目录

[一、 游戏背景 2](#_Toc444954641)

[二、 游戏规则描述 3](#_Toc444954642)

[地图 3](#_Toc444954643)

[游戏单位 3](#_Toc444954644)

[单位共有属性 4](#_Toc444954645)

[游戏模式 5](#_Toc444954646)

[进食 5](#_Toc444954647)

[指令 5](#_Toc444954648)

[移动 5](#_Toc444954649)

[使用技能 5](#_Toc444954650)

[购买技能和升级技能 6](#_Toc444954651)

[技能列表 6](#_Toc444954652)

[技能说明&满级特技 7](#_Toc444954653)

[计算方式 7](#_Toc444954654)

[其他 8](#_Toc444954655)

[球形生物与目标生物死亡的判定 8](#_Toc444954656)

[三、 平台使用说明 8](#_Toc444954657)

[此处有待更新 8](#_Toc444954658)

[四、 AI编写指南 11](#_Toc444954659)

[接口说明 11](#_Toc444954660)

[查询指令 11](#_Toc444954661)

[行动指令 11](#_Toc444954662)

[特殊指令 12](#_Toc444954663)

[其他指令 12](#_Toc444954664)

[编写ai.cpp 12](#_Toc444954665)

[简单的例子 12](#_Toc444954666)

1. 游戏背景



*茫茫宇宙，万古洪荒。在原始而**恶劣的环境下，资源稀缺，妖孽横行，万千生命为了自己的生存，用丰富的战斗技能和策略努力争夺星际之间的版图。在这惊心动魄的战斗中，弱小的生命渐渐走向消亡，而遗留下的强大生命，则会继续着这如同诅咒一般的轮回。*

*而今年，2016年，则是他们的第17次轮回。经过17个轮回的洗礼，他们来到了21世纪，从过去的生命体实物，进化成为了依托于计算机的数码虚拟生命。拥有数码虚拟生命的同时，他们希望在这一次的轮回中，能够有人来帮助他们终结这个诅咒。为此，电子系科协举办了第17届队式大赛，来向更多的人们寻求帮助。而你，也许就是我们所寻找的人！*

*在这里，你可以找到值得信赖的队友，挑战一波又一波测试赛，享受3D回放带来的视觉盛宴，最终与其他选手进行数轮正面对抗。在这无垠的太空中，正义还是邪恶、征服还是被征服，一切掌握在你的手中，开启属于你的队式十七吧！*

1. 游戏规则描述

游戏双方队员需编写AI程序在我们的平台上对战。对战开始时双方各控制一个生物球，在游戏地图内，取食和攻击其他玩家及其它物体，消灭其他玩家、吃掉大型中立生物、或时限到时拥有最大体积则获得胜利，但若在游戏过程中生命值小于历史最高生命值的四分之一，或者被其他生物食用，则被判为失败。

## 地图

地图采用三维正方体空间地图。玩家可以以任意角度向任意方向运动，但速度有上限。当然，玩家不能越出地图边界。

## 游戏单位

**球形生物**

这是唯一可以被AI控制的物体，主要属性有生命值、技能点两种。

选手所控制的球形生物的生命值 = ，因此可以通过观察其他球形生物的半径得知其他球形生物的生命值。除此之外的属性（技能点和技能等级）均不能被其他玩家观察到。玩家初始生命值为1000。

