

18组项目作业-仿王国保卫战说明文档

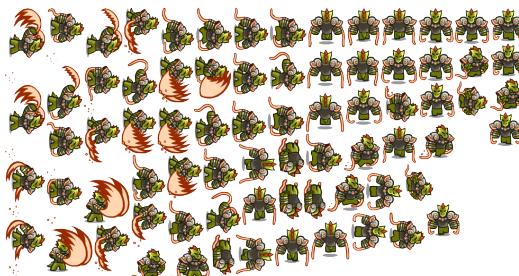
1 设计理念

选择对王国保卫战(KingdomRush，以下简称KR)开发一个仿制项目源起于我们对于王国保卫战这款经典塔防游戏的喜爱，从而想通过对该游戏的仿制获得一次完整的游戏开发实践经历与代码编写能力的训练。

本项目肇始于大一下高级程序设计课程项目。当时，魏良绪已尝试利用Qt对KR进行仿制(可参见我们的GitLab代码仓库¹)，但是发现了极大的开发困难，原因如下：1.KR的资源文件图片普遍是复合的，即一张贴图文件中包含了多个帧状态图片，从而在开发时需要不断进行手动裁剪(已在下文给出示例图片)；2.Qt的相关库过于复杂。Qt新手在进行游戏开发时往往会淹没在其中，并且由于Qt并不是一个专门应用于游戏开发的工具，想利用它来开发KR这样的较为复杂的塔防游戏非常吃力。在高程的项目实践中，魏良绪花费了大量时间才完成了主界面的开发。



(a) 高程项目的开发成果:游戏主界面



(b) KR资源文件中的贴图文件示例

图 1

因此，要完成该项目的开发，我们迫切需要能够解决以上两个困难的方法。幸运的是，通过调研，我们找到了能够从一张贴图文件中加载大量帧状态图片的游戏引擎:Cocos2d，并寻找到了可利用的资源文件以及坚实的开源项目基础²，从而开始了我们的项目开发。

¹https://git.nju.edu.cn/Amazarashi/myproject_kr

²<https://github.com/exmex/KingdomRushFrontiers>

2 游戏借鉴、区别与贡献

2.1 项目借鉴以及对于原项目的理解

项目完全借鉴自王国保卫战：前线(KingdomRush:Frontiers)。由于项目的庞大体量以及新引擎的上手难度，我们选择首先基于已有项目³进行逐文件的代码复现和学习，再尝试自主进行对原项目的完善以及功能扩展实现。我们在十月到十一月逐步完成对原项目的成功复现和学习⁴，并在此过程中深度了解了Cocos2d的相关使用方法、项目的层级结构、游戏中怪物与士兵的状态机流动机制、Cocos2d的渲染方式、父子节点机制等等，高度把握了原项目的代码架构。在复现、学习的过程中，我们认为原项目接近底层的实现(例如与ui绘制相关的Scene文件夹中的类、在每关通关后管理存档内容更新并在每关卡中读取怪物的BaseMap类等)已相当成熟并且与Cocos2d的机制高度绑定，没有必要再进行重构优化。但是，我们发现了原项目的一系列游戏逻辑上的缺陷(例如伤害计算机制没有真正考虑单位护甲——相当于直接废弃了护甲机制；每波怪物刷新时间间隔过短难以处理等)、资源文件上的不足(例如防御塔资源文件中对于防御塔的介绍严重错误：塔名不匹配、数值不正确，怪物资源文件同样如此等等)、游戏存在bug(例如按照原项目总计十三关的设定，当通关最后一关时，选关界面中最后一关的旗帜并不播放等)；并且，我们认为原项目仍有相当多的内容进行扩展实现(参见下文我们所做的扩展实现)。从而，在十一月至十二月间，我们开始了对于原项目的完善以及自主进行新功能的实现。

2.2 区别与贡献

首先，KR原游戏是基于lua语言编写的，而我们的项目完全基于cpp编写。

名称	修改日期	类型	大小
_assets	2021/12/7 4:13	文件夹	
all	2021/12/7 4:13	文件夹	
all-desktop	2021/12/7 4:13	文件夹	
kr1	2021/12/7 4:13	文件夹	
kr1-desktop	2021/12/7 4:13	文件夹	
lib	2021/12/7 4:13	文件夹	
conf.lua	2021/12/7 4:13	Lua 源文件	1 Ki
features.lua	2021/12/7 4:13	Lua 源文件	1 Ki
Kingdom Rush.zip	2025/6/23 17:08	ZIP 文件	349,421 Ki
log_levels_release.lua	2021/12/7 4:13	Lua 源文件	1 Ki
main.lua	2021/12/7 4:13	Lua 源文件	19 Ki
main_globals.lua	2021/12/7 4:13	Lua 源文件	1 Ki
version.lua	2021/12/7 4:13	Lua 源文件	1 Ki

