

文件编号: Flyhigher-001

受控状态: ☒ 受控 ☐ 非受控

保密级别: ☐ 公司级 ☐ 部门级 ☒ 项目级 ☐ 普通级

采纳标准: CMMI DEV V1.2



Flyhigher

互动式 AI 飞机大战游戏平台

概要设计

Version 1.1

2019.07

Written by Untitled

Untitled 团队

All Rights Reserved

目录

- 一、前言4
- 二、项目描述4
- 三、UNTITLED 团队介绍4
 - 郭江龙4
 - 张赓鑫5
- 四、功能简介5
 - 1. 功能设计5
 - 2. 核心模块功能描述8
- 五、开发工具与技术11
 - 1. UI 界面开发语言及框架11
 - 2. 主逻辑开发语言及框架12
 - 3. 网络接口设计语言及框架12
 - 4. 预定义 AI 设计语言及框架12
- 六、成员分工12
 - 郭江龙12
 - 张赓鑫13
- 七、应用对象及环境13
 - 1. 应用用户13
 - 2. 运行环境13
- 八、结语13

一、前言

本《Flyhigher——互动式飞机大战游戏平台 概要设计》(以下简称《概要设计》)是针对本次 C++工程训练实践课题的实践项目所做的概要设计,涵盖项目开发初期的需求分析以及功能设计。本《概要设计》作为 Untitled 项目组本次项目的初期规划,对之后的开发过程有着重要的指导性意义。

二、项目描述

“Flyhigher——互动式飞机大战游戏平台”项目是一个互动性游戏平台。游戏玩家可以使用预定义的 AI 模型,或者自定义使用平台接口设计自己的 AI 模型,进行飞机大战游戏,角逐胜负。玩家也可以通过策略性的实时操作与 AI 模型对抗,在飞机大战游戏的过程中逐渐学习人工智能相关的知识,设计更加强大的 AI 模型。

三、Untitled 团队介绍

Untitled 团队成立于 2018 年 11 月,曾参与“Z-HACK 算法设计大赛”、“微软创新杯”、“智能互联大赛”等多项赛事,团队成员均曾获多个省级以上奖项。团队成员如下:

郭江龙

项目负责人,研究机器学习、数据分析、数据挖掘方向。自幼热爱编程,自学 Java、C++、Python 等多门语言,曾获得“2015 first 科技挑战赛 北京选拔赛 一等奖”、“2016 全国青少年奥林匹克信息学竞赛 提高组河北赛区 省级二等奖”、“2018 北邮网安杯 全国中学生网络安全大赛 优秀

奖”、“2018 Z-HACK 算法设计大赛 季军”等多个奖项。

张赓鑫

项目开发主要贡献人，研究机器学习、爬虫、Python Web、前端开发等方向，熟悉 Python、C++、Java、Lua、NodeJS 等多门语言，曾获“2016 全国青少年奥林匹克信息学竞赛 提高组山东赛区 省级二等奖”、“2018 Z-HACK 算法设计大赛 季军”等多个奖项，具有一家公司项目的软件著作权。

四、 功能简介

1. 功能设计

(1) 支持多玩家同时对抗的飞机大战界面显示。

游戏界面由平台根据 AI 选手的策略进行显示。比赛过程中，AI 选手的所有操作都会由平台进行判定并在游戏界面显示相应的动画效果。游戏地图内随机生成可由 AI 选手开采利用的资源，游戏内角色开采资源、攻击、侦测、移动等动作都会根据角色的不同状态产生一系列不同的动画效果，在策略竞争的同时使比赛具备优秀的观赏性。

(2) 编写设计飞机大战的对战策略流程，构建游戏主逻辑框架。

游戏主逻辑采用 ticks 计时，换算实际时间单位是 20ticks/s；总场景设计多个地图（宽度介于 500px~2500px，高度介于 500px~1500px 之间），游戏中弹不分敌友。

游戏支持至少 2 个玩家的对战，具体玩家数量由各对战地图决定。对战地图指定了各玩家可能的出生点，然后随机分配出生点（也可以在开始游戏时由 AI 玩家指定）；每一局游戏拥有 5 分钟的总游戏时间（6000ticks），若时间用尽，则判断所有玩家为负。