此外还有视野、速度、技能等级（0级表示没有此技能）属性。

初始视野为5000。速度值可以通过速度矢量来控制，但上限为100。初始各项技能等级均为0。

**能量源**

地图上随机生成的不会移动的点状物，玩家食用后可增加40点生命值。

**光之隧道**

地图上的某些固定点不定时生成的高级能量源（约100回合生成一个，所有对应固定点均有光之隧道则不生成），玩家食用后可增加1~5点技能点。

**吞噬者**

地图上的固定球形物体，若被玩家触碰到（即吞噬者处于球形物体内，包括相切），即会令玩家生命值减少1/3，同时吞噬者自己消失（玩家球心位置不变）。

**目标生物（Boss）**

大型球形生物，行动迟缓且行为类似于随机游走，食用后即取得游戏胜利，同时可以食用玩家，但不会对玩家发起攻击。

目标生物可以被伤害，且会因此而减少生命值，从而使其更容易被吞食。

目标生物可食用能量源、光之隧道,但遇到吞噬者不会发生任何变化。

### 单位共有属性

|  |  |
| --- | --- |
| 共有属性 | 说明 |
| team\_id | 所属队伍0/1, 无主为-2 |
| id | 单位编号 |
| type | 单位种类 |
| pos | 单位坐标 |
| radius | 单位半径 |
| long\_attack\_casting | 远程攻击蓄力时间（没有则为-1） |
| shield\_time | 护盾剩余时间（没有则为0） |

**球形特有属性**

|  |  |
| --- | --- |
| 属性 | 说明 |
| health | 生命值 |
| max\_health | 历史最大生命值 |
| vision | 当前视野范围 |
| ability | 技能点 |
| speed | 当前移动速度 |
| skill\_level[kSkillTypes] | 技能级别 |
| skill\_cd[kSkillTypes] | 技能冷却时间 |
| dash\_time | 加速剩余持续时间 |

## 游戏模式

### 进食

* 进食可以增加生命值，受到攻击和使用部分技能会减少生命值。
* 进食光之隧道可以增加技能点（具体数值随机），购买技能和升级技能需要消耗技能点。

若某个可被食用的物体的球心位于可进食的生物体内，且生物的半径与物体的半径之比大于1.2，则该物体立即被该生物食用并消失。

生物之间的进食，在进食后进食生物的生命值将等于原来的生命值加上被进食生物的生命值，但技能点不会改变。

每次进食后（包括能量源、光之隧道、吞噬者），进食生物的球心不发生改变。

进食不需要指令操控，生物将自动进行进食操作。

### 指令

每回合前选手可以为己方单位设定该回合要执行的指令。

* 无效的指令将被忽视。
* 每个回合只能发一条指令（包括升级技能/使用技能/移动），如果发送多于一条指令，则只有最后一条有效。

指令包括：

### 移动

玩家可以以任意角度向任意方向运动，但速度有上限。当然，玩家不能越出地图边界。

运动控制不考虑惯性和阻力，AI一给出速度矢量，玩家就立即以该速度矢量运动。

每单位时间运动一次，每次运动根据速度及速度矢量算出在每个方向上的运动距离。

### 使用技能

使用技能可在瞬间完成，不影响运动。

技能有冷却时间，使用技能后一段时间内不可再次使用同一技能，具体数值与技能种类有关。

### 购买技能和升级技能

选手只要拥有技能点，即可随时随地购买技能和升级技能，且购买和升级可以瞬间完成，不影响运动。

购买新技能的价格 = 技能基础价格 （n为已拥有的技能个数）

升级已有技能的价格 = 技能基础价格 （n为升级前该技能等级）

每种技能的最高级别均为5。

技能一旦购买将一直拥有。

### 技能列表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 技能名称 | 基础价格 | 零级 | 一级 | 二级 | 三级 | 四级 | 五级 | 冷却时间 | 消耗生命值 |
| 远程攻击 | 1 | 伤害：0 | 100 | 200 | 300 | 400 | 500 | 80 | 10 |
| 距离:0 | 3500 | 4000 | 4500 | 5000 | 5500 |
| 近程攻击 | 1 | 伤害：0 | 500 | 700 | 900 | 1100 | 1300 | 80 | 50 |
| 范围：0 | 2000 | 2200 | 2400 | 2600 | 2800 |
| 护盾 | 2 | 防护时间：0 | 30 | 45 | 60 | 75 | 90 | 100 | 0 |
| 加速 | 1 | 速度增量：0 | 20 | 40 | 60 | 80 | 100 | 100 | 40 |
| 持续时间:0 | 40 | 40 | 40 | 40 | 80 |
| 视野（被动） | 2 | 视野范围：5000 | 6000 | 7000 | 8000 | 9000 | 10000 | 0 | 0 |
| 生命提升（非技能，一次性） | 1 | 每次**使用**后立即增加500点生命值 | | | | | | 0 | 0 |

