



一、项目分工

学号	名字	角色	班级	职责	贡献
16340284	张丰露	组长	上午班	负责部分逻辑实现，界面美化，部分游戏 UI 素材，写实验报告	25%
16340274	杨元昊	组员	下午班	负责部分逻辑实现，游戏平衡性测试，UI 与人物交互，写实验报告	25%
15355119	姚雪辉	组员	上午班	负责部分逻辑实现，物理引擎实现，部分游戏素材，技能动画	25%
16340316	朱逸渠	组员	上午班	负责部分逻辑实现，地图实现，部分游戏素材，人物移动动画	25%

二、开发环境

Windows10

Visual studio2017

Cocos2d-x 3.17

Python 2.x

三、项目阐述

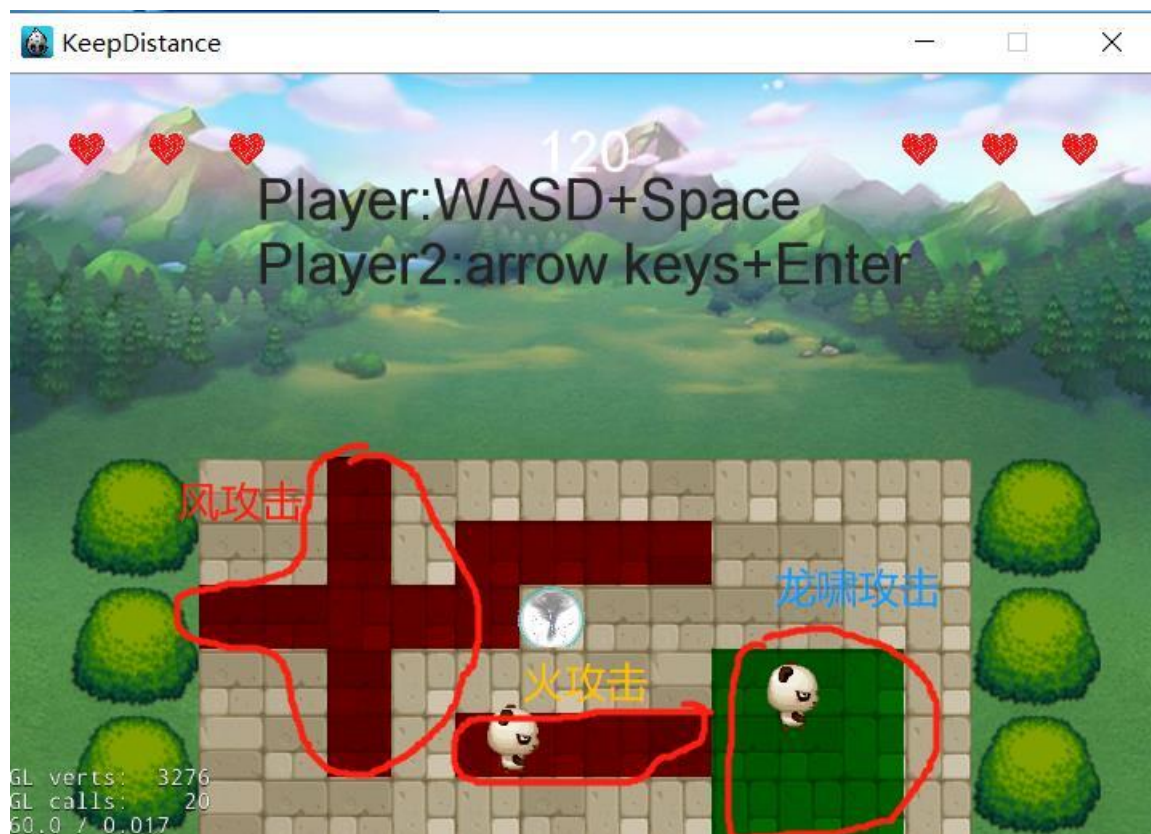
游戏名称： KeepDistance

游戏简介： 两个玩家通过释放技能占领方格并改变颜色，在自己的领域内，移动速度加成，在敌方领域内，移动速度减少。玩家所在的区域被敌方玩家占领的一瞬间视为被攻击，玩家被击退一格并扣除血量。

