

# Arkmidnight 程序报告

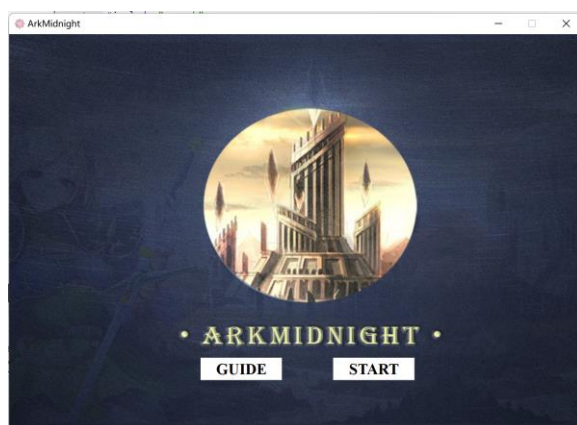
明复自衡队 刘松瑞 林坚铮 李之怡

## 一、程序功能介绍

Arkmidnight 是一款即时制战略塔防类游戏，我们利用 Qt 进行程序的编写，并利用 Qt creator 结合 Photoshop 等软件进行游戏 UI 的设计，实现了多关卡、多回合的人机对战，并设计了种类丰富的“塔”和“怪物”，拓展了游戏的可玩性。

下面就程序各功能进行介绍。

### 1. 主菜单栏



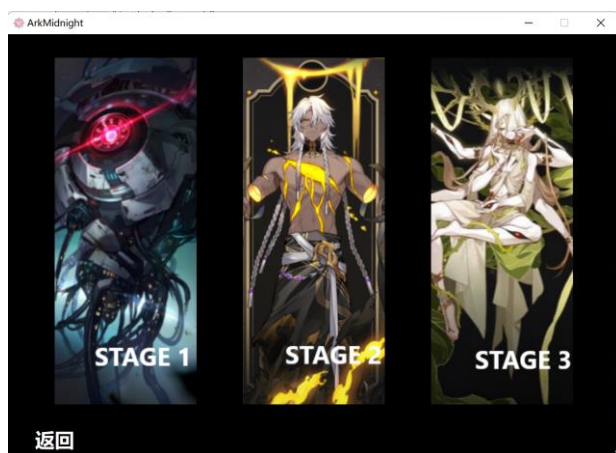
主界面展示了本款游戏的主题。玩家可以在主菜单栏点击按钮，选择进入“GUIDE”（游戏导引）和“START”（开始游戏）。

### 2. 游戏导引



游戏导引分为三页，为新手玩家全面介绍了游戏的玩法，玩家可以通过页面左下角、右下角的“QUIT”、“NEXT”、“LAST”三个按钮在不同页和主界面之间自由切换。

### 3. 选择关卡



在选择关卡界面，玩家可以通过点击相应区域，进入三个不同难度的关卡。每一个关卡包含若干回合。从第一关至第三关，设计了不同初始的地图，难度逐步升高，这样的设计照顾到新手玩家的体验也增添了玩家挑战的乐趣。

## 4. 游戏功能

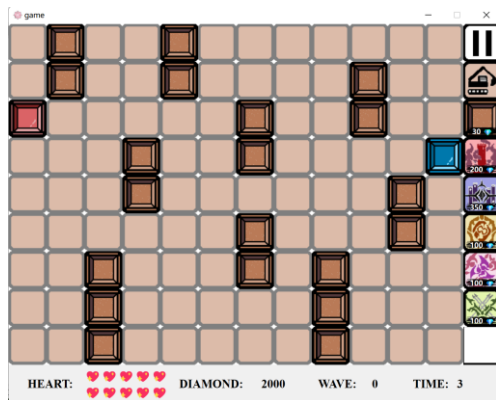
### 4.1 塔的放置

进入游戏界面之后，玩家可以看到初始的游戏地图及商店界面、自己拥有的生命值和钻石数、关卡的用时。在回合开始前 30 s 以及回合进行阶段，玩家都可以用自己的钻石数购买塔，并在地图上的合法位置放置塔。

玩家点击界面右侧商店中欲放置的塔，再点击地图中想要放置的空位，就实现了塔的购买和放置。对于不合法的位置（比如，放置塔的位置堵住了怪物通向终点的路），会提示“不要堵路”；对于钻石数不够的情况，会提示“你没钱了”。