### 技能说明&满级特技

* 远程攻击
  + 发动远程攻击，在固定的蓄力回合数之后，若目标在可攻击范围内则一定能命中指令中设定的目标使其受到伤害，否则此攻击失效。
  + 5级时在命中设定的物体目标后，被命中物体会得到持续时间30的不可移动的负面状态。
* 近程攻击
  + 瞬间使附近的非己方生物受到伤害，伤害较大，范围较小。
  + 5级时在一次技能发动后，可获得持续时间30的满级护盾效果（该护盾对所有玩家不可见）。
* 护盾
  + 一段时间内不受玩家攻击影响（其他玩家可见护盾剩余时间）。
  + 使用时仍然可以被食用。
  + 仍然无法避免接触吞噬者造成的伤害。
  + 4级时技能强化为持续时间内免疫食用。
  + 5级时技能强化为持续时间内无视吞噬者。
* 加速
  + 使用后可在持续时间内获得加速效果。
* 视野（被动技能）
  + 升级后立即生效
* 生命回复（非技能，一次性生效）
  + 第n次使用此技能的价格=基础价格\*2^(n-1)
  + 使用后立即生效
  + 使用次数上限为5次

### 计算方式

* 结算顺序
  + 1.使用技能
  + 2.所有物体的移动
  + 3.物体间的食用
* 距离 
  + 两点在空间中的欧式距离
  + sqrt(dx \* dx + dy \* dy + dz \* dz)

