

WrightEagleBASE代码导读之二

windywinter

学习笔记

1 comment

2010.9.11

WrightEagleBASE逻辑结构

WrightEagleBASE的代码可以分为几大部分——工具类、程序结构类、数据结构类、模型类、信息类、决策类。

工具类

Utilities、Types：功能函数和公共数据结构，如计时工具和安全数组等，以及与球队相关的基本的数据类型、宏定义。

Geometry：几何工具，点/向量（Vector）、射线（Ray）、直线（Line）、矩形（Rectangular）、带有最大最小值的 $y = a + \frac{b}{x+c}$ 型曲线（RecipracalCurve）、圆（Circle）。

Net：神经网络工具。

DynamicDebug：调试工具，记录message log，根据message log重现比赛场景以便调试。

Logger：调试工具，记录除message log之外的其他类型log。

Plotter：调试工具，调用GNUPlot画图。

TimeTest：测试函数或过程耗时的工具。

Thread：线程工具。

UDPSocket：UDP接口。

NetworkTest：测试网络延迟的工具。

ParamEngine：从配置文件、命令行、server消息中读取参数的引擎。

程序结构类

Client, Parser, CommandSender：在正常比赛时，

Client::RunNormal()是程序入口，之后创建三个线程——Parser，负责从server接收消息并填充信息类；CommandSender，负责向server发送原子命令；Logger，负责记录log，message log不由Logger记录。Client本身作为决策线程。

