

# 1：代码部分

## 1.1: 整体架构

代码分为四部分

**全局常量，资源和函数--窗体及其衔接---游戏中控系统---游戏中的各种单位类和对象**

**1.1.1 全局资源：**我定义了一个全局常量类 Constant，来存储圆周率等数学常量，以及游戏边界等游戏全局常量。我还定义了一个全局资源类 Resources 来存储和预先加载所有图片，音乐资源。我还定义了一些能在全局应用的函数—求距离，生成随机数和随机坐标，它们被定义在 GlobalFunction.cpp 里

**1.1.2：窗体及其衔接：**一个窗体中控类 GameMainClass 组合了所有窗体类，并连接了对应了信号与槽，能够实现窗体和 bgm 的切换

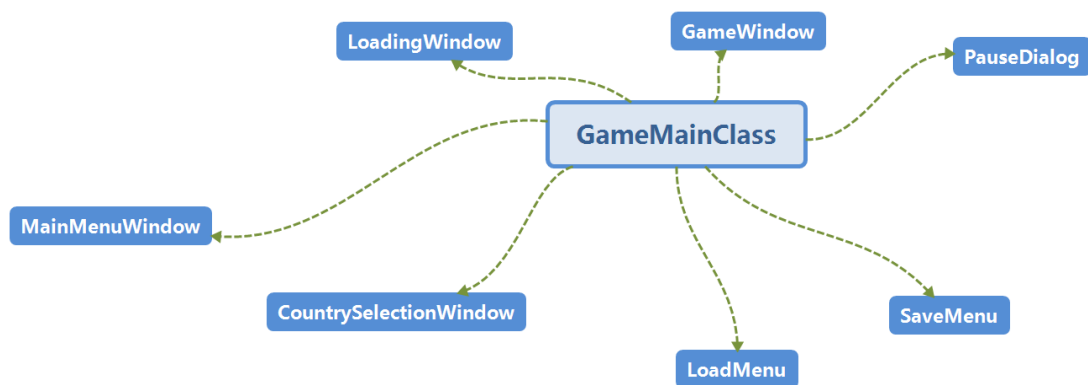


图 1：窗体的组合关系

**1.1.3：游戏中控：**游戏窗体 GameWindow 有一个成员对象是游戏中控 GameCentralControl 的实例。

这个中控有四个功能：1：负责存储所有游戏里的单位 2：进行主事件循环，处理游戏里单位间的交互（比如捕食，攻击） 3：它的槽函数能实现游戏单位的创建 4：和游戏窗体交互，修改窗体的显示还有控件信息

