**3D弹幕游戏开发——需求分析**

1. **需求导出**

**1.1利益相关者**

1.玩家：游戏的使用者

2.游戏设计者：对游戏内容、玩法的设定

3.程序实现者：负责游戏程序的编写

**1.2运行环境及约束**

操作系统：Windows 10

**1.3需求描述**

用户在主界面按下任意键开始游戏。游戏开始后，敌人会自动生成弹幕。玩家需要通过键盘和鼠标的输入来控制游戏中的人物角色的移动，躲避弹幕。当角色与弹幕碰撞时，生命值-1；当生命值为0或超过时间后，游戏结束。显示器显示游戏画面，包括游戏中的玩家、弹幕、敌人以及背景环境的渲染。扬声器播放游戏音效，包括背景音乐、弹幕产生、撞弹音效。

通过对上述游戏流程的分析，可以得到四个列表：

对象列表：玩家角色、敌人、弹幕、键盘、鼠标、显示器、扬声器

服务列表：弹幕生成、输入检测、碰撞检测、画面渲染、音效渲染

约束列表：项目在12月底之前完成

性能列表：画面帧率要达到20fps以上

1. **界面原型**



图2.1 游戏进行中的界面原型



图2.2 游戏进行中的界面原型

1. **分析建模**

**3.1用例图及用例**



图3.1 用例图

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1 |
| 用例名称 | 开始游戏 |
| 用例说明 | 用户在主界面按下任意键，开始游戏，进入游戏画面。 |
| 参与者 | 玩家 |
| 前置条件 | 1.程序在主界面 |
| 后置条件 | 1.弹幕生成  2.输入检测  3.碰撞检测  4.画面渲染  5.音效渲染 |
| 基本路径 | 用户在主界面按下任意键 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1.1 |
| 用例名称 | 弹幕生成 |
| 用例说明 | 游戏过程中，电脑自动生成不同类型的子弹，形成弹幕，向四周发射。子弹类型有粒子型与激光型，运动方式包括固定弹和自狙击。玩家需要控制角色来躲避这些弹幕。 |
| 参与者 | 玩家 |
| 前置条件 | 1.游戏开始 |
| 后置条件 | 1.碰撞检测  2.画面渲染  3.音效渲染 |
| 基本路径 | 游戏开始后，自动执行 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1.2 |
| 用例名称 | 输入检测 |
| 用例说明 | 游戏过程中，检测用户输入。用户可以通过鼠标改变角色朝向，并使用键盘按键WASD进行前后左右的移动，以躲避弹幕。 |
| 参与者 | 玩家 |
| 前置条件 | 1.游戏开始 |
| 后置条件 | 1.角色位置信息更新  2.碰撞检测  3.画面渲染  4.音效渲染 |
| 基本路径 | 用户移动鼠标、用户按下键盘上的WASD |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1.3 |
| 用例名称 | 碰撞检测 |
| 用例说明 | 通过计算角色与每个子弹的距离，以及子弹的类型大小，判定角色是否与弹幕发生体积碰撞，若发生碰撞，角色的生命值-1，当角色生命值为0时，游戏结束。 |
| 参与者 | 玩家 |
| 前置条件 | 1.游戏开始 |
| 后置条件 | 1.更新角色生命值属性  2.判定游戏是否结束  3.画面渲染  4.音效渲染 |
| 基本路径 | 游戏开始后，循环执行 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1.4 |
| 用例名称 | 画面渲染 |
| 用例说明 | 将游戏画面渲染出来，并通过屏幕显示。渲染的元素包括角色、弹幕、敌人、环境等。 |
| 参与者 | 玩家 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本路径 | 程序运行状态下，循环执行 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.1.5 |
| 用例名称 | 音效渲染 |
| 用例说明 | 将游戏音效渲染出来，并通过扬声器播放。渲染的元素包括游戏背景音乐、弹幕发射的音效、角色中弹的音效等。 |
| 参与者 | 玩家 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本路径 | 程序运行状态下，循环执行 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.2 |
| 用例名称 | 暂停游戏 |
| 用例说明 | 游戏过程中，检测用户输入。用户可以按下`ESC`键，暂停游戏，并出现选择菜单，选项包括继续游戏、回到主界面。 |
| 参与者 | 玩家 |
| 前置条件 | 1.游戏开始 |
| 后置条件 | 1.画面渲染  2.音效渲染 |
| 基本路径 | 用户按下`ESC`键 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例编号 | UC.3 |
| 用例名称 | 退出游戏 |
| 用例说明 | 用户在主界面按下`ESC`键，退出游戏。 |
| 参与者 | 玩家 |
| 前置条件 | 1.程序在主界面 |
| 后置条件 |  |
| 基本路径 | 用户按下`ESC`键 |
| 扩展路径 | 无 |
| 补充说明 | 无 |

**3.2类图**

将需求导出中获得的三个实体类：玩家角色（player）、弹幕（danmaku）、敌人（enemy）抽象得到抽象类GameObject类，它包含了用于渲染游戏元素的基本信息，如坐标、方向、大小、模型等，同时它还含有一个用于绘制游戏元素的接口draw()。将GameObject类的draw()工作委托给专门用于绘制图形的Renderer类来实现。碰撞检测由Collision\_decetor类实现，它通过获取玩家对象和弹幕对象的相关信息进行体积碰撞计算。类之间的关系如下：