游戏玩法： 玩家一 WASD 控制上下左右的移动，SPACE 释放技能，玩家二四个方向键控制上下左右的移动，ENTER 释放技能，每隔十秒，地图中间会出现技能道具，通过接触技能道具，改变技能。释放技能的时候会占领方格，敌方玩家可以通过释放技能重新占领已经被占领的方格。



游戏攻略：



首先不同的技能所占领的方格形状是不一样的，也就是说攻击距离以及方向是不一样的。

风攻击，十字形的攻击。

优点：四个正方向的中距离攻击。

缺点：无法打出对角线的攻击。

火攻击，直线形的攻击。

优点：一个正方形的远距离攻击。

缺点：只能攻击一个方向。

龙啸攻击，正方形的攻击。

优点：八个方向的近距离攻击。

缺点：攻击距离短。



高玩建议：

第一：由于在自己的领域内有速度加成，也就是有更大的机动性和灵活性，所以要多占领自己的领域。即使是游戏一开始，就要开始占领区域，新手一般会直接攻击敌对玩家，但是由于没有占领领地，机动性较差，所以不太容易获胜。

第二：移动的时候不能释放技能，释放技能的时候不能移动，所以要在移动和释放技能做取舍，一般被攻击的最好的时机，就是当你释放技能后的零点几秒。

游戏亮点：

第一：贴切的背景音乐以及音效。游戏开始界面采用舒缓奇幻风格的背景音乐，引人入胜。游戏开始则采用轻松欢快的泡泡堂背景音乐，俏皮可爱。人物移动的音效，非常 Q，配合萌萌的人物，恰到好处。人物死亡配合超级玛丽的音效，让人瞬间带入到童年时代。

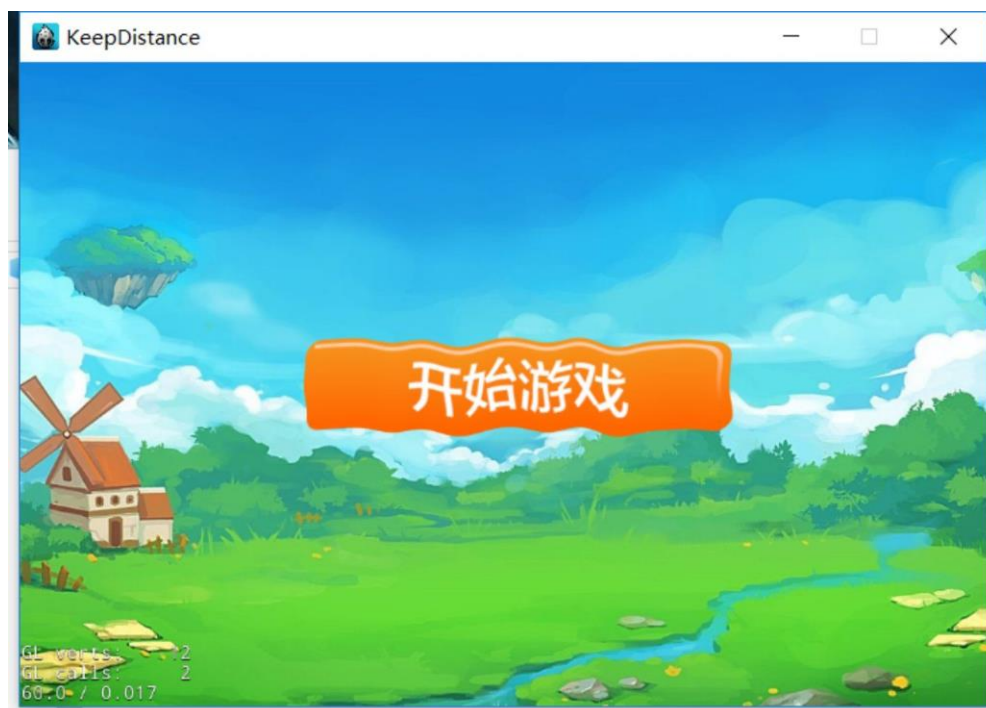
第二：炫酷的游戏画面。无论是风攻击，火攻击还是龙啸攻击，技能效果十分酷炫，让人顿时身处电闪雷鸣，刀山火海，虎啸龙吟之中，音效的配合，让人身临其境。