玩家也可以先点击“挖掘机”，再点击地图中的墙和塔，将墙和塔移去。

塔分为攻击类和辅助类。当辅助类的塔放置到地图中后，其周围的攻击塔会显示出相应的标志，例如，“同谐”塔可以提高其附近攻击塔的攻击力。



### 4.2 怪物出击



当回合开始时，各类怪物会从生成点（红色格子）出发，按照程序内部搜索得到的最短路径向终点（蓝色格子）前进。怪物具有一个血条，显示怪物实时的血量，怪物受到附近塔的攻击时血量减少，当血量为 0 时怪物死亡，从地图中消失，玩家获得一定量的钻石；如果怪物到达了终点，则玩家的生命值减少（以 ❤️ 数呈现）。

不同的怪物有各自的技能和特点。例如，有些怪物会炸毁周围的塔，有些怪物可以自己“回血”，而飞行怪会从生成点直线飞向终点。

当玩家的生命值为 0 时，玩家该关卡挑战失败；若该关卡所有回合结束后，玩家仍存余生命值，则玩家该关卡挑战成功。

#### 4.3 游戏的暂停、快进与退出

玩家可以通过游戏界面右上角的暂停键来暂停游戏，关卡计时器也随之暂停。

当玩家处于回合前 30S 时，若已经完成塔的放置，点击暂停键可以提前开始回合；也可以随时退出游戏，返回选择关卡界面。当回合开始后，点击暂停键会显示“战斗还未结束”，玩家可以点击按钮选择继续游戏或退出游戏。



#### 4.4 游戏结束进行结算

当玩家取得关卡的胜利或失败，会进入游戏的结算页面。

若玩家失败，则会跳出失败窗口，并显示玩家在该回合击杀的怪物数和积累的钻石数。点击“确认”按钮，会回到游戏关卡选择界面。



若玩家胜利，则会跳出胜利窗口，并显示玩家在该回合击杀的怪物数、剩余的生命值和积累的钻石数。点击“确认”按钮，会回到游戏关卡选择界面。

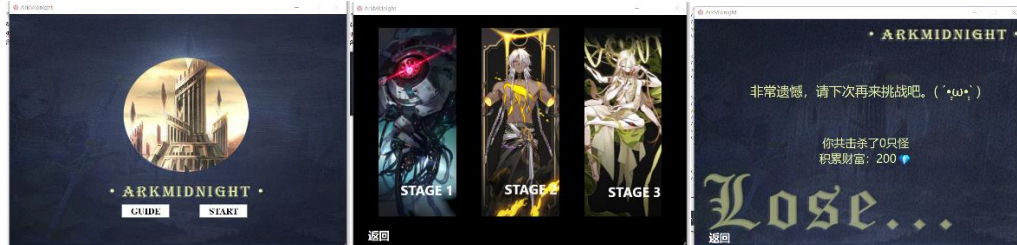
## 二、项目各模块与类设计细节

整个项目分为游戏选择界面与界面跳转部分、游戏游玩进行部分；

### 1. 游戏选择界面与界面跳转部分

#### 1.1 Arkmidnight.h、Arkmidnight.cpp、Arkmidnight.ui:

包含 ArkMidnight 类，其父类为 QMainWindow，其窗口包括以下界面：



其槽函数主要用于页面间切换、页面显示：

void on\_btn\_guide\_clicked(); 跳转到指导界面，新建一个 guide 类并显示，隐藏自身

void on\_btn\_start\_clicked();

void on\_pushButton\_clicked(); 这是跳转到游戏界面，包含关卡数据，新建一个 game 类并显示，隐藏自身

void on\_pushButton\_2\_clicked(); 跳转到游戏界面，包含关卡数据

void on\_pushButton\_3\_clicked(); 跳转到游戏界面，包含关卡数据

void on\_pushButton\_return\_clicked();

void on\_pushButton\_return\_2\_clicked();

void vis(); void vis2(); 用于从 guide1、game 类返回，转到结算页面：

#### 1.2 guide1.h、guide1.cpp、guide1.ui:

包含 guide1 类，其父类为 QWidget，其窗口包括以下界面：



其信号函数用于发出信号，使 ArkMidnight 类显示：

void SignalInGuide\_openArkMidnight();