Recent Posts

硬币称重问题

猜硬币问题——零和博弈的纳什均衡

Eclipse CDT配置Pretty Printing

WrightEagleBASE代码导读之四

八九一五•三五挽歌

终于解决了Ubuntu 12.04开机显示

“无法应用原保存的显示器配置”的问题了

对Bitcoin全网运算能力上升的估计

Recent Comments

ljlin on GUESTBOOK

sunstorm on 猜硬币问题——零和博弈的纳什均衡

私募内参 on 从centos安装debian

三无屌丝 on Merry Christmas 5

苏理 on Merry Christmas 5

sunstorm on 折腾Nexus 10

WindyWinter on 折腾Nexus 10

Categories

他山之石

付费评论

公告板

学习笔记

心情文字

解题报告

资源共享

Blogroll

Ai.Freedom

Eruca的懒虫窝

E鱼の水

I'm moper

karljppj

Ma Shuo's Blog

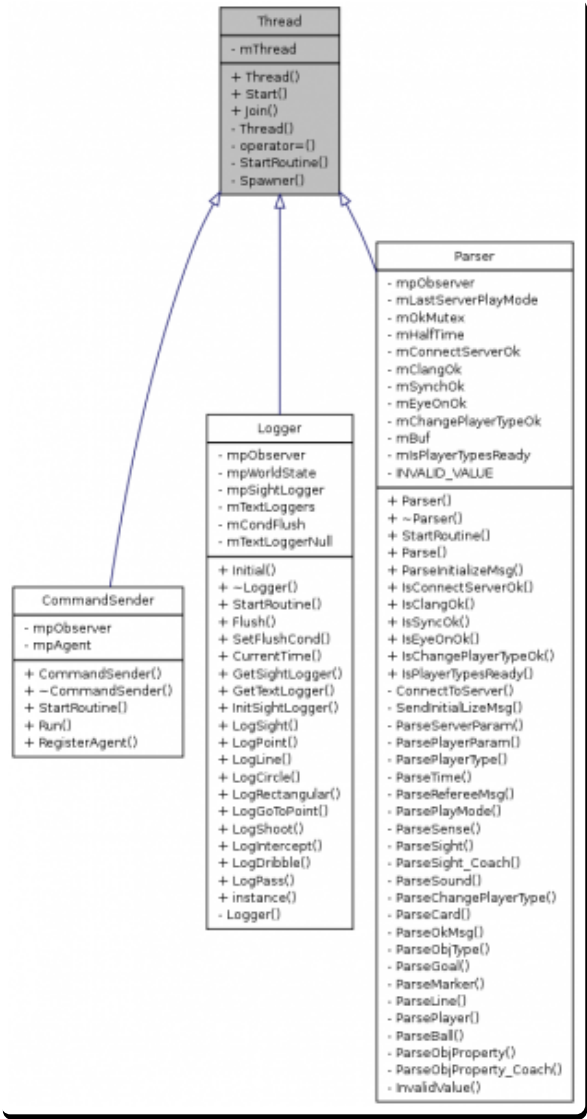
Prince

听听

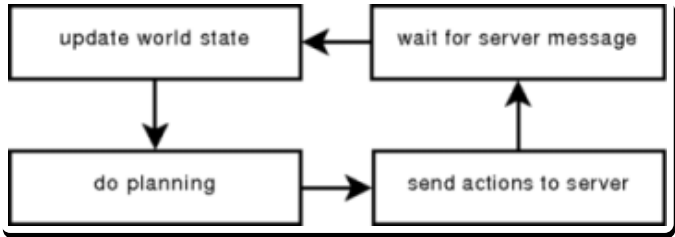
大江东去

天矢の翼

小蛋壳儿



Player, Coach: Client带有一个虚函数Run(), 应该被实现为决策过程, Player和Coach均继承自Client并分别实现了Run()。Player的Run()要复杂一些, 具有明显的层次结构, Coach则非常简单, 没有明显的分层, anyway, 它们都负责实现如下循环:



数据结构类

Agent、AgentID: Agent代表一个智能体——球员或教练, 所有的决策都是围绕Agent进行的, Agent提供了决策使用的信息类和原子命令动作接口, 决策类皆是关于Agent的函数, 即传入Agent实例, 返回决策。使用Agent中的原子命令动作接口会把对应的原子命令放入命令队列, 等待CommandSender调用Agent的SendCommands()发送, 如果该Agent代表的是程序自身的话, 这个特殊的Agent由Client::ConstructAgent()生成。通过“克隆”已有的Agent可以产生一个新的Agent实例, 新的Agent与原来的Agent对环境的认识是一致的 (i.e., 拥有相同的世界状态), 赋予它不同的号码, 即可用于计算队友或对手的决策。Agent有一个mReverse (反算) 标记, 为真, 表示该Agent是代表程序自身的Agent的对手, 为假表示队

新的开始
无人区@寒水远山
沫沫的泡泡
海浪家园
源源小世界
翼若垂天之云
驯鹿的日子
鱼Fish过生活

Login

Username

Password

☒ Remember Me

Login →

Register
Lost Password

Sponsor

MEYU.NET
Free .TK Domain

友。AgentID用于比较两个Agent是否相同。

Unum: 球员号码, 是个整数。Unum必须在某一个Agent的世界状态里才有意义, 正值代表该Agent的队友, 负值代表该Agent的对手。0值一般无意义, 在某些情况下代表教练。

BallState、PlayerState: 用于存储球和球员的状态的数据结构。

PlayerState::mPlayerType是该球员的异构号码; PlayerState::mIsGoalie标记是否是守门员; PlayerState::mIsSentsed标记该球员是否是程序自身, 如果一个球员是程序自身, 它的很多信息可由server的sense信息得到, 与其他球员信息的获得途径不同, mFocusOn与此相关。

Time: 周期。mT为周期数, mS为在某些定位球模式下经过的周期数。

ActionEffector: ActionEffector是Agent的原子命令动作接口, 即实际的命令队列维护、发送、检查皆由ActionEffector完成。ActionEffector同时也存储server发给Client的命令计数。ActionEffector的静态成员函数和ActionEffector.h里的工具函数属于模型类, 提供一些模型计算功能。

模型类

模型类主要进行一些数值计算。

Dasher: 计算跑动相关的数据和动作, 如跑到某一点需要的周期数、最佳原子动作序列。

Kicker: 计算踢球相关的数据和动作, 如将球按指定方向踢出能得到的最大速度、按指定速度踢向某一方向的最佳原子动作序列。

Tackler: 计算铲球相关的数据和动作, Tackler被设计为目标驱动式的, 即Tackler集中于计算将球铲到某一方向所需的原子动作、可得的速度等。

InterceptModel: 求解理想截球模型。

Evaluation: 基于位置的评价体系, 即sensitivity。

信息类

server发来的原始信息经处理和不同程度的抽象后得到一系列可供决策使用的数据。信息是决策的基础。不同的信息更新和组织策略会导致不同的决策模式。

ServerParam、PlayerParam: 管理参数, 分别读取server.conf和player.conf初始化, 再由server发送来的同类参数更新。PlayerParam里有一套独立的HeteroParam, 负责处理异构信息。