母舰：母舰血量 2000，移动速度 1px/10ticks；含有舰载炮，舰载炮发射弹药直径 20px，伤害 60，炮弹移动速度 5px/tick；射击周期 100ticks；母舰每局游戏可以使用一次特殊武器，特殊武器发射速度大小等同于舰载炮，可以被拦截，特殊武器击中目标会造成小范围 AOE 伤害，半径 100px，中心伤害 2000，伤害削弱 20/px，移动速度等同于舰载炮，弹药直径 50px；母舰碰撞体积是半径 30px 的圆。

轰炸机：轰炸机血量 500，移动速度 1px/2ticks；可回航母恢复血量，恢复速度 1/tick，消耗资源 20p/time，发射弹药直径 10px，伤害 80，炮弹移动速度 10px/tick；射击周期 50ticks；轰炸机不可再造；轰炸机碰撞体积是半径 10px 的圆。

战斗机：战斗机血量 200，移动速度 10px/tick；可回航母恢复血量，恢复速度 5/tick，消耗资源 5p/time；战斗机具有机炮，机炮发射弹药半径 2px，伤害 20，炮弹移动速度 20px/tick；射击周期 10ticks；可采集资源，战斗机被歼灭可以再造，消耗资源 100p，建造时间 200ticks；战斗机碰撞体积是半径 5px 的圆。

资源：资源是游戏中所有操作都需要的消耗品，单位为 p，在一定的平衡规则策略下全地图自动刷新，形式为地图中的大小行星。玩家可以通过开采星球的方式收集资源。开采资源需要耗费一定的时间，只有战斗机可以采集资源。

唯一复活机制：只有玩家的战斗机可以重新建造。

失败机制：当一方单位被全歼时，失败。当游戏时间用尽时，失败。

胜利机制：当只有一方单位未被全歼时，该玩家胜利。

视野：每个战斗机只能看到其运动方向左右 30 度区间，深度 200px 的实体，静止状态下，每个战斗机可以看到自身为中心半径

75px 的圆内的实体；每个轰炸机可以看到自身为中心半径 50px 的圆内的实体；航母可以看到自身为中心半径 60px 的圆内的实体。视野内的实体信息会以相对距离和相对方向的方式返回。

(3) 提供网络接口供玩家编写设计飞机的 AI 进行对战。

平台提供 WebSocket 接口供 AI 玩家进行操作，使用 JSON 格式传递数据。接口提供两种调用形式，请求和订阅数据。接口具有不同的频限规则。平台对 Golang、Python、Java 等语言提供 SDK 和接入文档协助玩家整合策略操作。

鉴权： 建立 WebSocket 连接时，玩家需要向平台发送鉴权信息以确定玩家身份。重复发送鉴权请求会触发平台断连。鉴权接口的频限规则为 1 次/1 秒。

请求： 玩家可以根据自身策略的需求通过建立的 WebSocket 连接向平台发送请求数据，平台接受请求后，会通过建立的连接发送相应的数据。请求接口必须在鉴权成功后才能成功调用，否则返回报错信息。请求接口的频限规则为 200 次/10 秒。

订阅： 玩家可通过向平台发送订阅请求来订阅不同的状态数据，订阅成功后，平台通过建立的 WebSocket 向玩家推送一次全量数据，并在玩家订阅的数据改变时，实时向玩家推送增量数据。平台只执行收到的第一个重复订阅请求。订阅接口必须在鉴权成功后才能成功调用，否则返回报错信息。订阅接口的频限规则为 200 次/10 秒。

(4) 提供使用监督式学习预训练完毕的 AI 作为预定义 AI。

项目后期，利用通过 C++ 语言设计完好的 WebSocket 客户端模型对通过 tensorflow、keras 预定义策略模型训练好的 AI 模型进行封装，构建预定义 AI。