其槽函数主要用于页面间切换、页面显示：

void on\_btn\_quit\_2\_clicked();

void on\_btn\_last\_to\_guide1\_clicked();

void on\_btn\_next\_to\_guide3\_clicked();等

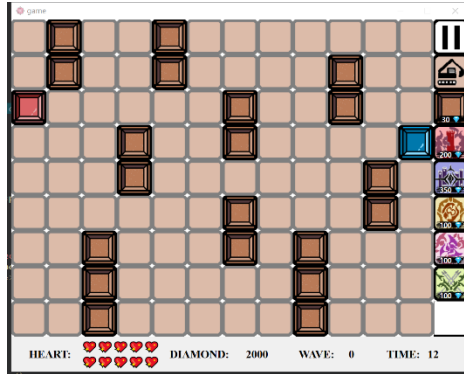
其中 quit 函数用于关闭自身后返回。



## 2. 游戏游玩进行部分

### 2.1 game.h、game.cpp、game.ui:

包含 game 类，其父类为 QWidget，其窗口包括以下界面：



其中屏幕右边一列，第一个方形为一个 QLabel 暂停键，其连接到 nextWave 类的弹窗，属于 game.ui，右边一列剩余的图片方块属于 choice 类；左边所有方块都属于 block 类。

#### 2.1.1 构造函数与析构函数

```
explicit game(class ArkMidnight *parent = nullptr,
              int stage_ = 0,
              std::vector<std::vector<std::vector<int>>> mob_to_spawn_ = {{{}}},
              std::vector<std::vector<int>> Mymap_ = {{}},
              int diamond = 1000,
              int heart = 5,
              std::vector<int> shop_ = {1, 2, 3, 4, 5, 6}
);
```

构造函数：用于初始化数据以及显示上图的各种方块、显示窗口下方的各类数据；stage 标识这是第几大关，mob\_to\_spawn 存有怪物数据，包括何时进场，血量几何等等；Mymap 包含关卡塔初始的分布信息；其他参数是初始血量和钻石数；以及 shop\_ 是这一关可以使用的塔的集合。

#### 2.1.2 塔的建造与寻路

bool inBoard(int x, int y);用于判断一个坐标是否落在左边的方块区域

bool inShop(int x, int y);用于判断一个坐标是否落在右边一列区域

bool getPath(index\* path, int& pathLen, int startx, int starty);

用于寻路，将空 index（index 是包含 x, y 两个 int 型变量的结构体）数组作为参数 path，输入起点的坐标，利用广搜找到一条从起点到 game 这个类的终点，返回 1；若未找到则返回 0。

#### 2.1.3 塔的摧毁与放置

void mousePressEvent(QMouseEvent\* event);

用于检测鼠标事件并响应，若落在右边商店区域则标记，之后落在左边方块区域则调用函数进行摧毁或者放置。当堵住怪物的路、没有钻石时创建 attentions 类进行提示。

bool destroy\_update\_and\_check\_and\_display(int y\_i, int x\_i, int type);

用于玩家摧毁塔，摧毁塔需要检测方块是否可摧毁等，如果塔有其他效果要进行删除，怪物的路线也需要重新寻路；用于自爆怪物摧毁塔，炸毁塔需要检测方块是否可炸毁，其他需要处理的事同上。

void build\_display(int y\_i, int x\_i);用于放置塔时方块的显示以及增益效果更新

bool build\_update\_and\_check(int y\_i, int x\_i);用于放置塔时改变怪物的路线

### 2.1.4 随时间进行更新游戏行为

void onTimeOut();