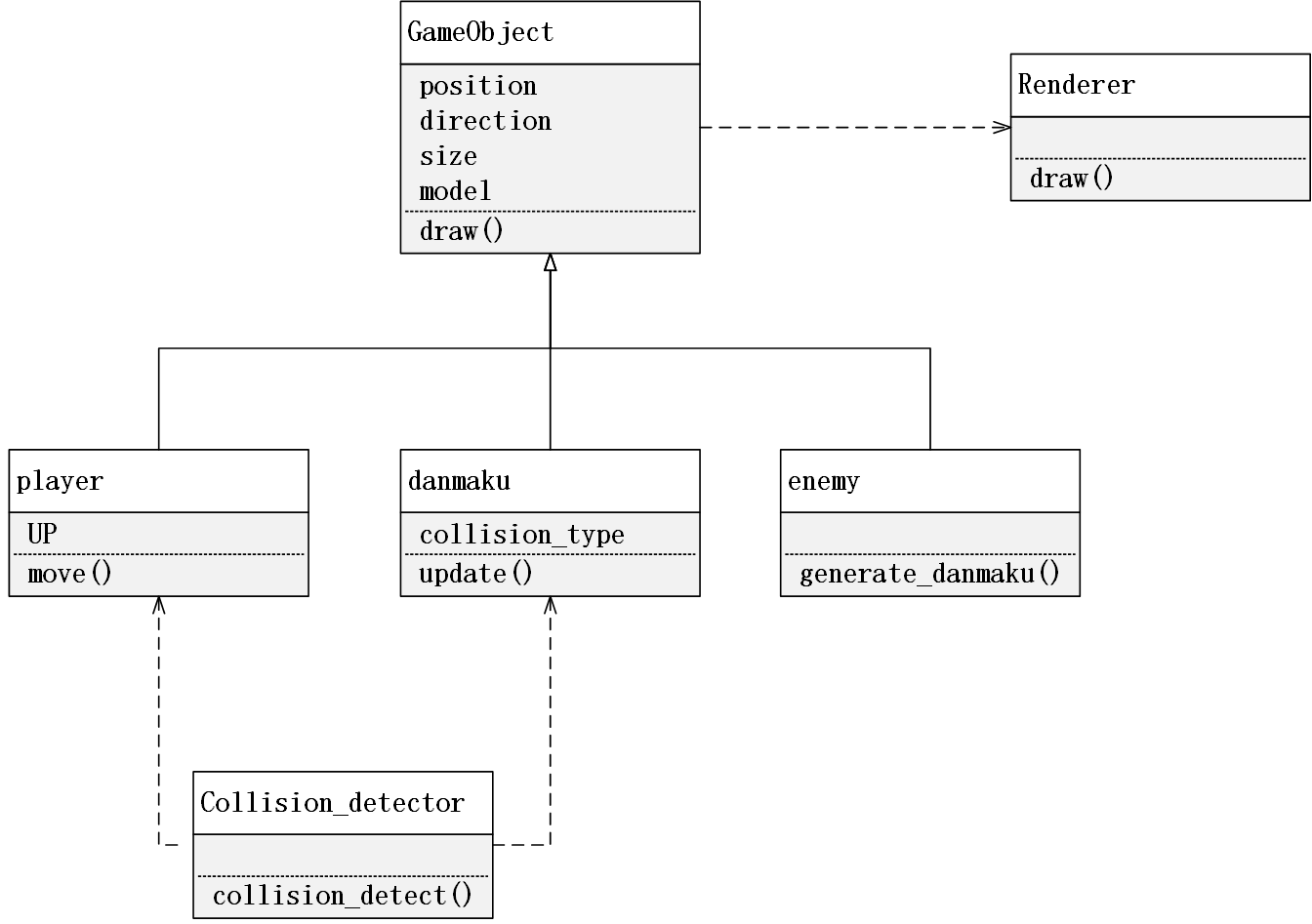


图3.2 类图

**3.3时序图**

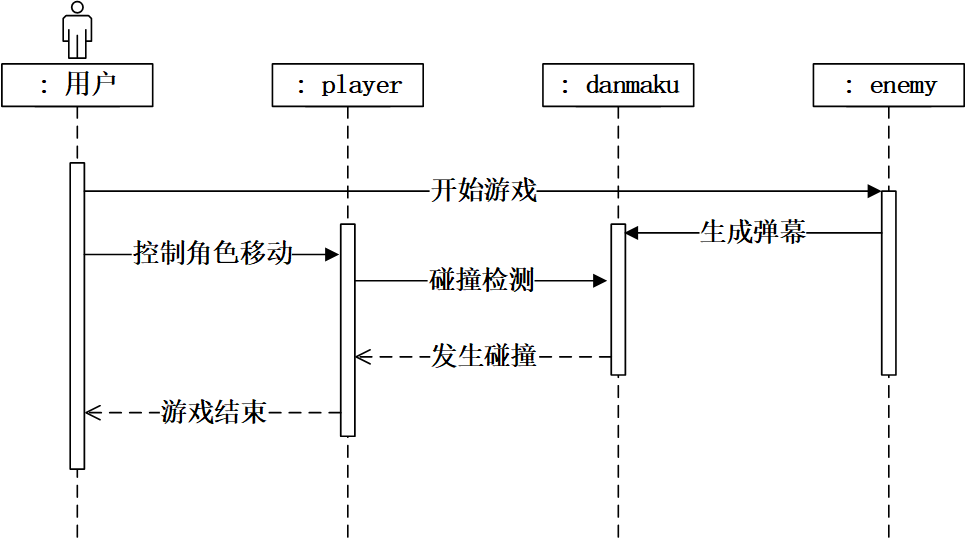


图3.3 时序图

**3.4状态图**



图3.4 状态图