HAW Hamburg

Präsentation AW 2 – SoSe 2009

Gregor Balthasar

JARTS - JAdex plays RealTime Strategy

Ein Jadex-Framework zur einfachen Integration von BDI-Agenten in ein Echtzeitstrategiespiel

Gliederung

- Einführung
  - Testumgebung für verteilte KI „meets“ Videospiel
  - Projekt JARTS
- Vergleichbare Arbeiten
  - FlexBot
  - GameBots
  - AgentContest
  - State of the Art?
- Abgrenzung der Arbeiten von JARTS
  - Szenariobedingte Unterschiede
  - Unterschiedliche Zielsetzungen
  - Gemeinsamkeiten?
- Fazit

Gliederung

- Einführung
  - Testumgebung für verteilte KI „meets“ Videospiel
  - Projekt JARTS
- Vergleichbare Arbeiten
  - FlexBot
  - GameBots
  - AgentContest
  - State of the Art?
- Abgrenzung der Arbeiten von JARTS
  - Szenariobedingte Unterschiede
  - Unterschiedliche Zielsetzungen
  - Gemeinsamkeiten?
- Fazit

Testumgebung für verteilte KI „meets“ Videospiel

- Testumgebungen für verteilte KI (hier am Beispiel Multiagentensysteme) notwendig
- Ansprüche an eine solche Testumgebung:
  - dynamisch
  - konsistent
  - nicht-deterministisch
  - komplex
  - ggf. Mensch-Agent-Interaktion

⇒ Computerspiele bieten so geartete Umgebungen

Zu beachten: Zielsetzungen von Testumgebungen können variieren!

Projekt JARTS

- Jadex-Framework
- Kapselt Kommunikation zwischen Agent(en) und einem Echtzeitstrategiespiel (möglicher Kandidat: Spring-Engine)
  - Kommunikation über ein Netzwerkprotokoll
  - Sensoren über Jadex-Events
  - Effektoren über Jadex-Goals
- Zielsetzung:
  - Einsatz in der Lehre
  - Identifikation von Problemen bei der Entwicklung von MAS
  - Stabilitäts- und Geschwindigkeitstests
  - Bewertung der Glaubwürdigkeit von KI

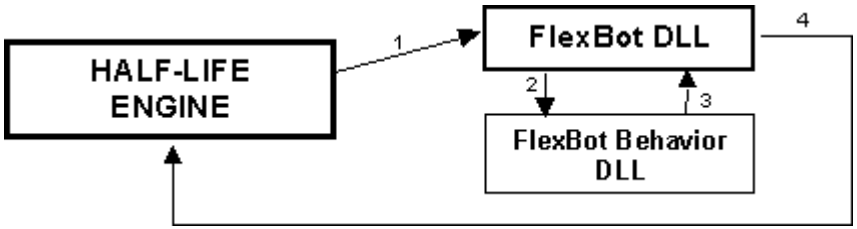
Quelle: http://springrts.com/

Gliederung

- Einführung
  - Testumgebung für verteilte KI „meets“ Videospiel
  - Projekt JARTS
- Vergleichbare Arbeiten
  - FlexBot
  - GameBots
  - AgentContest
  - State of the Art?
- Abgrenzung der Arbeiten von JARTS
  - Szenariobedingte Unterschiede
  - Unterschiedliche Zielsetzungen
  - Gemeinsamkeiten?
- Fazit

FlexBot

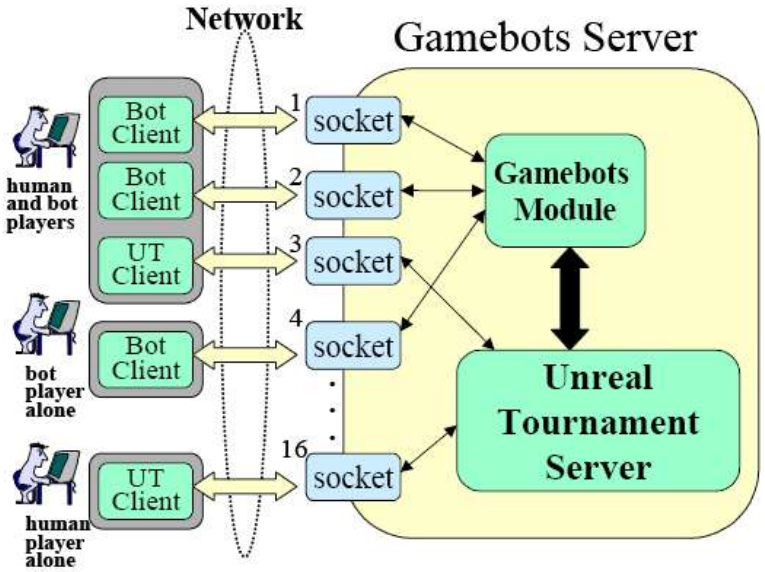
- Modification (Mod) für Half-Life
- Besteht aus einer DLL mit folgenden Eigenschaften:
  - Kapselung der Kommunikation mit Game-Server
  - DLL wird von Half-Life geladen:
    - Aufruf weiterer DLLs, die dann den eigentlichen Bot-Code enthalten
  - Bietet Sensoren und Effektoren per Methodenaufruf



