



spieleentwicklertreffen.NRW

Vortrag

KI in Sportspielen

Tim Janus, Leitender Programmierer, Z-Software GmbH

Die KI in „Handball Simulator 2010 –
European Tournament“

11.06.2010



Inhaltsübersicht

- Handball Simulator? Worum geht's?
- Struktur der Lösung
 - Hierarchy Goal driven
- Python zum Tweaken
- Das Tool „Tactical Move“
- Die Technik hinter „Tactical Move“



spieleentwicklertreffen.NRW

Was war es für ein Projekt?

Was sind die Randdaten?

Was die besonderen Herausforderungen?

HANDBALL SIMULATOR?

WORUM GEHT'S?



spieleentwicklertreffen.NRW

Handball Simulator 2010

- Start Apr. 2009 → Release Jan. 2010 (zur EM)
- 2-3 Programmierer, 1 Artist & Gamedesigner
- + Externe Mitarbeiter u. Handballbegeisterte





Handball Simulator 2010

- Erstes Handball-Action Spiel weltweit
- Herausforderungen:
 - Animationen (Vortrag von Oliver und Nils)
 - KI (Unser heutiges Thema)
 - Taktisches Gruppenverhalten
 - Enger Spielraum
 - Komplizierte Regeln



spieleentwicklertreffen.NRW

Eine hierarchische Ziel getriebenes Verhalten.

STRUKTUR DER LÖSUNG

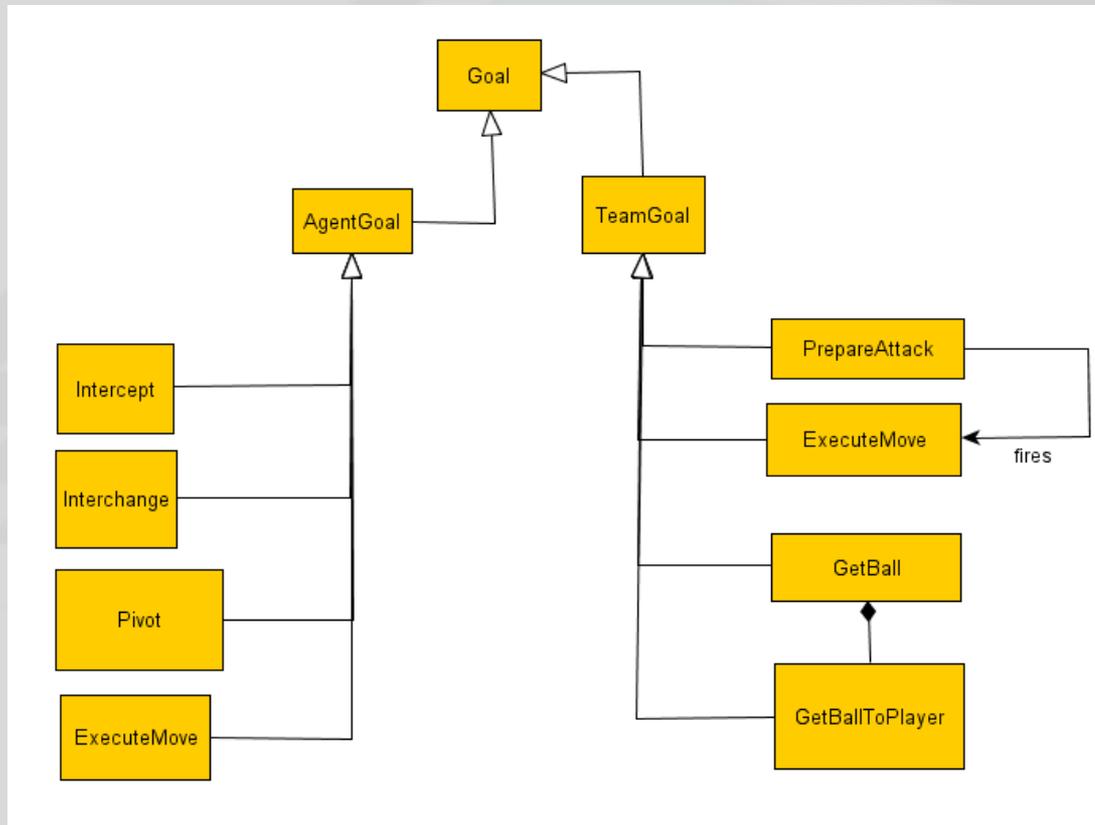


Ziele die die Architektur erfüllen soll:

- Auf einem hohem Level leicht verständlich
- Wiederverwendbarkeit
- Semantische Unterscheidung zwischen Team und Agent



Ausschnitt: UML ähnliches Diagramm





Welche Art „Goals“ gibt es?

- Agent
 - Autonome Goals z.B.: „Pivot“, „Interchange“
 - Atomare Goals z.B.: „Pass“, „Throw“, „MoveTo“
- Team
 - Regel Goals z.B.: „GoalKeeperThrow“, „Kickoff“
 - Taktische Goals z.B.: „PrepareAttack“, „Defendx_x“
 - Sub-Goals: z.B.: „GetBallToPlayer“



Was bedeutet Hierarchisch?

- Sub-Goal: GetBall – Verwendungen in:
 - Throw In
 - Kickoff
 - GetBallToPlayer (genauerer im nächsten Kapitel)
 - Auch als taktisches Goal



Wie läuft die Auswahl ab?

- Agent KI (Manager für Goals)
 - Implementiert auch eine reaktive KI
 - Blocken und Abfangen vom Ball
 - Standardselektion der Goals (sehr rudimentär)
 - Spezielle Goal Auswahl über aktives Team Goal
- Team KI (Manager für Goals)
 - Wählt nach Spielzustand ein Ziel aus



Ein paar Zahlen und Fakten

- Ca. 35 Verschiedene Goals
- Ca. 10 Aktionen → 7 Atomare Agenten Goals
- 15 autonome Agenten Goals
- 5 Regel Goals auf Teamebene
- 4 Taktische Goals auf Teamebene
- 2 Reine Sub-Goals auf Teamebene



spieleentwicklertreffen.NRW

Plug-Into the hierarchy

PYTHON ZUM TWEAKEN UND RAPID-PROTOTYPEN



Wieso Scripting?

- Hotloading (Turn-Around-Time)
- Tweaken
 - Durch Bewertungsfunktionen in den Subgoals
 - Passverhalten Reaktionszeiten, wer holt den Ball
- Rapid-Prototypen
 - Kameraverhalten in Python
 - Schnelle Perspektive Änderungen, u.Ä.



Wieso Python?

- Schon Erfahrungen
 - Build Script in Python
 - Während des Studium (Physiker beeindrucken)
- Zeitdruck
 - Hätte, wäre, wenn ;-)



Plug-Into the Hierachy

- GetBallToPlayer
 - Subgoal: GetBall
 - Delegate: rate_player
- Einige Delegates sind Python Funktionen.
- Standard-Implementierung meist C++.



spieleentwicklertreffen.NRW

Visuelles Erstellen von taktischen Spielzügen

DAS TOOL „TACTICAL MOVE“



Wieso reicht Python nicht aus?

Scripter (im Allg.)

- (+)Technisches Verständnis
- (o) Handball Verständnis

Handball-Experte (im Allg.)