**1.1.4：单位类和对象：**

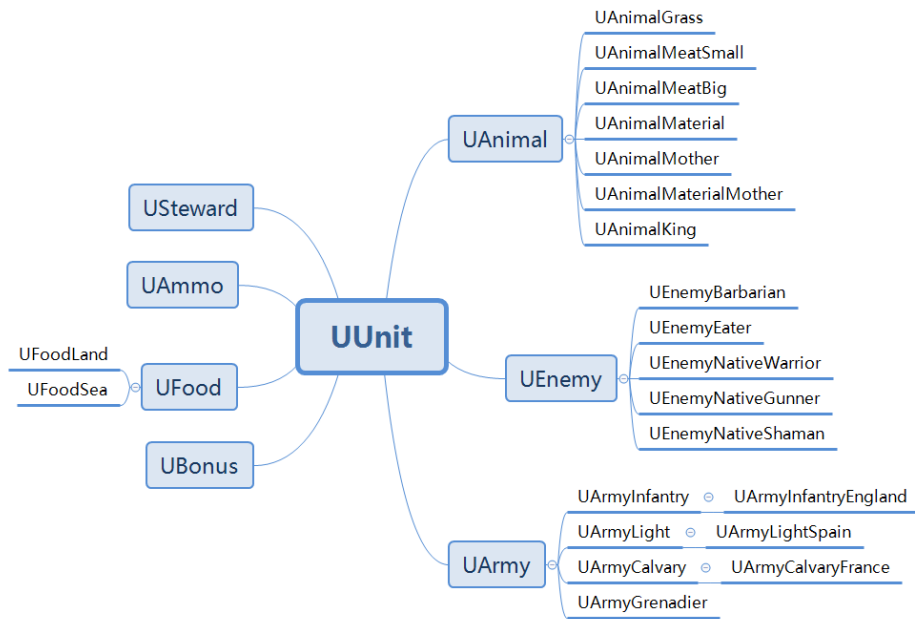


图 2：单位类的继承关系

单位类和对象主要负责存储单个对象的数据，进行自身的更新（比如判断升级，判断饥饿，判断死亡），以及提供接口变量和函数给中控进行操控。

单位类的构件方法是策略模式：

首先是一个基类 UUnit，定义了最基本的变量和函数：生命，位置，移动，是否有效之后是各种更具体的基类，比如 UAnimal，UEnemy，他们以虚函数覆盖的方式定义了其子类能用到的**绝大多数函数**。之后是各种具体子类，比如 UAnimalGrass，它们一方面以 const 的形式存储了其自身的基本属性（这样只需要修改 const 就可以修改类基本属性，方便开发者进行修改），并且在构造函数中真正对自己的基本属性进行赋值；另一方面，他们覆盖了一些基类函数，并且添加了自己的属性和函数。

这种方法的优点是**便于扩展**。比如要添加一个新兵种，你只需要定义一个继承 UArmy 的新兵种类型，并且添加对应的常量以及基础变量赋值，再覆盖需要覆盖的函数就可以了。