2. 核心模块功能描述

核心模块的总览图如下所示：

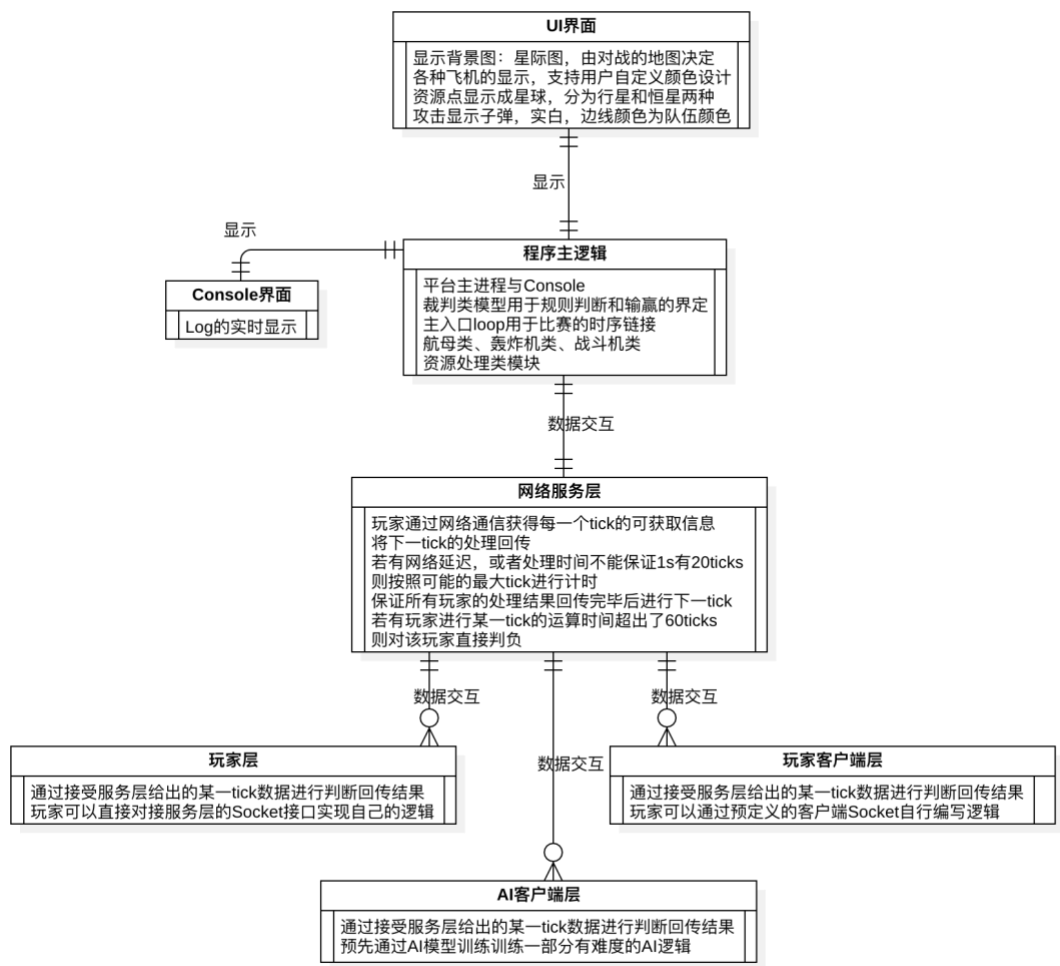


图 4-1 核心模块总览图

(1) UI 界面模块

子模块	概述
界面刷新模块	根据游戏的 fps 计算单位 tick 需要刷新的 fps 帧数，至少保证 1fps/tick，在每帧刷新界面

坐标计算模块	将当前 tick 游戏中各实体在图形界面中的位置
数据录制模块	记录每一个 tick 的显示信息

表 4-2 UI 界面模块子模块表

UI 界面模块将主策略模块在每个 tick 回传的实体信息进行坐标计算，之后实时刷新 在可视化界面中。UI 界面仅作为策略结果的展示，仅将每 tick 的实体坐标信息记录到文件，并可在游戏结束后，根据此文件回放比赛。

（2） 程序主逻辑模块

子模块	概述
主进程模块	主进程模块负责程序本身的入口和基础功能（选图，添加 AI 等），同时进行 log 的记录，并实时显示到 Console 界面。
裁判类模块	裁判类根据一局游戏的基本参数，构造游戏规则，并负责实时监管游戏过程中的数据流动，阻断非法操作。
时序模块	时序模块作为一局游戏的时间控制，负责进行 tick 的 loop 循环，调用每个 tick 的事件传递给其他模块。
飞机类群模块	飞机类群模块定义了游戏过程中出现的各个飞机模型，同时定义了他们具有的操作。飞机类群模块受裁判类模块的监管。
资源处理模块	资源处理模块负责一次游戏过程中的所有资源信息的储存。包括游戏玩家的资源储量、资源随机生成的位置、大小等。
数据录制模块	数据录制模块会保存策略操作数据，并允许复盘操作。