### 其他

#### 初始条件

双方各拥有一个生命值为1000、技能点为0的生物球。

#### 胜利条件

若有一名玩家达成下列条件之一，游戏即告结束，该玩家胜利。

* 其他玩家所控制的所有生物死亡
* 吞食了目标生物
* 时限已到，且自己生命值总和最高

### 球形生物与目标生物死亡的判定

当前生命值与历史最大生命值之比小于1/4或被其它生物所食用即告死亡。

1. 平台使用说明

此处有待更新

下载了安装文件之后，在英文路径下运行Deep Blue.exe，就会看到我们高大上的主界面:（决赛上3D版的主界面会更加酷炫！）



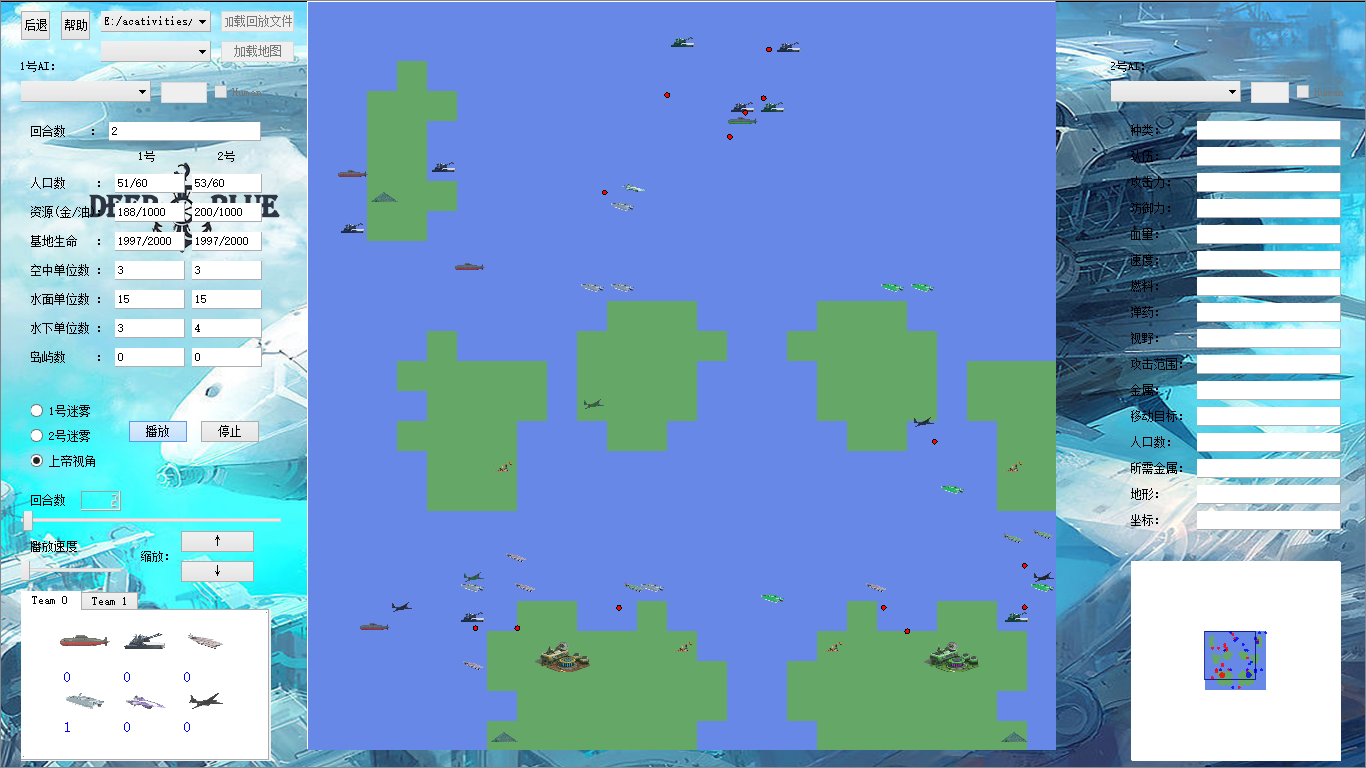
选择“单人游戏”之后，会进入这个界面：



在这里可以选择进行AI对战、人机对战或进行对战的回放。如果想要进行ai对战，请在两侧的ai选择区加载两个ai。如果想要进行人机对战，只需加载一侧的ai并在另一侧的human选择框上打勾。如果想要进行对战的回放，点击左上角的加载\*.battle文件观看回放。

操作方法：

1. ai对战时，可通过下侧的进度条来调整播放速度以及播放回合数，开关战争迷雾。
2. 人机对战时，下侧按钮将会失效。数字键1-6或者鼠标左键点击选择生产，左键选中单位后，a/s/m/f+左键对应攻击/补给/移动/修理，右键点击可以进行攻击/移动。
3. 观看AI对战时，点击播放按钮开始观看比赛回放
4. 点击左上角的按钮返回主菜单，或点击按钮查看帮助信息