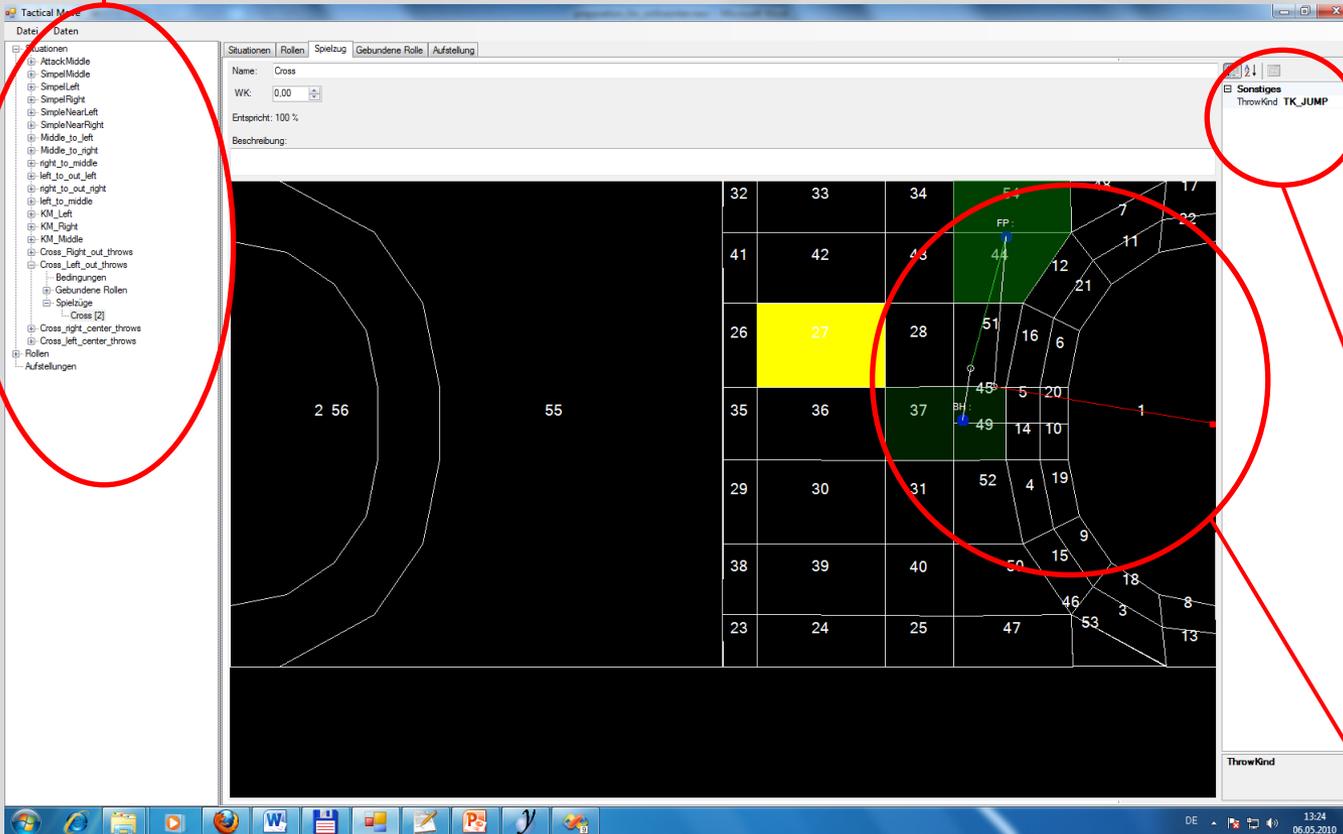
- (--)Technisches Verständnis
- (+++) Handball Verständnis

Mögliche Lösungen:

- Stellen-Ausschreibung (mit wenig Erfolgsaussichten)
- Tool als Brücke für den Handball-Experten



„Tactical Move“ – Das Tool 1/4



Situationen Baumartig
Situation
Bedingungen
Gebundene Rollen
Rolle
Spielzüge

Befehlsparameter
Propertygrid

Visuelle Darstellung
Ablauf Spielzug
Agenten Positionen



„Tactical Move“ – Das Tool 2/4

- Situationen
 - Vorbedingungen (exposed Engine Methods)
 - GetPassivePlayTime(), BallInGoalArea(Own)
 - Gebundene Rolle (VBs, Position auf Spielfeld)
 - Spielzüge
 - Befehlskette für jede gebundene Rolle



„Tactical Move“ – Das Tool 3/4

- Rollen
 - Eine Menge von Vorbedingungen
 - Akt. Instanzen: BallHolder, FieldPlayer, KM
 - Angedacht: Instanz Identifizierung „Pascal Hens“
- Weitere Funktion:
 - Aufstellung festlegen



Tactical Move – Das Tool 4/4

- Mögliche Verbesserungen:
 - Usability
 - Events a'la B angekommen → A loslaufen
 - Bessere Debugging Möglichkeiten
 - Nur rudimentär vorhanden
 - Idee von AIGameDev Interview:
 - Codegeneration in Python statt eigenes Format