表 4-3 程序主逻辑模块子模块表

程序主逻辑模块接受网络服务层传递来的信息，实时演算并判断合法性。程序主逻辑模块负责进行游戏的主过程。主逻辑模块会记录每个 tick 的输入数据和判断结果，游戏结束后可据此复盘。数据录制模块会记录被裁判模块判定为有效的 AI 操作数据，提供比赛原生数

据的回放功能，以供查错校验。

(3) Console 界面模块

Console 界面模块提供最简单的图形界面实时显示 log，并通过最简单的输入框进行简单的程序调试。

(4) 网络服务层模块

子模块	概述
数据模块	维护 WebSocket 连接，与玩家策略之间进行 通讯，接收、返回玩家策略的动作信息和执行结果。
连接模块	管理玩家的接入、断开。
数据录制模块	录制网络服务层接收的玩家操作原始数据，并允许复盘操作。

表 4-4 网络服务层模块子模块表

网络服务层为参与游戏的 AI 开放操作接口，并对选手发起的操作请求作出频率限制。

(5) 玩家层模块

玩家层模块并不需要本项目实际实现，玩家层模块是指直接使用 WebSocket 进行编程设计策略的玩家，这个模块可以直接对接服务层的 WebSocket 接口。

(6) 玩家客户端层模块

与玩家层模块不同的是，玩家客户端模块面向对 WebSocket 网络通信技术不太了解的玩家，提供一层对接 WebSocket 服务层接口的封装。玩家客户端层模块可以有很多种类，可以是不同的编程语言，项目后期会逐渐丰富。

(7) AI 客户端层模块

AI 客户端层模块是本项目预设计的策略，使用机器学习技术进行训练获得得类玩家策略模块。AI 客户端层模块等价于玩家客户端层模块，只是策略是预先经过机器学习训练，而不是由玩家决定的。项目中后期会提供一些训练好的固定策略的 AI。

五、 开发工具与技术

1. UI 界面开发语言及框架

UI 界面使用 C++ 语言的 Qt 框架进行开发，Qt 是一个 1991 年由 Qt Company 开发的跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。Qt 是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器 (Meta Object Compiler, moc)）以及一些宏，Qt 很容易扩展，并且允许真正地组件编程。

2. 主逻辑开发语言及框架

主逻辑采用 C++语言进行开发，框架自行设计，具体框架设计 UML 用例参见《Flyhigher——互动式飞机大战游戏平台 详细设计》（以下简称《详细设计》）。

3. 网络接口设计语言及框架

网络部分采用 C++语言实现的 WebSocket 接口与玩家策略进行通讯。WebSocket 是一种在单个 TCP 连接上进行全双工通讯的协议，使服务端和客户端之间的数据交换变得简单，允许服务端向客户端主动推送数据。在 WebSocket 中，服务端和客户端只需要一次握手两者就可以维持持久性的连接，并进行双向数据传输，符合与策略进行实时通讯的设计需求。

4. 预定义 AI 设计语言及框架

预定义 AI 设计使用基于 C++语言的自行设计的客户端框架同服务端 WebSocket 进行交互，策略使用 python 语言的 tensorflow 和 keras 库设计的模型训练，经过一段时间的监督式学习最后完全设计作为预定义 AI 策略。策略模型及训练过程参见《详细设计》。

六、 成员分工

郭江龙

主策略层 60%，UI 界面层 80%，网络服务层 20%，玩家客户端层

20%，AI 客户端层 70%。

张赓鑫

主策略层 40%，UI 界面层 20%，网络服务层 80%，玩家客户端层 80%，AI 客户端层 30%。

七、 应用对象及环境

1. 应用用户

面向对编程开发有一定兴趣玩家，包括想要学习一些 AI 开发知识的玩家。同时也支持想要挑战 AI 进行对战的玩家。项目后期会进行开源，不同开发者可以在已有框架的基础上进行升级改造，丰富游戏玩法。

2. 运行环境

项目采用 C++语言作为平台主逻辑，操作接口对外以网络接口形式提供，可对接多种支持网络编程的语言。UI 界面采用 QT 框架，目前支持 Windows 和 Mac OS 的界面显示。

八、 结语

本《概要设计》涵盖了本次项目计划的主要逻辑业务的定义和规范，对之后的开发过程具有指导性意义。Untitled 项目组会按照《概要设计》中的业务逻

辑进行开发，以最大效率完成最完美的项目设计。