用于在每个单位时间点调用下列函数，以及游戏结束、胜利判断和相关函数调用。

void protect\_tower();用于显示塔保护、增益的效果

void spawn\_mobs();生成怪物函数

void march\_mobs();怪物前进函数

void tower\_shoot();每隔一段时间塔射击函数

void bullet\_move();子弹移动函数

void laser\_emit();每隔一段时间激光发射函数

void game\_lose();游戏失败，跳转到游戏结算界面

void game\_win();游戏胜利，跳转到游戏结算界面

void mob\_dead(Monster\* mob);怪物死亡，判断血量，将其删除

void heal\_mobs();丰饶塔每隔一段时间会给周围怪物回血

void heal\_mobs2();回血怪每隔一段时间会回血

void bomb\_block();怪物爆破塔，用于 boss，每隔一段时间炸毁全屏的塔

### 2.1.5 其他函数：

void closeAnd(); 用于关闭并转到主界面

void closeAnd2(); 用于关闭并转到主界面

void putOnWaveNumber(); 用于显示波数在屏幕上

void heartShow(); 用于显示血量在屏幕上

void continueTim(); 用于继续计时

void on\_pushButton\_clicked(); 用于暂停键

## 2.2 nextwave.h、nextwave.cpp、nextwave.ui:

包含 nextwave 类，其父类为 QWidget，其在游戏中使用的窗口包括以下界面：



其信号函数：

void SignalInNextWave\_WavePutOn();用于 game 将回合数更新显示

void SignalInNextWave\_TimC();用于 game 重新开始计时

void SignalInNextWave\_Close();用于“不玩了”的选项，将 game 关闭并回到主界面

void SignalInNextWave\_Close2();用于“不玩了”的选项，将 game 关闭并回到主界面

下列槽函数调用信号实现各种功能：

void on\_pushButton\_clicked();

void on\_pushButton\_3\_clicked();

void on\_pushButton\_2\_clicked();等

## 2.3 choice.h、choice.cpp:

包含 choice 类，其父类为 QLabel，其用于 game 窗口右边一系列的商店显示以及选择商店时的效果显示。

## 2.4 monster.h、monster.cpp:

包含 monster 类，其父类为 QLabel:

Monster(class game\* parent, int hp\_, int spd, int x,int y);

基类以及派生类构造函数用于初始化怪物数据

`bool reachEnd(int x, int y);`用于判断此怪物是否到达终点  
`void showHp();`用于在怪物下面展示画出怪物血条  
`void (Flying_)Monster::forward();`用于怪物在地图上前进  
`void adjust();`用于怪物重新寻路时判断怪物的起点

## 2.5 block.h、block.cpp:

包含 `block` 类，其父类为 `QLabel`：

其构造函数用于存放各类数据

## 2.6 bullet.h、bullet.cpp:

包含 `block` 类，其父类为 `QLabel`：

包含两种子弹 `arrow_bullet`：其构造函数储存数据，从起点出发，由 `game` 每隔一段时间移动子弹；`laser_bullet`：其构造函数储存数据，从起点出发，画出到终点的一道激光：

## 2.7 attetions.h、attetions.cpp:

包含 `attetions` 类，其父类为 `QLabel`：

其与其槽函数用于显示持续一定时间的效果以及横条：

```
void onTimeupDestroy();  
void onTimeupDestroy_2();  
void onTimeupDestroy_3();
```

## 三、小组成员与分工

此项目由林坚铮，刘松瑞，李之怡参与完成

### 小组协作方式：

以读为主，快照隔离，写事务重叠则进行整合

### 小组分工：

刘松瑞负责塔的基类、怪物的基类、提示类、子弹和激光类以及各种特效；关卡数据策划；各种贴图设计

林坚铮拓展了各种塔的派生类、怪物的派生类、商店类的拓展、各种特效、游戏时的数据显示、下一波窗口提示、关卡结束结算和各种窗口跳转以及各种类的交互

李之怡负责商店类的创建与初始寻路代码，飞行怪的寻路算法，背景素材，初代塔的素材，初始结算页面

林坚铮共进行更新 26 次

刘松瑞共进行更新 23 次

李之怡共进行更新 4 次

## 四、小组成员与分工

通过本次程序设计实习 Qt 大作业 `ArkMidnight` 塔防游戏的编写，我们提升了对 Qt 语言的了解，并对类与对象的程序设计思想有了更深入的感受。同时，作为一款面向玩家的游戏，在人机交互的实现过程中，我们也不断打磨 `ui` 等细节，思考如何提升玩家的体验感，力求简洁、易懂。

在本次程序设计的过程中，我们仍有许多尚待改进的地方：一，类的封装仍有待进一步提升，在信息传递的过程中可以将位置等信息以统一的“位置”类进行存储；二，游戏的 `ui` 设计风格还可以进一步统一，还可以增加更多炫酷的特效，并且可以考虑改进为全屏模式，增加更多灵活的地图样式，给玩家更好的体验。三，还有很多功能尚待实现，比如存档功能，塔的升级，右键显示信息的菜单等等。