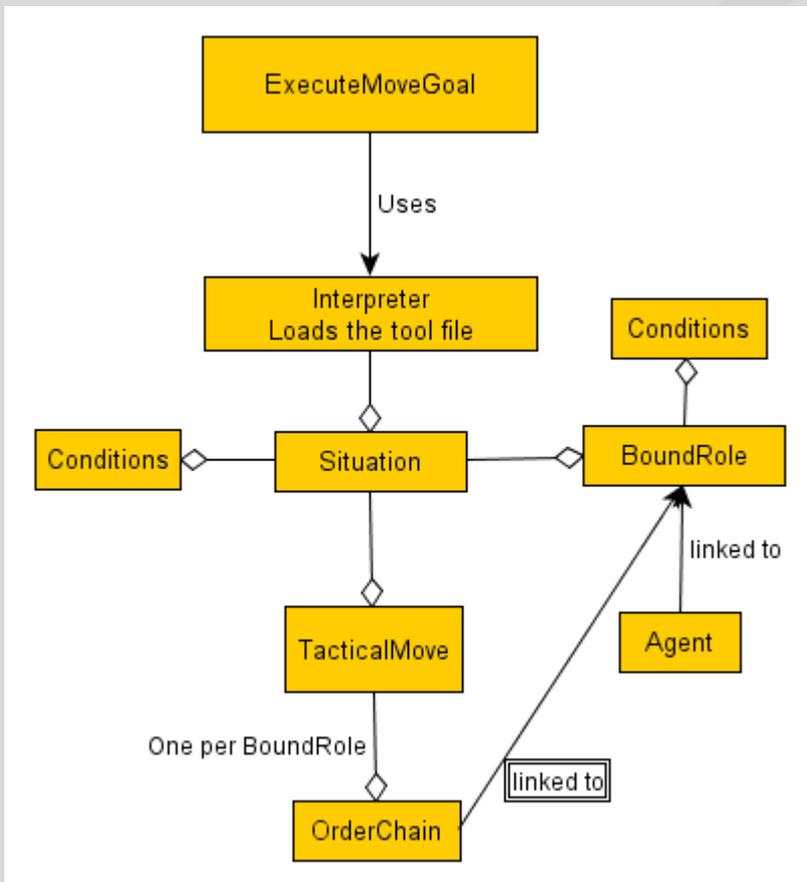
spieleentwicklertreffen.NRW

Eine Art Script-Interpreter

DIE TECHNIK HINTER „TACTICAL MOVE“



Eigentlich auch nur eine Scriptsprache!



- Interpreter:
 - Finde gültige Sit.
 - Wähle eine aus.
 - Versetze Agenten in ExecuteMoveGoal.
 - Überwache die Ausführung.
 - Nur eine Situation gleichzeitig aktiv



Gültige Situationen

- Vorbedingungen der Situationen erfüllt?
 - Für alle gebunden Rollen Agent finden:
 - Position auf Spielfeld
 - Vorbedingungen (z.B. Kreisläufer)
 - Agent als gebunden markieren
 - Bindung der Agenten wieder aufheben
 - Wiederholen bis keine Situation mehr vorhanden



Auswahl

- Situationen
 - 1. Priorität
 - 2. Häufigkeit
 - 3. Zufällig
- Spielzüge
 - 1. Häufigkeit
 - 2. Zufällig

Grund?



Mögliche Unterbrechungen

- Ballverlust
 - Ball abgefangen / abgeprallt
- Spielzustand
 - Tritt in den Torkreis
 - Ball im Aus
 - Ablauf der Passiv-Spielzeit



Debugmöglichkeiten

- Viele Infos
 - Besser mehr Kontrolle über Ausgabe.
- Next Step:
Laden von Situationen





Vielen Dank für die Aufmerksamkeit



Noch viel Spaß auf dem
set.NRW und dem LGF! :=)

- Zeit für:
 - Fragen!
 - Vorschläge!
 - Konstruktive Kritik!