon对编程比较感兴趣，之前学过C++和Python。Bradon喜欢游戏尤其是策略类的，并想做一些和游戏编程相关的项目。基于他此前没有相关的经验，在课上我们一起探讨了大体的计划安排，[项目规划](https://docs.google.com/document/d/1zVtWqDz4yxomFM0v1EehclM0uCZNa8JgKAnpYbQHj4E/edit?usp=sharing)分为两部分。

1. 第一部分旨在掌握游戏设计的基础，选择了Unity Learn上的辅导课程 Junior Programmer（[链接](https://learn.unity.com/pathway/junior-programmer)）。这门课主要介绍Unity 游戏设计平台的基本功能和知识，对后面在此平台上做游戏项目十分必要。
2. 第二部分是游戏项目部分，选择了一些入门的项目 – 井字棋（Tic Tac Toe, [链接](https://www.red-gate.com/simple-talk/development/dotnet-development/using-unity-ui-and-c-to-create-a-tic-tac-toe-game/)），乒乓球（Pong, [链接](https://github.com/zigurous/unity-pong-tutorial)），神庙逃亡 (Endless Run type, [链接](https://assetstore.unity.com/packages/templates/tutorials/endless-runner-sample-game-87901))和贪吃蛇（Snake Game, [链接](https://noobtuts.com/unity/2d-snake-game)）。Bradon可以挑几个感兴趣的项目做，它们会帮助更深入地了解游戏设计中的角色移动，游戏机制，碰撞探测，及程序化生成

课程作业:

1. 准备：
   1. 游戏项目会在Unity Engine上完成（C#语言）， 需要下载并安装Unity Hub （[链接](https://store.unity.com/download?ref=personal)）
2. 游戏基础：
   1. 开始学习辅导课程 Junior Programmer（[链接](https://learn.unity.com/pathway/junior-programmer)），与做游戏项目同步进行
3. 游戏项目
   1. 根据提供的链接了解游戏项目的大体内容，挑选出感兴趣的，下节课将制定具体的项目实施方案