

Julian Huppertz, minf2747, Informatik-Seminar SS2007, Prof. Dr. Sebastian Iwanowski

Spiele KI

Eine Übersicht über verschiedene Spieltypen und ihre
Erfordernisse an die Spiele KI

Inhaltsverzeichnis

1.0 <u>Einleitung, oder: Was ist künstliche Intelligenz?</u>	3
2.0 <u>Spiele KI</u>	3
3.0 <u>Die Klassifikation von Computerspielen</u>	4
3.1.0 <u>Actionspiele</u>	4
3.1.1 <u>Half-Life 2</u>	4
3.1.2.0 <u>Agenten</u>	7
3.2.0 <u>Strategiespiele</u>	8
3.2.1 <u>Command & Conquer 3: Tiberium Wars</u>	8
3.2.2 <u>Historie</u>	10
3.2.3 <u>Pathfinding-Algorithmen</u>	10
3.2.4 <u>Realistische Wegfindung</u>	11
3.3.0 <u>Sportspiele</u>	12
3.3.1 <u>FIFA 2007</u>	12
3.4.0 <u>Adventures bzw. RPGs</u>	14
3.4.1 <u>World of Warcraft</u>	14
3.4.2 <u>Ausblick</u>	16
3.5.0 <u>Simulationen</u>	16
3.5.1 <u>Die Sims 2</u>	16
4.0 <u>Abschluss</u>	18
5.0 <u>Anhänge</u>	19
5.1 <u>Bildverzeichnis</u>	19
5.2 <u>Literaturverzeichnis</u>	19

1.0 Einleitung, oder: Was ist künstliche Intelligenz?

Oft herrscht die Vorstellung vor, künstliche Intelligenz sei der „Mensch als Maschine“. Ein Computer der kreativ denken kann, komplizierte Problemstellungen lösen kann, kurz: sich so verhält, wie es ein Mensch in jeweiliger Situation tun würde.

Doch wonach beurteilen wir Intelligenz bei einem Menschen? Louis Thurstone (US-Amerikanischer Psychologe) gibt eine gute allgemeine Definition. Er ist der Meinung, Intelligenz ist eine Kombination aus folgenden Einzelfaktoren: Räumliches Vorstellungsvermögen, Rechenfähigkeit, Sprachverständnis, Wortflüssigkeit, Gedächtnis, Wahrnehmungsgeschwindigkeit und logisches Denken.

Hier zwingt sich die folgende Frage auf: Warum bezeichnen wir einen Menschen der derartige Fähigkeiten beherrscht oft als intelligent, den Computer an dem er arbeitet aber nur selten, und dass, obwohl man doch davon ausgehen können sollte, dass ein Computer zumindest große Teile oben genannter Fähigkeiten besser beherrscht. *[UNISTUTT]*

Ganz einfach: Er ist nach wie vor nur eine Maschine, welche von Menschen konstruiert und programmiert wird. Er löst komplizierte Problemstellungen nur in dem Maße, in welchem sein Programmierer fähig war, ihm ein angemessenes Problemlösungsverfahren zu implementieren.

Alan Turing (britischer Logiker & Mathematiker) entwickelte den nach ihm benannten Turing-Test, wonach ein Mensch einem Computer und einem Menschen Fragen stellt. Dies geschieht ohne Sichtkontakt, also ohne dass der Fragesteller weiß, wer wer ist. Kann er anhand der Antworten den Mensch nicht mehr von der Maschine unterscheiden, so gilt sie laut Turing als intelligent. *[WIKI: KI]*



[Bild 1] – Turing Test

2.0 Spiele KI

Verschiedene Spieltypen stellen verschiedene Anforderungen an eine künstliche Intelligenz. Heutzutage basiert kaum ein Spiel mehr auf nur einem Algorithmus für künstliche Intelligenz. Der Großteil aller Spiele vereint und kombiniert die verschiedensten Algorithmen um eine möglichst authentische KI zu ermöglichen.

Ich werde anhand 5 sich aktuell auf dem Markt befindlichen Spielen die verschiedenen Genres analysieren und deutlich machen, welche Anforderungen an die künstliche Intelligenz gestellt werden.

3.0 Die Klassifikation von Computerspielen

Eine Klassifikation von Computerspielen stellt sich als nicht einfach heraus, wie während der Arbeiten an diesem Dokument festgestellt habe.