图 2: 原KR游戏的文件结构

其次，我们对于原开源项目进行了全方位的完善以及扩展，内容如下：

³我们的基础项目：<https://github.com/exmex/KingdomRushFrontiers>

⁴我们对原项目进行复现时的代码推送记录仓库：https://github.com/AmazarashiEnduerRain/KingdomRush_Stu

2.2.1 资源文件方面

1. 修复了原项目中的防御塔与怪物资源文件(tower.xml和creep.xml)，在点开游戏的百科全书界面时，可以在防御塔和敌人界面阅读到正确的名字、正确的数值；
2. 修复了各关的介绍文件(level0-12_description.xml)，现在进入到关卡的难度选择界面时，可以看到正确的关卡描述；
3. 修复了原项目中第十关钢铁模式(level9_2_monsters.plist)、第十一关英雄模式(level10_1_monsters.plist)怪物路径存在的问题，原项目中对怪物路径的设置发生越界导致游戏在游玩过程中发生崩溃。我们增补了第七关钢铁模式的怪物配置文件(level6_2_monsters.plist)，原项目缺失了该文件导致第七关钢铁模式无法游玩；
4. 完善了各关的怪物配置文件(level0-12_0-2_monsters.plist)，进行了怪物量与怪物波数的扩充。原项目的每关仅仅设置了两波怪物，这样的怪物量过少；
5. 增补了地下部分关卡的战斗背景音乐。原项目缺失了地下部分战斗背景音乐，并在地下关卡调用了沙漠的战斗背景音乐。我们在Resources的sound文件夹中增补了地下部分背景音乐并在SoundManager中补充了接口，并在level11、12中调用了正确的背景音乐(参见level11、level12)。

2.2.2 代码方面

1. 修复了翼龙动画加载缺失的问题(MyAnimation)。现在，游戏不会如原项目那样在翼龙怪物刷新后发生崩溃。另外，删除了原项目中没有得到使用的冗余文件(如MagicIcon)；
2. 修复了进入第一关时开始动画结束后继续按钮点击无效的bug (BeginAnimation)；修复了原项目防御点击范围的bug (BaseTower)，原项目中的防御塔无条件响应触摸，导致建造防御塔后周围的地形无法再继续建造；修复了游戏失败后重新开始按钮与退出按钮点击无效的bug(Failure)；
3. 修复了第十三关通关后不播放旗帜胜利动画的bug(GameView);并修复了显示第十四关选关旗帜导致游戏崩溃的bug(GameView)；
4. 一定程度上修复了显示层级的问题(Terrain)。我们重设了地形的Z轴顺序，确保防御塔显示在前面，而不会如原项目那样被后方的怪物遮挡；
5. 使用了更优化的索敌算法(BaseTower)，提升了游戏的流畅度。原项目中，防御塔持续暴力搜索最近的怪物，导致了游戏的严重卡顿。我们将防御塔的索敌优化为当前目标存活的情况下不再进行搜索而是持续攻击已锁定的目标，从实现的效果来看也更加贴近原版；
6. 一定程度上重写了位置获取方式(BaseMonster和BaseTower的getActualPos函数)，使得位置获取更加明确；手动编写一系列get、set函数，避免CC_SYNTHESIZE宏展开后带来的隐式成员层级变换问题。我们在复现过程中发现，防御塔作为地形的子节点，而地形作为地图子节点，实际坐标要通过父节点获取，从而通过this->getParent()->getPosition()获取位置；而怪物刷新后，其自身实际坐标没有移动，而是其贴图精灵baseSprite在移动，因此获取怪物位置需要通过this->baseSprite ->getPosition()获

取。这些位置获取方式可读性差。另外，由于宏展开的问题，部分类存在貌似被设定为protected或private的成员被隐式变换为了public成员；

7.重新设定每波的出怪间隔，由20s提升至40s，给予更多处理时间(WaveFlag)；按照官方资料重设了防御塔、怪物数值(参见Monster、Tower文件夹中的所有代码)，并在TowerInfoLayer中进行了相应的显示内容的修改；