## 1.2 游戏流程

### 1.2.1 窗体切换

窗体切换通过 GameMainClass 里一系列信号-槽的连接实现，流程如下

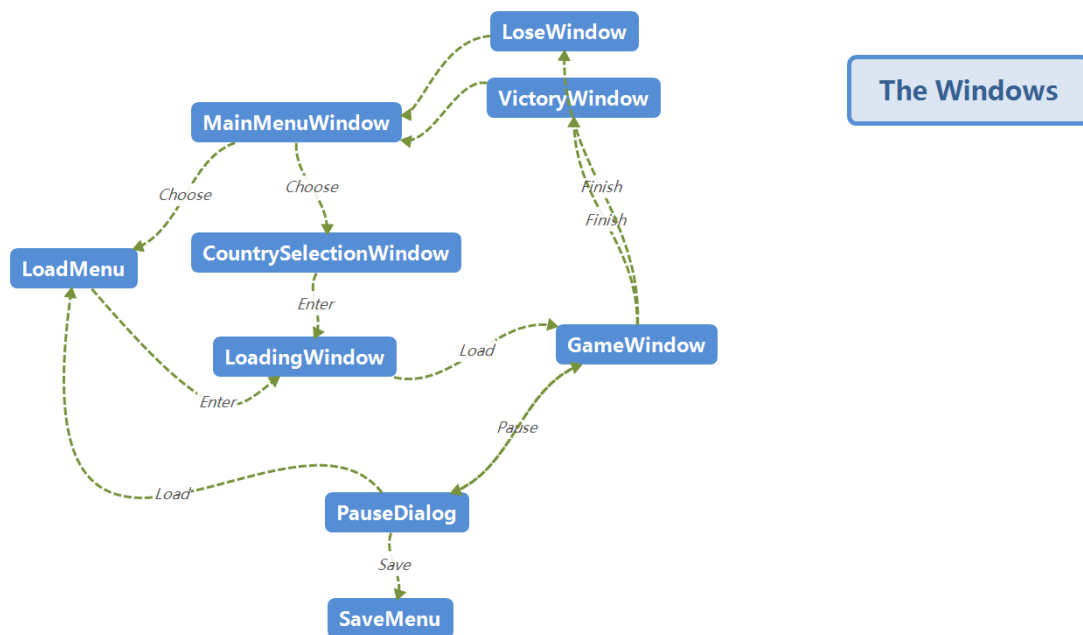


图 3：窗体的切换流程

### 1.2.2：游戏界面

游戏的基本结构是中控类 GameCentralControl 的实例对象负责存储数据，更新游戏全局信息和游戏单位信息。游戏窗体 GameWindow 负责更新控件和绘图，以及开启和关闭游戏主循环。各个单位类的对象负责存储单个对象的属性，并且提供接口供主事件循环更新。游戏流程如下