Das liegt daran, dass die Computerspiele-Industrie nicht nur ständig neue Spiele auf den Markt bringt, sondern auch daran, dass viele Spiele aus einem so genannten Genre-Mix bestehen. Das heißt sie lassen sich nicht einer Art von Spiel zuordnen, sondern gehören gleich mehreren Genres an. Allein Wikipedia bietet schon ca. 50 verschiedene Computerspiel-Genres an zwischen denen sich unterscheiden lässt.

Hinzu kommt, dass eine Klassifikation auf Grund der oben genannten Faktoren in wenigen Jahren schon wieder als veraltet gilt. Vor einem guten Jahrzehnt z.B. ordnete man Spiele (nach Fritz/Fehr) in die Kategorien „Abstrakte Denk- und Geschicklichkeitsspiele“, „Kampfspiele“, „Funny-Games“, „Simulationen“ und „Spielgeschichten“ ein. [UNIKOELN]

Zu heutiger Zeit lassen sich die Spiele, welche sich zurzeit am Markt befinden in viele verschiedene Genres einordnen. Ich habe mich dazu entschlossen, die folgenden Genres mit jeweils einem Beispiel vorzustellen:

- Action Spiele
- Strategie – Spiele
- Sport – Spiele
- Simulationen
- Adventures, bzw. RPGs

3.1.0 Actionspiele

Bei Actionspielen handelt es sich um die Art von Spiele, an welche zu erst gedacht wird, wenn man von einem Computerspiel spricht. Während heutzutage Actionspiele zu großen Teilen als „Ballerspiele“ (oder noch schärfere Formulierungen) bezeichnet werden, liegt die ursprüngliche Bedeutung in dem Wort „Action“ (engl.: Handlung). Ein Actionspiel fasziniert einen Spieler dadurch, dass es ihm eine Handlung bietet, der er folgen kann. Es wird eine Aufgabe gestellt, die es zu meistern gilt.

3.1.1 Half-Life 2

Viel gelobt, von vielen gekannt: Half-Life 2. Es wurde von der Firma Valve entwickelt und im Jahr 2004 auf den Markt gebracht.

Ich möchte im Folgenden an dem Beispiel Half-Life 2 vorstellen und erläutern, welche Anforderungen ein 3D-Action Spiel (in diesem Fall: ein Ego-Shooter) an die künstliche Intelligenz stellt.

In Half-Life 2 kommen viele Techniken der künstlichen Intelligenz zum Einsatz. So gibt es nicht nur stupide Leveldekoration mit denen der Spieler interagieren kann (beispielsweise Knöpfe, Schalter und Hebel), sondern es gibt viele verschiedene vom Computer gesteuerte Widersacher, die versuchen dem Spieler das Leben schwer zu machen.

Diese gegnerischen Einheiten sollen nicht bloß reagieren (schießt du auf mich, schieße ich zurück), sondern vielmehr selbstständig und autonom handeln. Das ist es ja, was eine glaubwürdige künstliche Intelligenz ausmacht. Der Anspruch an die künstliche Intelligenz von Half-Life 2 verlangt von den gegnerischen Einheiten nicht nur dass sie einen geeigneten Weg von A nach B finden, dass sie beurteilen können wie sie am schnellsten von einem Punkt zum anderen gelangen, sondern auch dass sie auf den Spieler reagieren. Sie müssen sich je nach Situation unterschiedlich verhalten. Z.B. erfordern es manche Situationen, dass die künstlichen Gegenspieler Teams bilden, sich gegenseitig unterstützen, Deckung geben, um den Spieler so zu überlisten.

Folgender Screenshot stellt eine typische Situation aus Half-Life 2 dar. Der Spieler (nicht zu sehen) kämpft mit Teamkameraden gegen feindliche Aliens. Dabei handelt jede vom Computer gesteuerte Einheit ganz autonom:



[Bild 2] – Eine Spielsituation aus Half-Life 2

An Hand des obigen Screenshots möchte ich noch mal ein paar Anforderungen deutlich machen und schlicht weg aufzählen:

- Die vom Computer gesteuerten Teamkameraden müssen zwischen „gut“ und „böse“ unterscheiden können