8.完全重构了伤害计算方式(参见Bullet、Soldier、Monster文件夹中的所有代码)。原项目中几乎没有考虑护甲对于伤害的影响或是脆弱地使用伤害数值减去单位的护甲值以计算最终伤害。我们更贴近原版地将护甲设定为伤害抵消因子，按比例计算最终伤害，同时避免了高甲怪物越打血越多的情况。并且，我们重设了所有单位的护甲数值；

9.重设了防御塔的造伤机制(参见Tower文件夹中的所有代码)。原项目中的防御塔是通过创建子弹、设定子弹数值与动作，并让子弹最终完成对于怪物的造伤的。但我们将原项目脆弱地硬编码了防御塔对自身子弹伤害的数值设定，从而无法应用防御塔的升级逻辑。我们将其修改为防御塔以自身攻击力设定子弹数值；

10.我们完全重构了援助士兵的相关逻辑并创建了新的援助士兵(参见ReinforceSoldier1、2以及我们新创建的ReinforceSoldier3)。原KR游戏中，援助士兵技能可以召唤三个援助士兵，并且援助士兵可以随着科技升级改换外观、数值和攻击方式；但原项目的实现仅仅能够召唤两个援助士兵，且援助士兵不会随着科技升级进行外观、数值和攻击方式的转变。我们重构了援助士兵的状态机，并重构了Arrow子弹类和在BaseSoldier中添加新的状态，使得点出最高等级科技后的援助士兵能够在远程射击和近战阻挡两种攻击方式间切换并播放正确动画；我们重构了援助士兵的外观设置以及对于他们相应动画的加载方式(MyAnimation)，并为他们实现了科技树升级逻辑，现在他们可以动态加载外观、数值、动画以及攻击方式。我们创建了新的援助士兵实例ReinforceSoldier3并为他加载了相关动画资源。我们修改了TouchLayer中援助士兵的召唤逻辑，现在可以召唤三名援助士兵。并且，我们为援助士兵实现的远程攻击逻辑与近战逻辑完全解耦。经过实机测试，新的援助士兵相关逻辑运行正常；

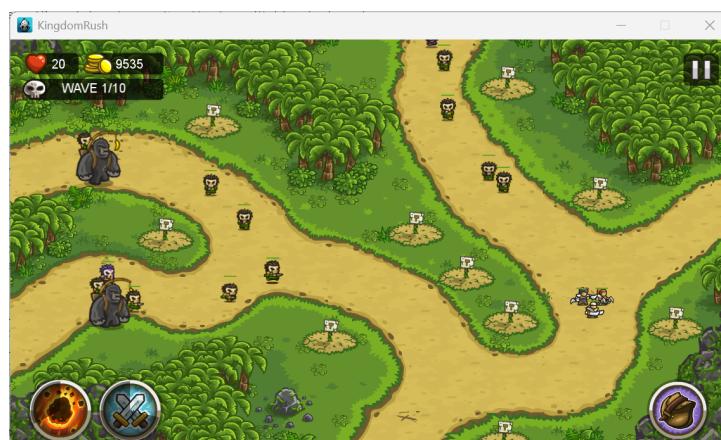


图 3: 所实现的召唤三个援助士兵；且援助士兵随着升级可以改换外观、获得远程攻击能力

11. 我们完整实现了科技树升级逻辑，现在游戏的升级界面能够正常运作。原项目的升级界面仅仅是贴图而没有实装功能，而我们通过大量修改实装了升级系统(参见各士兵文件、各防御塔文件，各子弹文件，TouchLayer文件，PlayerStateMenu文件，各怪物文件等)。现在，随着科技树界面的升级，箭塔的攻击范围、伤害以及建造花费，法师塔的伤害与攻击范围，炮塔的攻击速度、攻击伤害等，火雨的伤害和陨石数量等，援助士兵的数值、外观、攻击伤害等会真实发生改变。尤其是，现在升级后的火雨技能会真正获得陨石数量的增加，并在地面真实地产生岩浆(Lava)实体持续灼烧敌人；并且，我们引入了最高级炮塔科技所带来的震晕(Stun)机制。而且，在科技树升级后，火雨和援助士兵的技能冷却时间会真正缩短；