第三：玩家博弈的哲学。整个游戏的区域仅仅巴掌大的地方，却包含着攻守的智慧。玩家需要步步为营，在缭乱的走位中，迷惑敌方，颇有乱花渐欲迷人眼之意。又需要根据不同的技能采取不同的战斗策略，万变不离其宗，以不变应万变。还需要时刻绷紧神经，找出攻击的时机，胜负只在千钧一发之际。小小地图，万千世界。

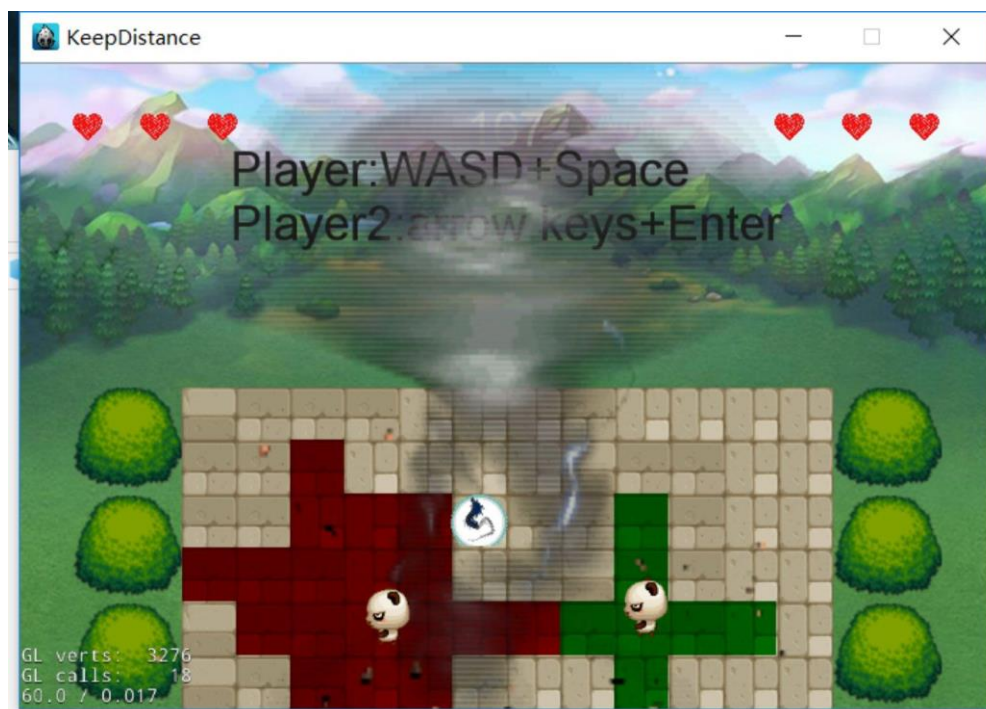


四、项目展示

游戏进入界面

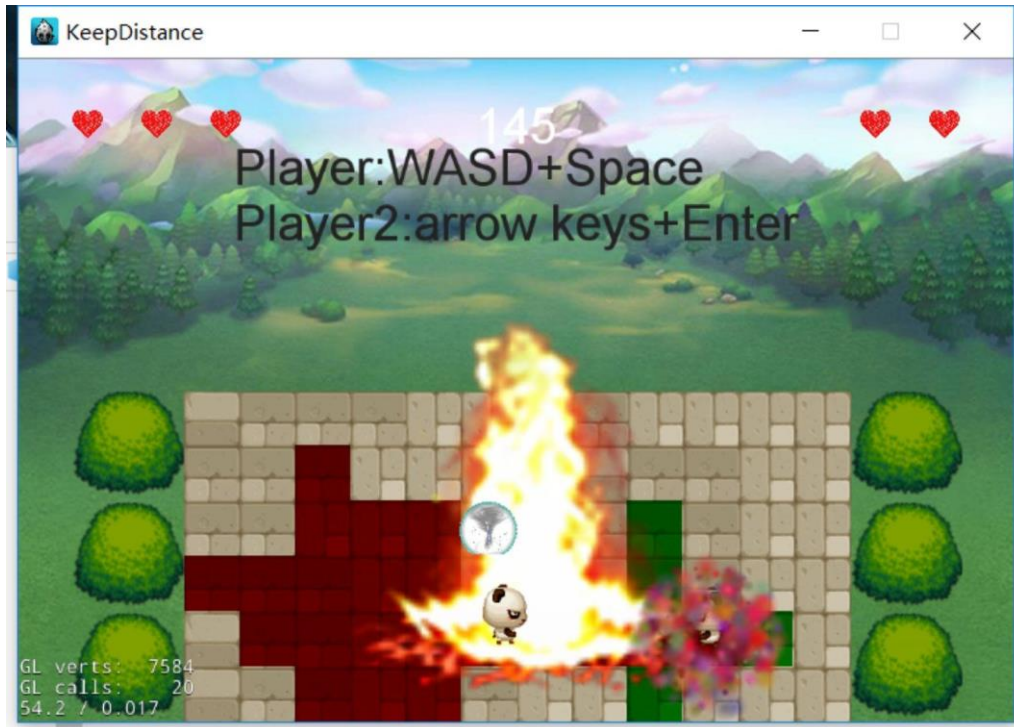


释放风技能

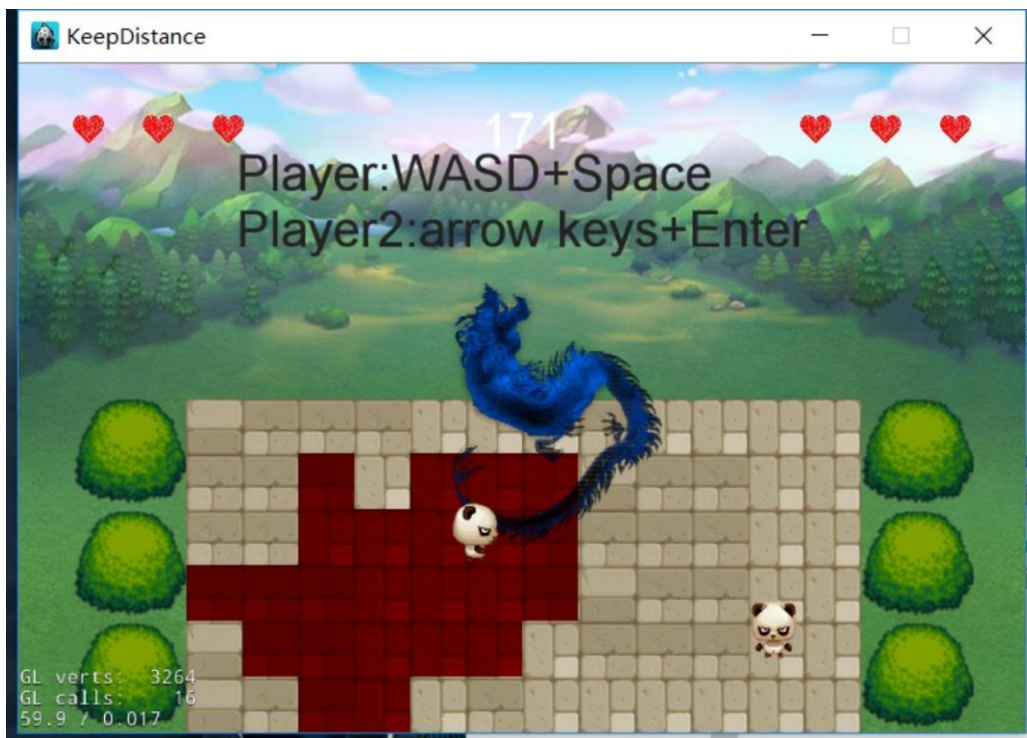




释放火技能，以及人物被攻击到的粒子效果

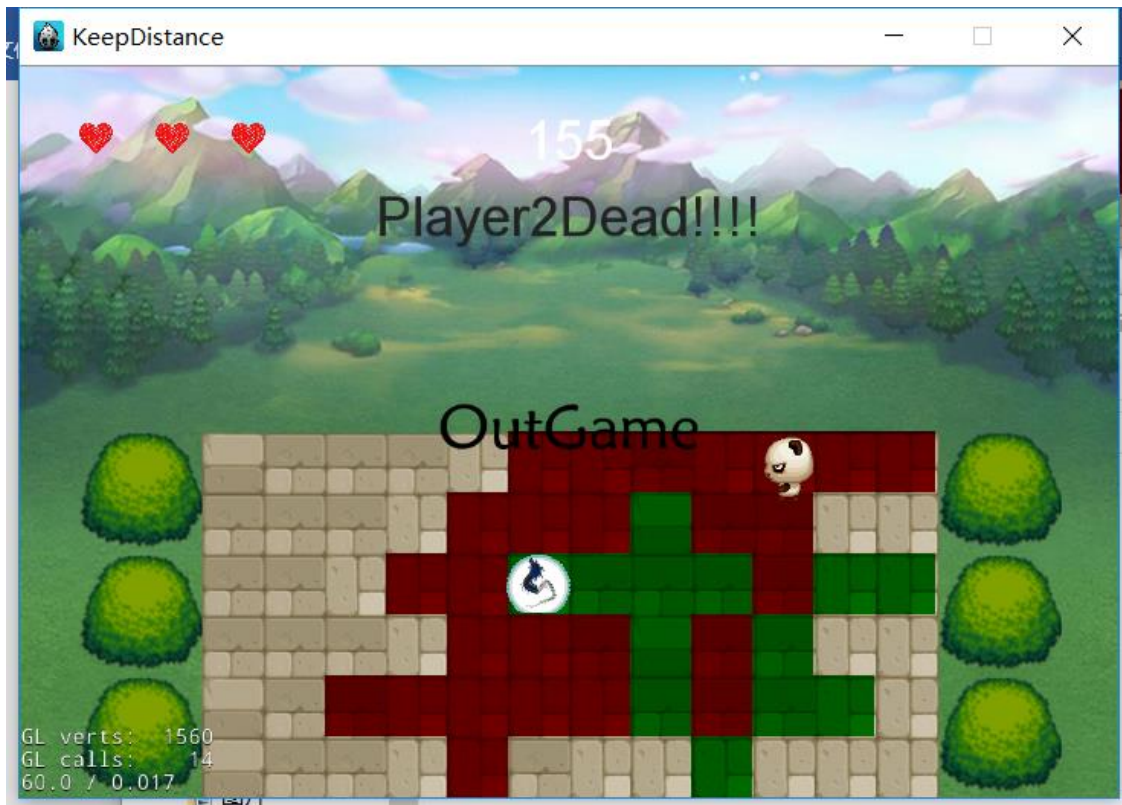


释放龙技能





游戏结束：



五、项目难点及解决方案