Parser: 负责解析server发送过来的“字符串”, 提取出Client的感知(Perception), 转存到Observer里面; 提取各种参数, 转交给ServerParam和PlayerParam处理。同时, Parser还负责跟server取得连接, 并发送初始化信息。

Observer: Observer是Parser线程和主线程的共享数据, 它从Parser获得的感知主要包括: 视觉、听觉和自身感知等。Observer的信息是相对信息, 即坐标等皆在以球员为原点、身体方向为X轴正方向的坐标系(球员坐标系)中, 是Client代表的球员的视角。Observer一般不跟高层决策打交道, 要供决策使用的信息还需进一步处理。

WorldState、HistoryState: 将Observer中的相对信息换算成全局信息(坐标转换到以场地中心为原点, 进攻方向为X轴的坐标系——场地坐标系中), 并利用历史信息、短期预测等多种途径进行计算, 更新得到WorldState, 原来的WorldState变成历史。WorldState包括了球、球员等场

上所有对象的基本状态信息（BallState & PlayerState）。WorldState中的信息不采用任何一个球员的视角，而采用某一个队伍的视角。一份WorldState无需任何改动就可以供所有的“队友”决策使用，镜像对称后得到的新WorldState可以供所有“对手”决策使用。HistoryState管理历史WorldState。

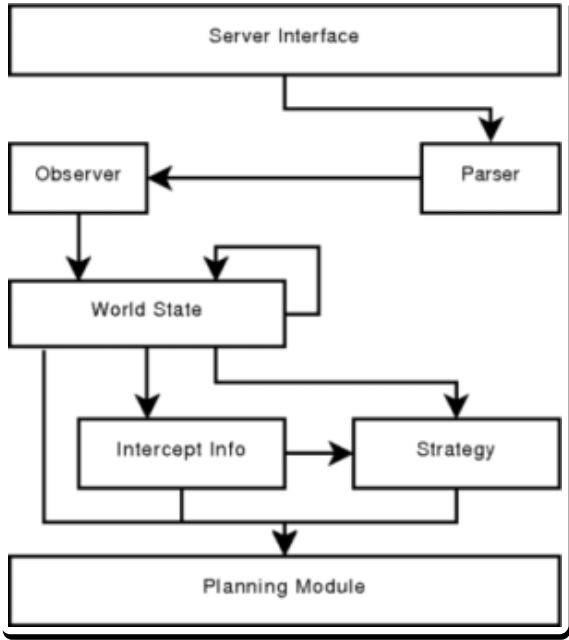
WorldModel：WorldModel提供双方视角的WorldState和HistoryState。

InfoState：对WorldState里的信息进一步计算和抽象得到InfoState，包括：

- PositionInfo：各种相对位置、角度信息；
- InterceptInfo：截球信息，由修正求解理想截球模型的结果得到。

Strategy、Analyser：比InfoState更抽象的上层决策数据，Strategy用于进攻，Analyser用于防守。两者均采用某个Agent的视角，通过WorldState、InfoState和自身历史进行更新。历史决策信息、决策要使用的一些历史相关信息也存储在两者里面，并以不同的形式进行更新。

WrightEagleBASE中的信息流动过程（PositionInfo与InterceptInfo地位相同，Analyser与Strategy地位相同）：



InfoState、Strategy、Analyser等类的继承关系：





Formation：提供阵型信息，算是一个独立的系统，既有底层的从配置文件读入的某阵型的布置方案，也有高层的需要Strategy和Analyser配合的计算当前采用何种阵型的更新过程。大部分信息采用队伍视角，Formation::instance里的内容则与代表程序自身的Agent绑定。