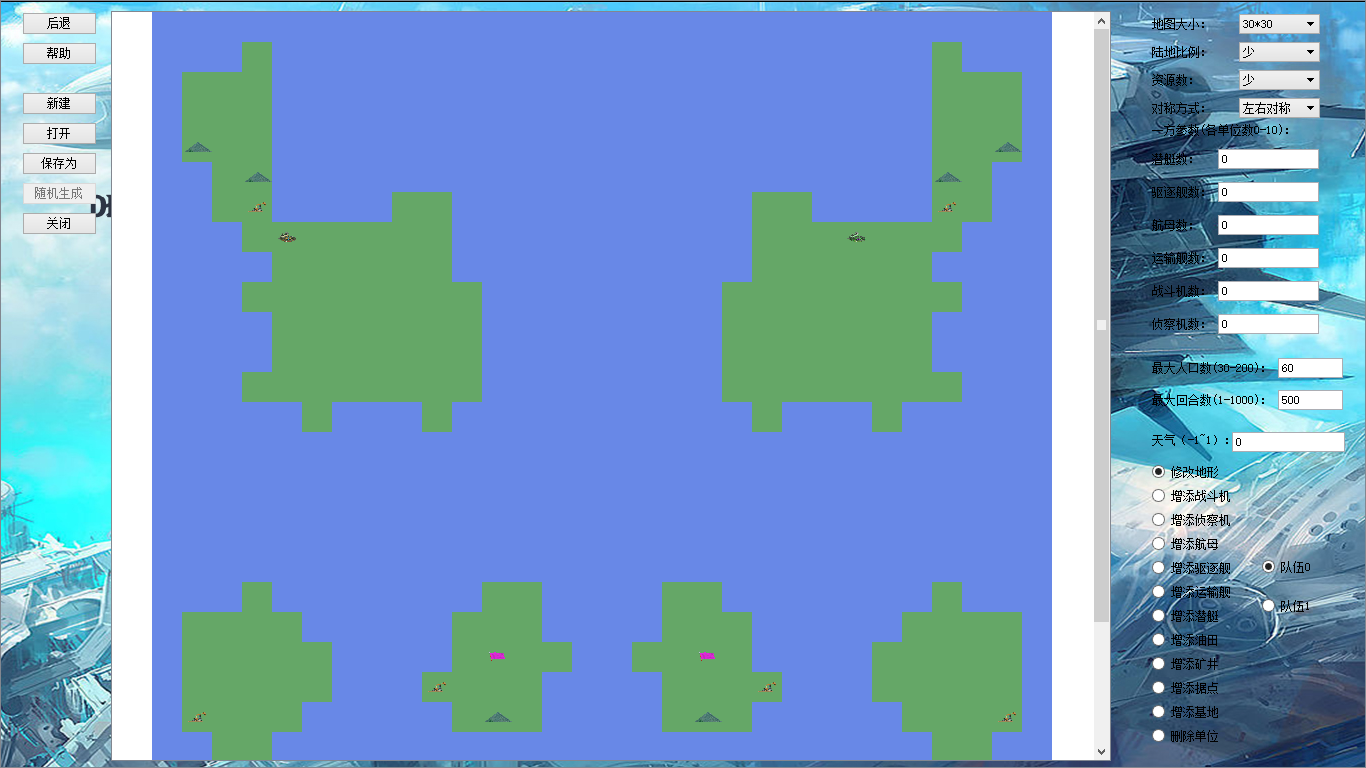


（给你们偷偷看一眼高大上的战争回放画面，决赛的3D版更是简直酷炫到没朋友）

在主菜单中选择“地图编辑”，进入地图编辑界面。

在编辑地图时，首先请选择地图的规模，陆地比例，资源数，对称性，然后选定各单位数以及最大人口数和回合数。可以用滚轮或者按键缩放地图详细查看，也可以通过随机生成按钮来生成新地图。（随机生成的地图可能有一定不合理的地方，如遇到请重新生成）

一定记得保存。



1. AI编写指南

## 接口说明

以下接口均来自于basic.h，如有未尽之处，请仔细参考该文件。

### 查询指令

通过以下三个函数可以获取游戏中的各类信息。

const Map \*GetMap(); // 获取视野内的单位信息

函数返回的结构体如下：

struct Map {

int time; // 当前的回合数，也可用 GetTime 查询

Object objects[kMaxObjectNumber]; // 视野内的所有物体

int objects\_number;

};

const Status \*GetStatus(); // 获取己方单位的状态函数返回的结构体如下：

struct Status {

int team\_id; // 自己的队伍编号

PlayerObject objects[kMaxPlayerObjectNumber]; // 自己所有单位的列表

int objects\_number;

};

int GetTime(); // 查询当前游戏时间

### 行动指令

说好的零基础，自然不会骗大家。基本操作指令学长们已经帮大家写好了，直接调用如下函数即可。说明：user\_id是己方球形生物的单位编号，target\_id是对手球形生物的单位编号。

void Move(int user\_id, Speed speed); // 移动，参数是速度矢量

void LongAttack(int user\_id, int target\_id);

void ShortAttack(int user\_id);

void Shield(int user\_id);

void Dash(int user\_id);

void HealthUp(int user\_id);

void UpgradeSkill(int user\_id, SkillType skill);

## 编写ai.cpp

我们的平台会调用各位到时候编写的void AIMain()，因此这是诸位编程的基础，想想在未来一段时间里把这短短几行变成一个高大上的程序，是不是有点小激动呢~

|  |
| --- |
|  |

#include "teamstyle17.h"

void AIMain() {

// Write Your AI Codes Here :-)

}

那么代码怎么写呢……

### 简单的例子

基本指令的接口前文已有描述，大家直接使用，就可以编制出一个可以运行的AI了。除了基本指令外，还有一些大家可能会遇到的常见操作，在此给各位一点小小的参考。

（这里应该有一些例子）

**好了，学长们也只能帮这么多了，战术与实现还要诸位自己思考~**

**战幕将起，洪荒且乱。为了胜利，前进吧，少年！**