问题：按键体验不流畅，手指没离开键人物就停下了。

原因：方向运动的调控由 `OnKeyPressed` 和 `OnKeyReleased` 确定，`OnKeyPressed` 时设置运动，`OnKeyReleased` 时取消运动。问题在于，如果我先后按下多个键，然后释放其中一个，运动将会取消，而玩家还有至少一个手指在按着键位。

解决方案：使用一个列表来维护按键，按下一个键的时候该键入表，松开的时候出表，运动时遍历整个表来计算出运动的方向。

问题：两个人物可以互相推动。

原因：物理引擎原理层面的问题。两个刚体都有相对方向的速度。

解决方案：无解，只能通过碰撞时设置无弹性，但是由于刚体特性，如果这时一个玩家按下运动键，他依然会获得一个速度，由于两者都



是刚体，另一个玩家会被推动。

问题：tileMap 的位置映射不对，玩家以为的脚下的地块和程序中的脚下有时不会重合

原因：二维表示三维空间，又无透视，导致看的很别扭。

解决方案：重新设置锚点。

六、项目总结

本次项目我们最重要的收获是对游戏开发的流程更加清晰，从确定玩法，分工合作，使用 git 工具协同工作到测试功能整个流程都完整地走了一遍。最大的遗憾是代码风格有些抱歉，双玩家代码没能复用。还有就是网络联机的部分实在超出目前的水平，没能实现。

在整个项目的实现中，我们对实现方案进行了许多次的迭代，主要原因是开始的太早，很多知识之后才接触到。比如开始实现行走区域的限定的时候，使用的是试探并直接限定的方法，后来学到物理引擎的知识，就改用了物理引擎。很多之前难以解决的 bug 消失了（当然也带来了新的 bug）

虽然我们这次用的知识都没有超出课程范围，但是在解决各种类型 bug 的过程中也学到了很多新的知识。比如物理引擎的刚体部分的一些实现细节。

总体来说，这次作业非常愉快，学到很多东西，辛苦老师和 TA 啦~