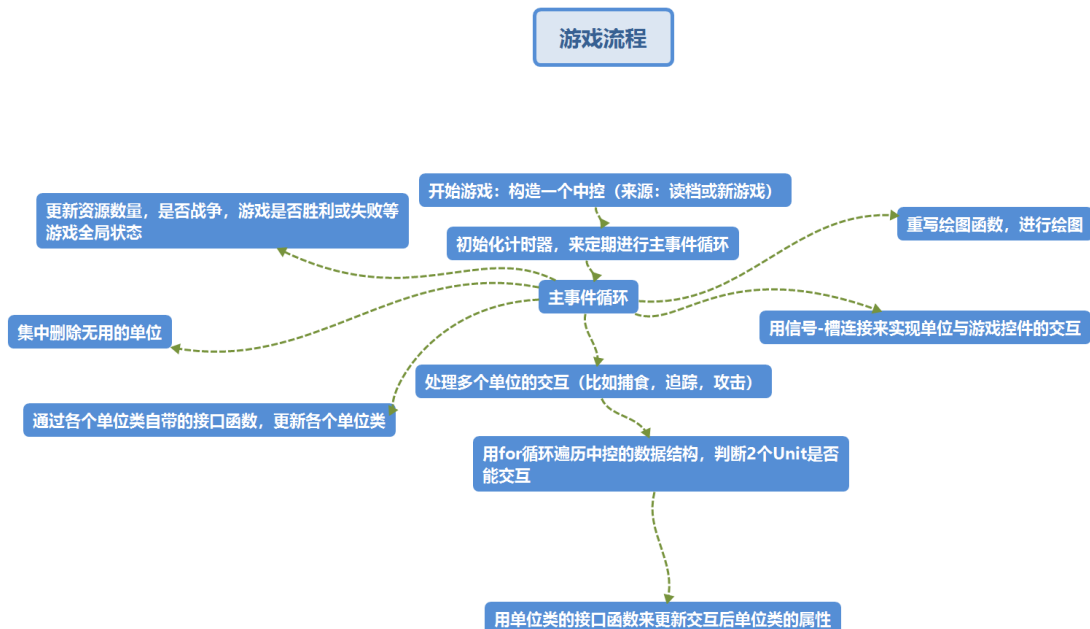


图 4：游戏流程

### 1.3：代码风格

**1.3.1：命名：**我的命名规范是所有全局的东西还有类和对象都用帕斯卡命名法，也就是全部单词首字母大写。单位类加上前缀 U。类的成员函数和成员变量前加 m\_前缀，而成员常量则加 m\_c 前缀。全局变量和函数前加 g\_前缀。

而我的局部变量则都是小写字母，单词之间用\_连接

**1.3.2：注释：**我的每个文件上方都有注释。部分类和所有函数的声明（.h）都有注释。

## 2：具体功能及测试

**我在用户文档里对游戏背景和游戏机制做了全面的描述，如果对游戏机制不清楚可以看用户文档。**

**2.1：能正常编译运行**

**2.2：有正确的各种界面和鼠标键盘事件交互**

测试方法：点击开始游戏并选择国家后会进行加载，之后进入游戏界面。游戏中按 esc 会进入暂停界面。鼠标左右键点击游戏区域会产生食物，点击按钮有对应的操作。

**2.3：饲料：**实现了两种饲料：陆地和海洋饲料，可以升级，也可以增加饲料数量。左键产生陆地饲料，不能移动；右键产生海洋饲料，会缓慢下降。

**2.4：鱼类（动物）**

我实现了 7 种动物，其中包括大作业要求的四种动物，并且实现了他们的升级，捕食，随机运动等功能

动物名称和对应	图片	特性	价格与生产
小型草食动物（古比鱼）		有四级，通过吃食物积累经验。经验达到一定程度会升级。三级升级四级有很大概率失败。	价格 100 金钱，2,3,4 级动物分别生产铜币，银币，金币。
小型肉食动物（小型食肉鱼）		吃 1 级小型草食动物。	生产金币，价格 1000 金钱，需要进入城镇（1 级城墙）才可以生产。
大型肉食动物（大型食肉鱼）		吃小型肉食动物。	生产钻石，价格 10000 金钱，需要进入大型城镇（2 级城墙）才可以生产。



繁殖性草食动物 (古比妈咪)		有 2 级，通过吃食物积累经验。	价格 200 金钱，需要城镇（1 级城墙）生产。会定期在自己所在位置生产小型草食动物，2 级生产速度是 1 级二倍。
小型金属动物		有三级，通过吃食物积累经验，经验达到一定程度会升级。	价格 200 金钱，需要城镇（1 级城墙）生产，1,2,3 级生产 1,2,3 级建材
繁殖性金属动物		吃食物	会生产小型金属动物，需要 500 金钱，需要城镇（1 级城墙）生产
巨型蟾蜍		不会饥饿，生命很多	会生产金币和三级建材。价格 10000 金钱，需要城市（3 级城墙生产）。

表 1：我实现的动物列表

测试方法：达到动物的生产要求，在左下方生产即可

## 2.5：敌人和入侵

游戏会刷出敌人，刷出敌人前几秒窗体上方会有提示。我实现了五种敌人，包括游戏要求的两种

名称	图片	特性
野兽（蓝色怪物）		会攻击离自己最近的动物，攻击力高。但是可以用鼠标点击它放置炸弹，只是会误伤周围单位。死后掉落金币。 任何时候都有
贪吃蛇（贪吃兽）		会走向离自己最近的食物或动物并且捕食。但是吃东西会掉血。死后掉落金币。




		城墙到达 1 级才会有
土著战士		擅长近战攻击的敌人，只攻击军队 城墙到达 2 级才会有
土著火枪手		擅长中距离射击的敌人，只攻击军队 城墙到达 2 级才会有
土著萨满		随机将视野内一个动物转化成野兽 城墙到达 3 级才会有

表 2：我实现的怪物列表

测试方法：将游戏进行到对应的阶段

## 2.6：宠物（军队）：

我的游戏以“殖民地规模”来替代“关卡”，在升级城墙后，可用的动物，宠物，建筑，以及刷出来的敌人都会有很大不同。升级城墙后一段时间有提示。胜利条件是建成三级

城墙后，购买胜利标记。



我实现了六种宠物，其中包括要求的 1 种宠物

名称	图片	特性	价格
事务官（蜗牛）		不能攻击，战争状态下会躲进帐篷。 和平状态下会搜集视野内的掉落资源。	1000 金钱，需要城镇（1 级城墙）生产
战列步兵		肉搏能力较强，远程输出较差，速度较慢。 战斗方式是在中距离先快速开三枪，然后进入近战模式冲锋。	200 金钱 100 材料，需要兵营生产。
轻步兵		不能肉搏，生命较少，射速慢，但是攻击距离长，而且远程攻击力高。 战斗方式是一直随意移动，然后开枪	消耗 200 金钱 100 材料，需要兵营生产。