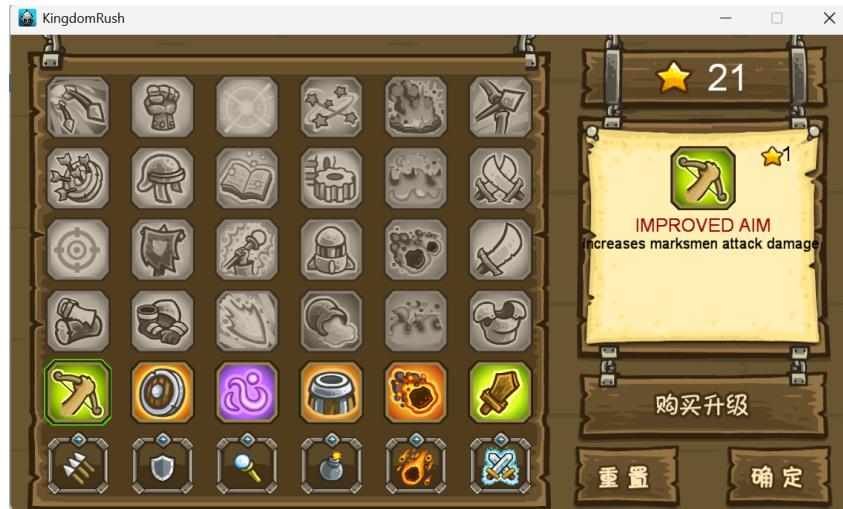


图 4: 游戏升级界面。现在，升级系统真正实装了

12. 我们在GameView中实现了选关界面的英雄按钮与相关动画。原项目中，下方按钮栏仅有商店、升级和百科全书。我们希望实现点击英雄按钮后进行相应的场景切换，但由于资源文件缺乏我们未能实现。



图 5: 新的游戏选关界面

13. 实现了新怪物：SaurianDarter和SaurianSavan，这是原项目中所没有的怪物。

我们在MyAnimation中加载了相关动画资源，创建了两个新类，并在BaseMonster中注册了新的怪物类型。我们修改了MonsterInfoLayer，以在点击时能够呈现新怪物的相关信息。新怪物的实机测试表现良好，各状态下的动画均没有问题；

14. 我们为配合炮塔的升级机制，为怪物实现了震晕动画、震晕状态，并且重新设定了怪物对士兵的造伤方式。我们在BaseMonster中为怪物新增了震晕状态stateStunned，并添加了震晕的动画精灵。当怪物被炮弹震晕时，怪物的头顶会出现环绕的星星视觉效果。并且，在怪物被震晕时，它们不会对士兵造成伤害；

15. 我们实现了t200(即机甲塔)的炮弹爆炸的特效文字动画。原项目中的MechaBomb未被真正加以使用，我们将Mecha的子弹调用与bomb类解耦，转而使用自身的专属子弹MechaBomb。我们在MyAnimation中实现了工厂模式的特效文字跳动淡出动画，并让MechaBomb在爆炸时对显示了KBOOM的特效文字动画；