- Sie müssen sich in eine Position bringen können, aus derer sie den Feind gut treffen können, er sie aber nicht. In diesem Fall befinden sich die „Guten“ quasi im ersten Stock, der Feind überwiegend im Erdgeschoss
- Sie müssen – ausgehend von einem Zielpunkt, den sie erreichen wollen – einen Weg dorthin finden können. Dabei ist es wichtig, dass sie sich nicht gegenseitig im Wege stehen und behindern
- Sie müssen in der Lage sein, die Situation eigenständig zu beurteilen und entsprechend zu agieren, Beispiel:
- Wer befindet sich in der Überzahl? Die „Guten“ oder die „Bösen“? Soll ich flüchten?
- Ist es sinnvoll wenn ich einem Kameraden zu Hilfe eile oder soll ich versuchen möglichst auf eigene Faust viele gegnerische Einheiten auszuschalten?
- Reicht meine Munition noch oder sollte ich besser davonlaufen?
- Habe ich irgendwelche taktischen Vorteile, wenn ich beispielsweise Lampen oder Ähnliches zerschleße?

Wie man leicht erkennen kann, werden immense Anforderungen an ein derartiges Spiel gestellt, um möglichst glaubhaft und realistisch zu wirken.

Doch sollen Nicht-Spieler-Charaktere (im Folgenden NPCs genannt) nicht nur dazu dienen, dem menschlichen Spieler einen Gegenspieler zu bieten, viel mehr kommen in vielen aktuellen Ego-Shootern auch NPCs vor, welche dem Spieler ein Freund sind und ihn in seinem Handeln unterstützen. Und es wäre doch langweilig, wenn ein derartiger befreundeter vom Computer gesteuerter Spieler bloß hinter seinem menschlichen Verbündeten her rennt. Hilfreich wäre also eine Erkennung des NPCs, was der Spieler mit seinem Handeln zu bezwecken versucht, um ihn angemessen zu unterstützen.

Beispielsweise könnte der NPC erkennen, dass wenn der Spieler gerade schleicht, er besser nicht Aufmerksamkeit auf sich ziehen möchte. Der NPC könnte also sich eben so verhalten, ruhig sein und dem Spieler folgen. Lädt der Spieler gerade seine Munition nach, so sollte der NPC nach Möglichkeit dem Spieler Deckung geben und ihn beschützen. Das primäre Ziel ist es also, den Spieler mit sinnvollen Tätigkeiten zu unterstützen, in den verschiedensten Spielsituationen und nicht den Spieler in seinen Absichten zu stören indem er „nicht nachdenkt“ bevor er handelt.

Den NSCs sollte es natürlich ebenso möglich sein, miteinander zu agieren. So ist es nicht nur von Vorteil, wenn sie den Spieler in seinem Handeln unterstützen, sondern sie sollten ebenso sich gegenseitig unterstützen. Gemeinsam mit einem Verbündeten NSC Taktiken planen, ihn beschützen und dergleichen anderer Handlungen.

Anhand des Beispiels „Half-Life 2“ möchte ich nun die so genannten Agenten vorstellen, welche in fast allen aktuellen Action-Spielen Einsatz finden. Natürlich kommen in Half-Life 2 auch noch viele weitere Strategien und Methoden zum Einsatz, der Ansatz von Agenten bietet allerdings einen guten Einblick über die gewünschten Fähigkeiten von Computer gesteuerten (Gegen-)Spielern.

3.1.2.0 Agenten

Eine gute Definition eines Agenten liefert uns Michael Wooldridge:

“An agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment on behalf of its user or owner (figuring out what needs to be done to satisfy design objectives, rather than constantly being told)” [UNIBONN]

Ein Agent ist also ein vom Computer gesteuerter Spieler, welcher aus eigener Initiative heraus handelt und ebenso auf Änderungen in seiner Umwelt reagiert. Darüber hinaus können Agenten durchaus soziale Fähigkeiten besitzen, d.h. mit anderen Agenten kommunizieren, wie sie es im Beispiel Half-Life 2 auch sind.

Folgende drei Typen eines Agenten gibt es:

3.1.2.1 Reflex-Agent

Der Reflex-Agent beschreibt die einfachste Art der Agenten. Er nimmt über seine Sensoren einen Zustand der Umwelt auf, wertet diesen aus, wählt eine entsprechende Reaktion aus und führt diese durch.