		攻击离自己最近的敌人。	
骑兵		肉搏能力强，移动速度快，但是只能肉搏。 战斗方式是冲向最近的敌人肉搏。	消耗 500 金币 200 材料，需要马厩生产
掷弹兵		肉搏能力强，手雷集体杀伤，但是移动慢。 战斗方式是先扔手雷 1 次攻击敌人，然后冲锋。	消耗 500 金钱 200 材料，需要军火库生产。
特色单位		三个国家的特色兵种，属性和一般兵种不同，但是战斗方式一样。	消耗 1000 金钱 300 材料，需要城市（3 级城墙生产）

表 3：我实现的的宠物（军队）列表

## 2.5：建筑升级和国家系统

游戏界面右边有一栏建筑列表，你可以花相应的资源购买建筑。不同城墙等级能购买的建筑不同。建筑会提供各种增益，或者解锁兵种。详情见用户文档。

每局游戏开始前都有选择国家选项，不同的国家特色兵种不同，特色能力也不同，详见用户文档。

## 2.6：暂停界面和存档

在游戏里点击 esc 可以暂停

开始游戏和暂停界面都有读档按钮，可以读取.sav 存档

可以在暂停界面里存档为.sav 文件

但是问题是只能正常读取符合格式的存档，也就是在游戏里保存的存档。如果擅自修改存档游戏会崩溃。

## 2.7：音效

游戏主界面和游戏有两首不同的 bgm

在近战，开枪，扔手榴弹都有对应的音效

测试以上功能的便捷方法：

```
//enemy numbers per attack by the level of wall
const int m_cEnemyBarbarianNumberByLevel[5]{1,2,3,4,0};
const int m_cEnemyEaterNumberByLevel[5]{0,1,2,3,0};
const int m_cEnemyNativeWarriorNumberByLevel[5]{0,0,1,2,0};
const int m_cEnemyNativeGunnerNumberByLevel[5]{0,0,1,2,0};
const int m_cEnemyNativeShamanNumberByLevel[5]{0,0,0,1,0};

//constants
    //peace and war control
const int m_cWarBeginThreshold{300};           //The min point to start war
const int m_cWarBeginPossibility{30};          //The possibility of war
const int m_cWarWarningTime{50};               //The min time of war warning
const int m_cWarReadyTime{300};               //The time between ready and war

    //start constants
const int m_cFoodStart{500};
const int m_cMaterialStart{500};
const int m_cMoneyStart{1000};
const int m_cBombDamageStart{50};
```

游戏的常量都定义在 GameCentralControl.h 里，以上的常量分别是不同城市等级下敌人数量；战争爆发的时间和概率；起始的资源数目。可以通过修改这些东西来快速达到测试要求，测试功能。具体的测试方法如果不明确可以查看用户文档。

### 3：参考文献

- 【1】 <https://wiki.totalwar.com/>  
[http://civilization.wikia.com/wiki/Civilization\\_Games\\_Wiki](http://civilization.wikia.com/wiki/Civilization_Games_Wiki) 游戏的兵种设计和贴图参考了全面战争和文明维基
- 【2】 主事件循环的方法参考了大作业范例中用计时器刷新的方法
- 【3】 <https://blog.csdn.net/dijkstar/article/details/48848915> 参考了资源文件预先加载方法
- 【4】 [https://blog.csdn.net/qq\\_1223113455/article/details/51462980](https://blog.csdn.net/qq_1223113455/article/details/51462980) 参考了键盘鼠标事件处理方法
- 【5】 <https://blog.csdn.net/duanwuqing/article/details/4856609> 参考了计时器用法
- 【6】 <https://blog.csdn.net/seanyxie/article/details/5811931>



【7】 <http://www.kugin.com/qtdocument/qtextstream.html>

【8】 <http://www.kugin.com/qtdocument/qfile.html>

以上是做存读档的参考文献

【9】 [https://blog.csdn.net/qq\\_39836658/article/details/78467907](https://blog.csdn.net/qq_39836658/article/details/78467907) 音乐播

放参考

【10】感谢计 76 沈诣博同学对我的资源加载，文件读取方面的指导