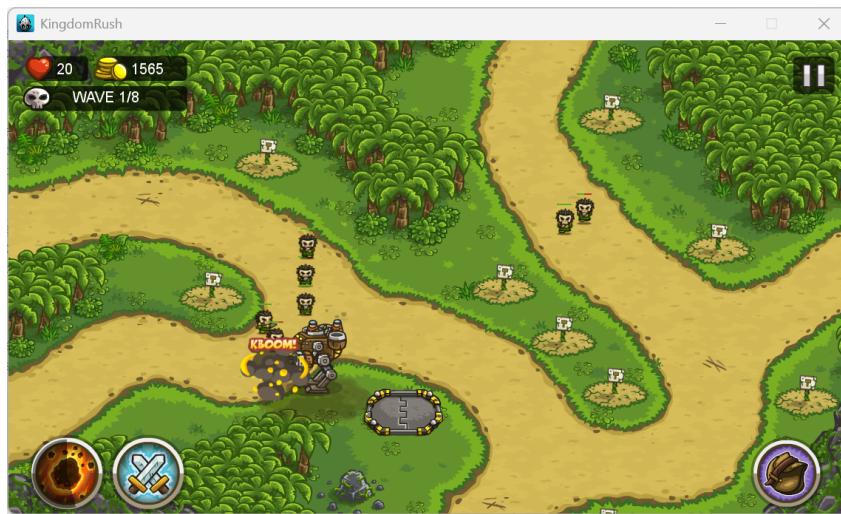


图 6: 新的特效文字动画

16. 我们实现了复仇女神塔(CossBowHunterTower)射杀敌人时的特效文字动画，并按照官方资料重设了复仇女神塔的技能伤害数值。复仇女神塔的特效文字动画分为普通击杀(SHUNT)与技能击杀(AACK)两类。我们将复仇女神塔的子弹调用与Arrow类解耦，创建新的子弹类CossBowArrow并让复仇女神塔调用自身专属的子弹，从而正确实现了它的特效文字动画。我们优化了它的技能触发机制，原游戏中复仇女神塔会在每十次攻击时触发连射技能，而原项目设定每三下普攻即触发技能；

17. 我们重设了地震塔的钻头(Drill)逻辑。原游戏的钻头对于敌人一击必杀并且会播放特效文字动画，而原项目的钻头仅对敌人造成100伤害且没有相关动画。并且，原项目的地震塔当攻击范围内不存在敌人时会持续释放钻头。我们在EarthquakeTower中加入了钻头的冷却时间计时器，并为钻头实现了特效文字动画SPLAT，重设钻头为一击秒杀并且只有在完成冷却后才能释放；

18. 我们为士兵、大法师塔、死灵法师塔、部落掷斧手塔均实现了攻击时的特效文字动画(请参见所有士兵类、ArchMageBolt、NecrBolt、TotemAxe)。现在，大法师塔在击杀敌人时会出现ZAPOW! 动画，死灵法师塔在击杀敌人时会出现ZISHH! 动画，

士兵(除刺客Assassin)在交战时会概率触发模拟兵器声音的SOK!或POW!动画，部落掷斧手塔在击杀敌人时会出现THUNK的特效文字动画；

19.引入炮弹(bomb)的溅射伤害机制，适配炮塔的升级逻辑，现在炮弹范围内敌人造成伤害时会更加精确地计算溅射伤害。MechaBomb中也引入了该机制；当点出炮塔的相关科技后，炮塔将无视溅射距离，对范围内的敌人造成一致伤害；

20.我们对炮塔引入装填时间(reloadTime)逻辑(参见BaseArtillery)，从而能够通过炮塔科技升级加快攻速(SimpleArtillery的三类中，以reloadTime频率调用攻击)；我们完善了炮塔的动画时序配合(原项目中开火动画与炮弹飞出不匹配)，现在炮管的动画与炮弹的飞出更加一致；

21.我们实现了level12中背景的巨龙(参见level12)及其相关动画(参见MyAnimation)。巨龙会在背景中呈现出睡觉的特效文字动画，且每隔一段时间会有打呼噜和吐息的动画效果。原游戏中，此处的巨龙会飞起来喷吐龙焰并对防御塔造成影响。我们视图实现，但需要的相关技术过于复杂于是搁置。

