

TermProject 第二阶段报告

姓名: 邵钰乾 学号: 211240036

May 16, 2022

1 完成进度

本项目第二阶段的必做部分在 5 月 3 日全部完成, 详细说明如下:

- **素材收集:** 本项目中全部贴图都是我亲手画出来的, 使用的网站是[这个](#)。贴图的基本尺寸是 64 * 64, 有一些大了。
- **开始菜单:** 本游戏中, 开始菜单有“教程”、“PVP 模式”、“PVE 模式”和“退出游戏”。“教程”使用了一个单独的游戏场景, 摆放了软墙和游戏说明, 玩家可以在这个场景下操作并熟悉控制方式, 同时了解道具功能。“退出游戏”就是退出游戏。
- **地图:** 目前的地图是品类齐全的, 并且具备 20 * 15 的标准尺寸。
- **玩家:** 玩家会根据朝向改变形象, 且有基于坐标位置切换的逐帧动画 (这使得移动速度快时动画切换频率高, 反之亦然)。
- **机器人:** 目前, 机器人会尝试向玩家移动, 通过计算炸弹位置和爆炸范围躲避炸弹, 并以一定时间间隔释放炸弹。目前智能不高。同样也配有动画。
- **炸弹:** 炸弹在爆炸时会产生冲击波, 冲击波根据炸弹等级向四周延伸。在每一格冲击波产生后, 会在该格进行伤害判定。同一个炸弹不同距离的冲击波并不是同时完成伤害判定, 而是向四周延伸。
- **道具**
 - “增加速度”: 将玩家的速度大幅提升 (机器人吃掉该道具会获得同样效果);
 - “增加炸弹威力”: 炸弹的爆炸范围由 1 增加至 5 (机器人吃掉该道具会获得同样效果);
 - “增加炸弹个数”: 玩家允许在场景中国放下的炸弹总数由 1 增加至 8 (由于机器人以固定时间间隔投放炸弹, 该道具对机器人暂时没有效果);
 - “移动炸弹”: 玩家与某炸弹位于同一格时, 可以使用方向键将炸弹踢到下一堵墙之前 (该道具对机器人暂时没有效果)。
- **buff 限制规则:** 玩家和机器人都不可以无限制地叠加效果。在同时拥有的效果大于等于 3 (多次获得同一个效果以多次计算) 时, 新获得的效果会洗掉最先获得的旧效果。
- **其他显示**

玩家、机器人实时得分和获得效果情况统一在状态栏显示。此外, 在状态栏上还有玩家、机器人的名字和血条; 暂停按钮已经实装。

2 过程记录

在这里主要介绍遇到的问题以及解决方案:

1. 来自过去的遗产：Game Manager

在游戏中有一些不可避免的集中管理问题：玩家类需要获取场景信息做出判断，炸弹类和冲击波类需要访问玩家和机器人类，同时也需要访问场景进行全局修改……诸如此类。这时候可以使用一个在每个场景中唯一存在的 GameManager（或者 GameController）来处理这类需要访问场景信息来进行处理的问题。实际上这并不是我自己新想出来的，而是早期进行基于 GameMaker 和 unity 进行游戏编程时学习的通用做法。

我的具体实现中，在 GameManager 类中封装了以整型数组形式存储的实时地图和炸弹信息，以及访问地图中所有炸弹、墙、玩家、机器人的接口。同时，在地图中以确定坐标摆放炸弹、冲击波、道具；展示游戏胜利、结束、暂停的 UI；管理背景音乐这些全局操作也实际上由 GameManager 类完成。

实际上，我尽量尝试着将地图生成时所有操作搬入 GameManager 类的 onAttach() 函数中，使 MainWindow.cpp 更加简洁，这样易于管理。但是由于一开始在 MainWindow.cpp 中完成的大量工作不适合迁移，所以暂时搁置了。

2. 基于四点判断的碰撞检测

我规避了 ItemAt 和 collider 的做法，自行使用了一个基于四个检测点的碰撞检测方法。

以游戏角色的判定点（在这一点判定该游戏角色“在哪一格”），分别向上、向下、向左、向右画线，这四条射线与碰撞箱的交点作为判断点。例如游戏角色要向右移动，则考察上、下两个端点，若上、下两个端点所在的格右方有墙，则限制游戏角色目前可以向右行进的成都。以此类推，上下移动是判断左右两端点——就像猫的胡子一样。碰撞箱的尺寸比一格小很多，避免严丝合缝的检测对玩家操作精确度的过高要求。

3. 事件序列遗留引发的场景多次加载问题

在本阶段中遇到了一个大 bug——在通过按钮事件重载场景时，有一定概率场景被加载多次，导致卡顿等错误。这是因为 Qt 中，鼠标事件和 onUpdate 的调用并无明显关联。在一次点击下，按钮的点击事件可能被调用多次，使得多个“重载”事件进入 Qt 的事件队列。当第一个“重载”事件被执行后，旧场景中 item 被清除（这是基于我实现的将场景中物体全部 detach 的函数），新场景被加载；然而这时候还有“重载”事件遗留在事件队列里，造成多次重载。最后的解决方式添加一个“使能信号”，在按钮被按下一次后，它将“丧失活性”，点击它将不再具有任何作用。

感谢李晗学长帮我 de 出了这个 bug!!

4. 机器人

这次的机器人采用了贪心随机游走，定时释放炸弹的算法。在“普通”状态下，机器人会尽量朝靠近离它最近的玩家的方向移动（当某玩家死亡时，机器人转为靠近唯一存活玩家），若无法靠近则随机游走；机器人不会走“有危险（即在某炸弹爆炸范围内）”的格子。

在“紧急”状态（机器人坐在的格子是“危险”的，可能被炸到）下，机器人放弃接近玩家，以逃离危险区域为最高优先级。逃跑路线是随机的。当机器人发现已经到达安全区域，它会等待稍长于炸弹引线的一段时间。这是为了防止机器人在炸弹爆炸时再次走进爆炸范围（前文已经提到，冲击波有一定延时和顺序。冲击波造成伤害时，炸弹可能已经消失，机器人无法判定）。

最后呈现出的效果还不错。

3 实验心得

做完了就好

4 致谢

首先要感谢李晗学长的框架！太好用了！同时也要感谢朱宇博学长的项目文档。其次要感谢张明轩同学和龚至诚同学，我们进行了很多有益的交流。最后仍要感谢杨博一、吕佳迅、吴沂桐、刘翊等同学对我在游戏开发方面一直以来的支持。