决策类

Coach的决策系统较为简单，这里以Player的决策系统为准。决策系统又分为三层——行为决策、模式决策、高层决策。

行为决策：即BehaviorBlock、BehaviorDribble等以具体行为命名的类。决策的规划跟执行是分开进行的，规划由各BehaviorPlanner进行，执行由对应的BehaviorExecuter进行，这样可以每周期同时进行多个行为的规划但只选择具有最好收益的行为去执行。BehaviorPlanner和BehaviorExecuter利用ActiveBehavior传递信息——BehaviorPlanner规划的结果表现为一个ActiveBehavior，ActiveBehavior即代表一个“行为”，存储行为的收益值及执行时可能用到的所有数据，其执行过程调用相应的BehaviorExecuter进行，而BehaviorExecuter直接从ActiveBehavior读取信息，不再依赖于信息类。

模式决策：主要作用是决定Agent在MAS中的角色和任务，如在防守模式中分派盯防队员、在定位球模式分派开球者、接应者等。模式决策类有以下几个：

BehaviorPenalty：点球决策框架。

BehaviorSetplay：定位球决策。

BehaviorAttack、BehaviorDefense：一般的进攻和防守。

BehaviorGoalie：守门员决策。

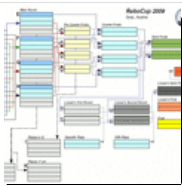
高层决策：指DecisionTree、VisualSystem和CommunicateSystem，这三个决策系统分别最终产生dash/kick/tackle/turn、turn/turn_neck和say类型的原子动作。VisualSystem和CommunicateSystem都是“请求-评价-执行”模式，向行为决策类提供请求接口，经过评价之后，决定哪些请求得到执行。DecisionTree在形式上是MDP的策略迭代算法，但实际上只做一个周期的搜索即选择最优行为执行。代码中DecisionTree的搜索方式是顺序执行模式决策，i.e.，首先执行BehaviorPenalty，当且仅当BehaviorPenalty无结果时执行BehaviorSetplay，当且仅当BehaviorSetplay无结果时执行BehaviorAttack，当且仅当BehaviorAttack无结果时执行BehaviorDefense（若是守门员则执行BehaviorGoalie）。



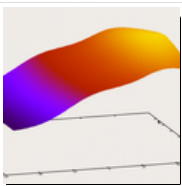
MySQL出错代码详解(@ericos)



提权时候用的2个代码和2个dll和exe文件(@swhack)



Champion, RoboCup 2009



WrightEagleBASE 代码导读



WrightEagle Base Code Release 2



小米之家-小米全系列现货发售！



约惠情人节！小米 M3现货特价2088元！数量有限



有利网：投资理财，好收益。立即领取15元红包！

<http://d.ream.at/introduction-to-wrighteaglebase-2/> Top

Trackback URI


Tags: RoboCup, WrightEagle

I think you are also interested in:

Previous post: 基于PubSubHubbub的Feed实时同步到Twitter服务

Next post: WrightEagleBASE代码导读

One Response



Rongya

2012-05-25 08:46:56

Mark,此帖必火

Reply

Write a Comment

Welcome, [ljlin](#). [换个马甲 »](#)

您可以使用以下HTML标签: <abbr title=""> <acronym title="">
<blockquote cite=""> <code> <i> <strike>

发表评论 (Ctrl+Enter)

Tags Cloud

Harry Potter 蒙特卡洛 popularity contest 范文 手算开方 数论 曲线拟合 布
车方案 整除性 校服 动态图 不等式 天气 ODF 异或 数轴 四面体 牙痛 代数
结构 UFW 链表 八皇后 中位数 神兽 连通子图 华为 树状数组 电子书 白鳍
豚 不定方程 Skype 欧拉路 云计算 coolcode 部分和 New Year 逆序对
Loebner Prize Cartesian Tree Eruca 仙剑 清明 self-number SNS 小i机器
人 WP-Syntax Fedora Core 奥运会 六间房 Star Wars 讲义 zmc 水饺 浮点
误差 游戏 哈密顿回路 Catalan Number OpenID 超级质数 STL

Info

梦.:如此短暂
[Feed](#) [Twitter](#)