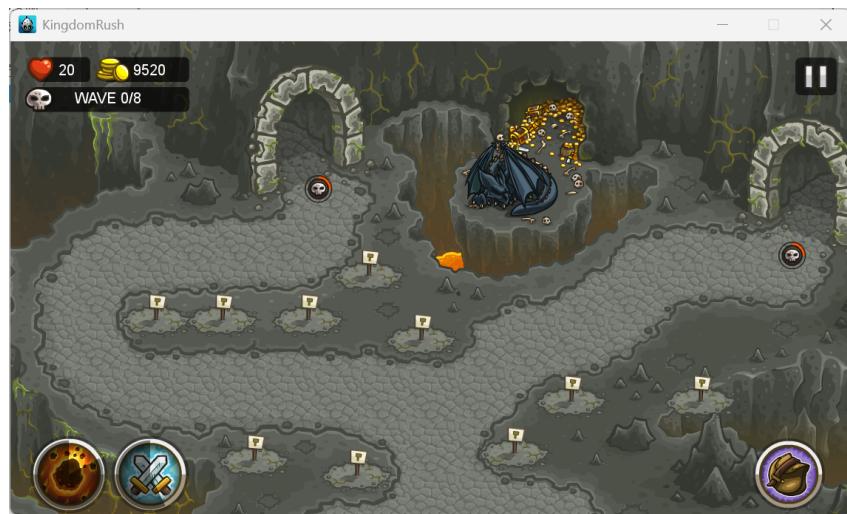


图 7: level12背景中的巨龙

22.我们优化了原项目中在建造塔时塔所显示的预览范围圈显示以及在对塔进行升级或高级塔升级时下一级的预览范围圈显示(参见updateTowerPanleLayer与updateTowerPanleLayer2以及TowerInfoLayer)。原项目中脆弱地硬编码下一级塔的范围为当前范围 + 20或是 + 40，带来了ui显示与实际范围不符的问题。我们在BaseTower中引入了nextScope1、2成员，重设下一级塔预览显示范围为nextScope，这样的改动也适配了升级逻辑；我们在TowerPanleLayer中进行了修改，贴近原版重设防御塔建造花费并让建造金钱消耗与升级逻辑适配；

23.我们增添了level13中的特殊塔矮人堡垒(参见Tower中创建的新文件DwarvenBastion)的实现并在MyAnimation中加载了其相关动画资源。我们创建了新的子弹Barrel(火药桶)来实现矮人堡垒的技能攻击逻辑(每过10s扔出火药桶炸伤敌人)，在BaseTower中注册了新的塔类型BASTION，在LoadingScene中增添了新的plist文件读取，修改了TowerInfoLayer以显示矮人堡垒的相关信息。我们微调了矮人堡垒上方

矮人火枪手的站位与扔出炸药桶的动画时序，实现了矮人火枪手开火时的枪口烟雾。现在经过实机测试，矮人堡垒的相关动画与逻辑运行正常；

24.我们增添了level13中的特殊塔矮人兵营的特殊士兵矮人勇士(Soldier中创建新类DwarfWarrior)，的实现并在MyAnimation中加载了其相关动画资源。我们完整实现了矮人勇士的攻击逻辑，其在技能冷却完成且血量低于一百时会触发饮酒技能并回血。饮酒后矮人勇士身上的泡泡特效与脚下的光晕特效动画已成功实现。经过实机测试，矮人勇士的相关动画与逻辑运行正常；

25.我们增添了level13中的特殊塔矮人兵营(参见Tower中创建的新文件DwarfHall)的实现，并在MyAnimation中加载了其相关动画资源。我们完整实现了矮人兵营的相关逻辑，包括开门动画与对矮人勇士的治疗和复活等。我们在BaseTower中注册了新的塔类型HALL，并在TowerInfoLayer中进行修改以显示矮人兵营的相关信息。经过实机测试，矮人兵营的相关动画与逻辑运行正常；

26.我们在Layer中增添了新类RingPanleLayer的实现。这是为矮人堡垒这一特殊塔创建的新的菜单实例。矮人堡垒不可出售，因此RingPanleLayer仅显示塔的范围圈与圆环菜单背景；

27.我们新引入了英雄类单位(参见Hero文件夹)。我们创建了英雄基类(BaseHero)，其具有三个技能状态，且具备相应的跑动、攻击、索敌等行为逻辑，具备自身的视觉精灵。由于英雄是直接添加在地图层上的，且有特殊的重生、死亡逻辑，我们为英雄单位加入了birth、die等函数接口，并加入重生计时器；由于英雄单位可以被选中并进行全图移动，我们为他加入了选中环的视觉精灵并为他绑定了自身专属的监听器；

28.我们增添了level13中的特殊英雄Rurin Longbeard的实现(参见Hero文件夹中的RurinLongbeard)。我们在BaseHero的HeroName枚举中为其注册了姓名，在MyAnimation中加载了其相关的动画资源，并完整实现了其行为逻辑。他会在每七次普攻后发动锤击技能造成真实伤害，并在非战斗状态下每秒回血42。在选中他时，他的身上会出现选中圈的反馈提示。经过实机测试，其相关动画与行为逻辑运行正常。RurinLongbeard的成功实现意味着我们对于英雄基类的成功实现，即英雄基类的相关逻辑已一定程度上足够完善，可以以此为基础开发更多英雄；

29.我们完成了level13的基础实现(原项目中的level13是空文件)。我们在选关界面添加了第十四关的相关旗帜，并确保了第十四关选关旗帜的相关逻辑与动画运行正常。我们编写了新的配置文件和新的关卡介绍xml文件(参见level13的相关怪物plist文件以及xml文件)，并修改了存档槽位和选关界面的星星数显示(原项目由于只设定了十三关，星星总数显示为65，我们修改为70)；