Ein Beispiel: Ein Agent nimmt wahr, dass ihm seine Munition ausgeht. Logische Konsequenz: Er lädt seine Waffe nach.

Ein Reflex-Agent kann darüber hinaus noch mit verschiedenen Zuständen oder auch Stati arbeiten, d.h. einmal wahrgenommene Umwelt-Zustände speichern.

Beispiel: Ein mit dem Computerspieler verbündeter menschlicher Spieler schießt auf den Computerspieler. Beim ersten Mal passiert noch nichts, außer dass der Computerspieler dem menschlichen Spieler eine Warnung schickt à la „Hey, ich bin in deinem Team!“. Passiert es zu einem späteren Zeitpunkt erneut, dass derselbe menschliche Spieler auf den Computerspieler schießt, „weiß“ er, dass der menschliche Spieler schon einmal auf den Computerspieler geschossen hat und erwidert das Feuer.

3.1.2.2 Ziel-orientierter Agent

Ziel-orientierte Agenten sind eine Weiterentwicklung von Reflex-Agenten. Sie können ein Ziel haben und es erreichen. Reflex-Agenten können dies nicht, da sie keine Vorstellung davon haben, wie sich die Umwelt verändert, wenn sie bestimmte Aktionen auslösen, und somit auch nicht wissen können, ob sie ihrem Ziel näher kommen oder nicht.

Wenn man das vorhergegangene Beispiel fortführt, würde ein Ziel-orientierter Agent beispielsweise das Ziel haben, auf keinen Fall den menschlichen Spieler zu provozieren. Wenn man davon ausginge, dass dieser das Feuer wiederum erwidern würde, sobald der Computerspieler auf den menschlichen Spieler schießt, würde es in diesem Fall bedeuten, dass der Computerspieler nur weitere Warnungen an den Spielern sendet, anstatt ihn direkt zu beschießen.

3.1.2.3 Nutzen-basierter Agent

Nutzen-basierte Agenten können Ziele in Teilziele aufteilen. D.h. sie kombinieren diese Teilziele mit einem Erreichungsgrad. Kann ein Ziel nicht mehr komplett erreicht werden, so wird versucht, ihm so nahe wie möglich zu kommen. [UNISTUTT]

3.2.0 Strategie-Spiele

Strategiespiele lassen sich meist einteilen in rundenbasierte Strategiespiele und Echtzeitstrategiespiele. In Strategiespielen geht es meist darum eine bestimmte Anzahl von Einheiten gegen den Computerspieler kämpfen zu lassen. Wobei man oft die Möglichkeit hat, sich eine Basis aufzubauen, verschiedene Technologien zu erforschen um so immer bessere Einheiten bauen zu können. Echtzeitstrategiespiele verlangen von einem Spieler die Koordination von allen Dingen gleichzeitig. Da die Echtzeitstrategiespiele (im Folgenden mit RTS abgekürzt) weiter verbreitet sind und meiner Meinung nach die interessantere Art von Strategiespielen sind, werde ich ein Beispiel aus dem Bereich der Echtzeitstrategiespiele vorstellen, an welche die KI auch höhere Anforderungen stellt, da nicht nur der Spieler alle oben genannten Dinge gleichzeitig erledigen muss, sondern der Computer auch gleichzeitig den verschiedensten Strategien nachgehen muss.

3.2.1 Command & Conquer 3: Tiberium Wars

Das aktuellste Kind der als Urvater der Strategiespiele geltenden Reihe, entwickelt ursprünglich von den Westwood Studios, welche 2003 allerdings in die bestehenden Strukturen von Electronic Arts L.A. integriert wurde. Auf den Markt gebracht wurde das Spiel in diesem Jahr.



[Bild 3] Screenshot Command & Conquer 3: Tiberium Wars

Als Vorzeige-Beispiel der Strategiespiele ist es für so ein Spiel natürlich auch Pflicht, eine gute und überzeugende künstliche Intelligenz auf die Beine zu stellen. Die Entwickler von Command & Conquer 3: Tiberium Wars haben ihre Hausaufgaben gemacht und es geschafft eine gelungene künstliche Intelligenz zu programmieren.