Quelle: http://www.cs.northwestern.edu/~gdunham/flexbot/about.htm

GameBots

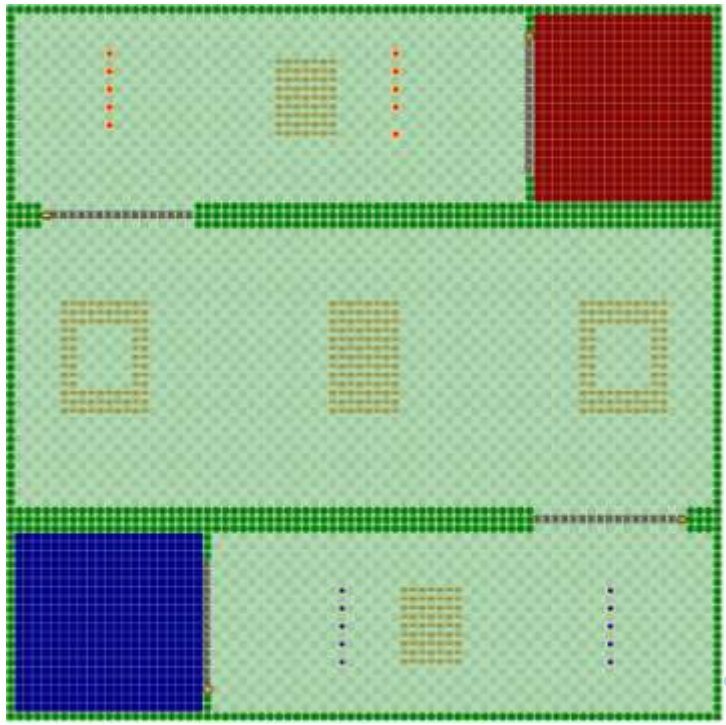
- Modul für Unreal Tournament (99/2003/2004)
- Kapselt Kommunikation mit dem Server:
  - Clients verbinden sich per TCP mit dem GameBot-Server
  - Sensor- und Effektordaten über festgelegtes Nachrichtenprotokoll
  - Plattformunabhängigkeit



Quellen: http://www.cs.cmu.edu/~galk/Publications/01/gamebots.pdf , http://www.unrealtournament2003.com/ut2003/screenshots.html

AgentContest

- Frei verfügbare Serverimplementation eines spezifischen Umgebungsszenarios
  - Verbindung über TCP
  - Sensoren/Effektoren realisiert durch Nachrichtenaustausch im XML-Format
  - Rundenbasiert



Quelle: Screenshot des Servermonitors

State of the Art?

- Bot-Frameworks „alt“
- Contestumgebung nur spielähnlich
  - Rundenbasiert => keine Echtzeitanforderung
  - Szenario wenig komplex (Agenten besitzen nur Bewegungseffektoren)

⇒ State of the Art schwer formulierbar

These: „Die bestehenden Frameworks reichen trotz ihres Alters aus, um die für sie gesetzten Ziele immer noch zu erfüllen und das Nutzen/Aufwand Verhältnis Frameworks für modernere Computerspiele zu erstellen wäre zu gering.“

Warum also trotzdem ein weiteres Framework?

Gliederung

- Einführung
  - Testumgebung für verteilte KI „meets“ Videospiel
  - Projekt JARTS
- Vergleichbare Arbeiten
  - FlexBot
  - GameBots
  - AgentContest
  - State of the Art?
- Abgrenzung der Arbeiten von JARTS
  - Szenariobedingte Unterschiede
  - Unterschiedliche Zielsetzungen
  - Gemeinsamkeiten?
- Fazit

Szenariobedingte Unterschiede

- Bisher nur Bot-Frameworks für First Person Shooter Spiele
- JARTS soll ein Framework für ein Realtime Strategy Spiel werden

⇒ Völlig andere Anforderungen an ein MAS

  - interessante Fragestellungen:
    - Instanzierung von zusätzlichen Agenten als „Commander“, „Sub-Commander“, etc. ?
    - Spielerunterstützung (sehr enge Mensch-Agent-Beziehung)
    - Testumgebung für Selbstorganisierte Systeme?

