

Технология коллективного моделирования агентов

Цель

Создать программную технологию позволяющую отлаживать, испытывать, обучать и исследовать ИИ агентов в сложном виртуальном мире. Технология должна позволять создавать виртуальный мир (в дальнейшем - мир) и поведение автономных объектов в нем без программирования. Технология должна позволять коллективно моделировать поведение множества ИИ агентов в одном мире, при этом программа ИИ агента должна работать на ЭВМ клиента и обмениваться необходимой информацией по сети Интернет с ЭВМ моделирующей мир. На ЭВМ клиента должна быть возможность наблюдать процесс моделирования поведения ИИ агента в мире.

Термины

ИИ агент - программа с элементами искусственного интеллекта моделирующая мозг некоего виртуального существа.

Мир (виртуальный мир) - совокупность объектов и их свойств моделируемая на ЭВМ в которой существует ИИ агент в теле виртуального существа.

Активный объект - объект имеющий поведение или свойства отличные от свойств обеспечиваемых физическим движком мира, например: «хищник» догоняющий по кратчайшему пути жертву, «пища убегающая по случайно меняющейся траектории.

Анализ игровых движков и конструкторов

Проводился анализ игровых движков и конструкторов с целью выяснения их пригодности для технологии коллективного моделирования агентов (ТКМА) в двух или трех мерном мире.

Под миром понимается модель среды в которой «живет» агент. Окончательно были сформулированы следующие требования к программному обеспечению ТКМА :

- 1) наличие поддержки продукта;
- 2) распространенность;
- 3) наличие реально работающего программного обеспечения (ПО);
- 4) наличие документации (хотя бы частично) на русском языке;

- 5) свобода и бесплатность применения (как минимальное требование – бесплатность применения в исследовательских и учебных целях);
- 6) легкость в использовании и освоении («легкость входа»);
- 7) пригодность для научно-промышленных применений;
- 8) отсутствие привязки к конкретному языку программирования для агентов/ботов;
- 9) возможность коллективного моделирования с использованием сети Интернет, при этом программа агента должна исполняться на ЭВМ пользователя и на ЭВМ пользователя должны быть доступны результаты коллективного моделирования агентов различными пользователями;
- 10) возможность отладки пользователем своего агента на локальной ЭВМ в локальном мире, без использования сети;
- 11) легкость построения нового мира пользователем;
- 12) наличие физического движка в программе моделирования мира. Было рассмотрено 144 игровых движка и конструктора, краткая информация о которых расположена на сайте <http://qcup.ru>

После исключения не поддерживаемого в настоящее время ПО, ПО с крайне ограниченным функционалом (например пригодного только для создания аркад), ПО не имеющего режима мультиплеер, необходимого для организации коллективного моделирования, а также ПО не имеющего бесплатных версий осталось около 20 игровых движков и конструкторов.

Следует отметить, что профессиональное ПО для разработки игр имеет весьма высокую стоимость: начиная от нескольких тысяч долларов на одно рабочее место с ограниченной поддержкой и без предоставления исходных кодов ПО и до многих сотен тысяч долларов для корпоративных лицензий с неограниченной поддержкой и предоставлением исходного кода игрового движка. Учитывая необходимость обеспечения п.11) из рассмотрения были исключены большинство игровых движков, т.к. игровой движок обычно не предоставляет возможностей создания мира (редактор сцен часто отсутствует).

В результате в дальнейшем анализ проводился для следующего ПО

с 2-D миром

Constructor (<http://www.scirra.com> <http://scirraconstruct.ru>)

Game Maker (www.yoyogames.com/make)

Eclipse(<http://www.freemorphmaker.com/index.php>)

с 3-D миром

Awakening (<http://www.awingsoft.com/>)

Unity (<http://unity3d.com/>)

Blender (<http://www.blender.org/>)

Попытка рассмотрения игрового движка Unreal Engine не удалась, на моем компьютере

программы пакета выдавали неустранимую ошибку. Кроме того, бесплатная версия данного движка требует 3 Гб свободного места на диске, 2Гб ОЗУ, что мне показалось чрезмерным для поставленных задач.

Среди ПО для разработки 2-D игр было отобрано два конструктора игр, их анализ приведен в таблице:

	Constractor 0.99	Game Maker 8
наличие поддержки продукта	+	+
распространенность	Слабо распространен	Весьма популярен
наличие реально работающего программного обеспечения (ПО)	+	+
наличие документации (хотя бы частично) на русском языке	+	+ (наличие не только форумов, уроков, но и справки на русском языке)
свободность и бесплатность применения (как минимальное требование – бесплатность применения в исследовательских и учебных целях)	+ (свободное ПО с открытым кодом)	+ (но для создания 3-х мерных игр и сетевых игр необходимо приобрести лицензию, стоимостью 25\$ USA)
легкость в использовании и освоении («легкость входа»)	+	+ (производит впечатление несколько менее сложного , чем Constractor)
пригодность для научно-промышленных применений	+/- (для промышленных применений мало пригодно вследствие слабой поддержки 3-D)	+/- (для промышленных применений мало пригодно вследствие слабой поддержки 3-D)
отсутствие привязки к конкретному языку программирования для агентов	Допустимо использование языка Python, за счет чего можно реализовать интерфейс мира с любой программой агента	Встроенный язык (только в профессиональной , коммерческой версии) похожий на C
возможность коллективного моделирования с использованием сети Интернет, при этом программа агента должна исполняться на ЭВМ пользователя и на ЭВМ пользователя должны быть доступны результаты коллективного моделирования агентов различными пользователями	Есть специальная библиотека, обеспечивающая сетевую игру	Есть специальная библиотека, обеспечивающая сетевую игру
возможность отладки пользователем своего агента на локальной ЭВМ в локальном мире, без использования сети	+	
легкость построения нового мира пользователем	+	+ (производит впечатление несколько менее сложного , чем Constractor)
наличие физического движка в программе моделирования мира.	+	+

достоинствам: большой объем русскоязычной документации, большие возможности программы как трехмерного графического редактора, возможность «оживления» объектов без программирования вообще, что ценно для построения простых ботов создающих сложный мир для ИИ агентов. К достоинствам Unity я бы отнес: высокий уровень игрового движка, высокий уровень физического движка, простоту освоения, возможность программирования непосредственно из Unity на JavaScript, C#, Python, к недостаткам: отсутствие полных возможностей трехмерного редактора (что для данного применения не слишком критично), необходимость применения скриптов для задания действий любым объектам, что повышает уровень «входа» для пользователей.

Данный анализ, безусловно, носит субъективный характер, т.к. автор не является экспертом в области разработки игр.

From:

<http://ailab.ru/wiki/> - AI Lab Wiki

Permanent link:

<http://ailab.ru/wiki/doku.php?id=%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8:%D1%82%D0%BA%D0%BC%D0%B0:start>

Last update: **2010/10/31 12:02**