30.我们完成level13的更完善的实现。目前level13贴近原版地从开局就拥有两座矮人堡垒塔和一座矮人兵营，并拥有可供调遣的矮人英雄。并且，我们在level13中贴近原版地实现了洞穴处的岩石遮挡，并设定在第三波怪物来袭时洞穴的岩石会碎裂并开始刷怪(我们在MyAnimation中加载了相关动画)。

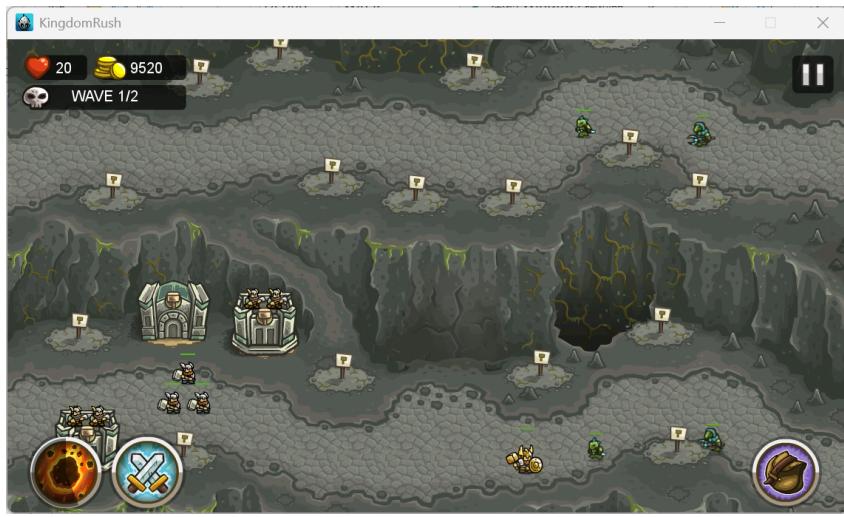


图 8: level13的关卡概览。图中展示了我们实现了的两类新怪物、洞穴处的岩石遮挡、矮人堡垒塔、矮人兵营塔以及矮人英雄单位

3 团队分工

魏良绪（队长）：

- 初期开发尝试，对游戏引擎的选择与评估。
- 对于开源项目进行深入的学习、理解与成功复现，并对原项目缺陷进行完善和修复。
- 修复了防御塔图鉴的相关资源文件(tower.xml)，和原项目中配置文件的缺陷。
- 统筹安排，跟踪管理进度；整个项目的分工。
- Resources的搜集与获取。
- 完成项目扩展实现代码方面的开发，思考并确定各新实例的具体逻辑。
- Github仓库的维护，分支管理与合并，还修复了上传Github遇到的奇奇怪怪的许多问题。
- 项目报告的完成；最终的展示、演示与答疑。

李想：

- 项目初选择的探讨。
- Monster中Jungle和Underground部分的怪物的部分代码实现。
- 后期代码bug的排查与修改。
- 协助队长对Github仓库的管理。

蒋成康：

- 项目初选择的探讨。
- 视频的构思、剪辑与制作。
- 负责矮人英雄相关资源确定。
- 主要负责游戏的实机测试检验，并进行代码bug排查。

郭鑫：

- 项目初选择的探讨。
- 视频的构思、剪辑与制作。
- 后期代码bug的排查与修改。
- 负责了游戏level5到9的怪物配置文件，补充了地下关卡的背景音乐。

刘可一：

- 项目初选择的探讨。
- 视频的构思、剪辑与制作。
- 后期代码bug的排查与修改。
- 负责矮人勇士相关资源确定。
- 对防御塔、怪物等参照官方资料进行数值核验，优化了防御塔范围圈的ui显示。
- 次要负责游戏的实机测试检验，并进行代码bug排查。

高致远：

- 项目初选择的探讨。
- 负责特效文字、矮人堡垒、矮人兵营的相关资源确定。
- 后期代码bug的排查与修改。
- 报告文档的协助撰写。

周梓昱：

- 项目初选择的探讨。
- 报告文档的协助撰写。
- 负责了游戏level0到4的怪物配置文件，完善各关介绍的xml文件。

- 后期代码bug的排查与修改。

焦子恒：

- 项目初选择的探讨。
- 负责游戏level10到13的怪物配置文件。
- 负责了新怪物的相关资源确定，核对修改了怪物图鉴相关资源文件creep.xml。
- 后期代码bug的排查与修改。