- Eheste Ähnlichkeit zur AgentContest Umgebung, aber…
  - …Echtzeitanforderung
  - …größere Komplexität des Szenarios (+ viele Effektoren)

Unterschiedliche Zielsetzungen

- Zielsetzung der Bot-Framework Testumgebungen:
  - Hauptziel: Einsatz in Forschung/Lehre
  - Sekundär: Testen der Glaubwürdigkeit von Künstlicher Intelligenz
- Zielsetzung der AgentContest Testumgebung:
  - Identifizierung von Schlüsselproblemen bei der Entwicklung von MAS
  - Bewertung von MAS-spezifischen Tools und Methodologien
- Zielsetzung der JARTS-Framework Testumgebung:
  - Einsatz in der Lehre
  - Bewertung von Geschwindigkeit/Stabilität eines MAS
  - Testen der Glaubwürdigkeit von Künstlicher Intelligenz
  - Testen der Möglichkeit Agenten zur Spielerunterstützung zu nutzen
  - Vorläufig nur für Jadex (respektive andere Java-basierte Systeme)

Gemeinsamkeiten?

- Sowohl die Bot-Frameworks als auch JARTS setzen auf bestehenden, komplexen Videospielen auf
  - ⇒ Vorteil: wegfallender Aufwand für die Entwicklung der eigentlichen Testumgebung (mit steigender Komplexität ist Balancing nötig => immenser Aufwand)
  - ⇒ Nachteil: Änderbarkeit der Umgebung nur bedingt möglich (Level-Editoren, Mod-Programmierung)

- GameBots, AgentContest und JARTS setzen auf eine TCP-Verbindung
  - => Bei JARTS aufgrund des interessanten Aspekts der Spielerunterstützung in sehr komplexen Spielen (das mit JARTS laufende MAS zwischen eigentlichem Game-Client und Game-Server)

Gliederung

- Einführung
  - Testumgebung für verteilte KI „meets“ Videospiel
  - Projekt JARTS
- Vergleichbare Arbeiten
  - FlexBot
  - GameBots
  - AgentContest
  - State of the Art?
- Abgrenzung der Arbeiten von JARTS
  - Szenariobedingte Unterschiede
  - Unterschiedliche Zielsetzungen
  - Gemeinsamkeiten?
- Fazit

Fazit

- Framework zur Anbindung eines Multiagentensystems an ein komplexes Realtime Strategy Spiel tatsächlich etwas Neues!

- Klare Abgrenzung zu den Zielen bestehender Bot-Frameworks und auch zu denen verschiedener Contests, die mit Testumgebungen aufwarten (auch wenn natürlich Überschneidungen in der Zielsetzung bestehen)

Literatur

- ADOBBATI, Rogelio ; MARSHALL, Andrew N. ; SCHOLER, Andrew ; TEJADA, Sheila: Gamebots: A 3D Virtual World Test-Bed For Multi-Agent Research. In: Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS, 2001
- BALTHASAR, Gregor: Agentenbasierte Schwarmsteuerung in einer kompetitiven, dynamischen Umgebung, HAW Hamburg, Bachelorthesis, 2008. – URL http://opus.haw-hamburg.de/volltexte/2008/634/pdf/ba_thesis_Gregor_Balthasar.pdf. – Zugriffsdatum: 2009.06.14
- BALTHASAR, Gregor ; SUDEIKAT, Jan ; RENZ, Wolfgang: On coordinating of artificial cowboys: Using jadex to implement herding agents. Erscheint in: Programming Multi-Agent Systems, 6th International Workshop, ProMAS 2008, Revised and Selected Papers, Springer, 2008
- BEHRENS, Tristan M. ; DASTANI, Mehdi ; DIX, Jürgen ; NOVÁK, Peter: Agent Contest Competition: 4th Edition. In: Postproceedings of Sixth International Workshop on Programming Multi-Agent Systems, ProMAS'08, LNAI, Springer, 2008

Literatur

- BRAUBACH, Lars ; POKAHR, Alexander ; LAMERSDORF, Winfried: Extending the Capability Concept for Flexible BDI Agent Modularization. In: Proc. Of PROMAS-2005, 2005
- BUSETTA, Paolo ; HOWDEN, Nicholas ; RÖNNQUIST, Ralph ; HODGSON, Andrew: Structuring BDI Agents in Functional Clusters. In: ATAL ’99, Springer, 2000, S. 277–289. – ISBN 3-540-67200-1
- DUNHAM, Greg: FlexBot Beta 1 Manual. – URL http://www.cs.northwestern.edu/~gdunham/flexbot/manual/index.htm. – Zugriffsdatum: 2009.06.13
- LIVINGSTONE, Daniel: Turing’s test and believable AI in games. In: Comput. Entertain. 4 (2006), Nr. 1, S. 6. – ISSN 1544-3574
- RUSSEL, Stuart J. ; NORVIG, Peter: Artificial Intelligence: A Modern Approach. Pearson Education, 2003. – ISBN 0-13-080302-2
- WOOLDRIDGE, Michael: An introduction to multiagent systems. Wiley, 2002

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?