Ein Echtzeitstrategiespiel verlangt vom Computer folgende, nicht unbedingt triviale Überlegungen anzustellen:

- Wegfindung und Terrainanalyse:
 - Wie kommt meine gesamte Armee von meiner Basis zur gegnerischen Basis?
 - Gibt es auf dem Weg Hindernisse? Gibt es zum Beispiel Flüsse, die man nicht ohne eine Brücke zu finden überqueren kann?
 - Wie kommen meine 30 Panzer vom Typ „Mega-Bumm“ nacheinander über eben jene kleine Brücke, oder sich gegenseitig den Weg zu versperren?
 - Gibt es auf dem Weg zur gegnerischen Basis Terrain welches ich lieber nicht überqueren sollte, da es meinen Einheiten schadet?
 - Wo baue ich was #1? (In C & C gibt es Gebiete wo häufiger mit so genannten Ionenstürmen zu rechnen ist)
 - Wo baue ich was #2? Wie richte ich meine Basis am besten ein, wo soll ich Verteidigungsanlagen hinbauen und wo zusätzliche Truppen zur Verteidigung postieren?
 - Gibt es bestimmtes Terrain, welches mir einen taktischen Vorteil bietet?
- Ressourcen-Planung, Forschung und Entwicklung:
 - Ich habe 27.500\$, wie gebe ich Sie am sinnvollsten aus um auch auf lange Sicht den größtmöglichen Nutzen daraus zu ziehen?
 - Soll ich das Geld lieber in eine neue Generation von Waffen investieren, oder baue ich lieber noch eine neue Raffinerie, um anschließend nicht nur die neuen Waffen zu besitzen, sondern auch neue Verteidigungsanlagen für meine Basis?
 - Es gibt so viele verschiedene Möglichkeiten neue Technologien zu erforschen.. welchen Weg soll ich nehmen und welche Technologie erforschen?
 - Wie spielt der Spieler? Baut er mehr Luft-Einheiten oder mehr Boden-Einheiten? (Es ist natürlich sinnvoll, sich der Taktik des Spielers entsprechend anzupassen)

- Taktiken/Strategien:
 - Wie spielt der menschliche Spieler und wie kann ich ihn am besten besiegen?
 - Welche Formation ist für diesen Angriff am geeignetsten?
 - Da schnelle Einheiten bei einem Komplettangriff der eigenen Armee vorher bei dem Gegner angekommen sind, müssen sich die Einheiten vor einem Angriff dort also erst wieder sammeln
 - Soll ich lieber in kurzer Zeitfolge Einheiten nachproduzieren, um sie in meinem Angriff unterstützen zu lassen oder häufe ich mir besser eine große Armee an, um meinen Feind zu überrennen?

Auch in Echtzeitstrategiespielen gibt es also zahlreiche Fragen, die „eine künstliche Intelligenz“ sich stellen muss, um erfolgreich zu sein. Das war auch bloß ein kurzer Abriss, natürlich gibt es in Strategiespielen ebenso die oben schon genannten Agenten.

3.2.2 Historie, Beispiele aus der Entwicklungsgeschichte von Strategie-KI

Im Folgenden eine kurze Übersicht, wie sich die künstliche Intelligenz von Anfang der 90er bis zur Jahrtausendwende verändert haben.

1992 Dune II, fast nur geskriptete Ereignisse
 1994 Warcraft I, KI zieht sich nie zurück, greift blind an
 1995 Command & Conquer, enorme Pathfinding Probleme der Tiberium Sammler
 1995 Warcraft II, „peon traffic jams“, 3 Ressourcen
 1996 C&C Red Alert, Waypoints, KI Bündnisse
 1997 Total Annihilation, Line of Sight
 1998 Starcraft, 3 ausgewogene völlig unterschiedliche Rassen, Script fähig
 1999 AoE2, intelligente Formationen, gutes Pathfinding
 ... heute: C&C 3, bessere Terrain Analyse, „hier könnte Hinterhalt passieren“, taktische Analyse der Situation

3.2.3 Pathfinding-Algorithmen

Da in Echtzeitstrategiespielen unter anderem die Wegfindung hervorsteht, möchte ich anhand des A*-Algorithmus kurz erklären, wie so ein Pathfinding abläuft.

3.2.3.0 A*-Algorithmus

Der A* - Algorithmus wird von Kennern oft als der ultimative Pathfinding-Algorithmus bezeichnet, und das nicht zu Unrecht, liefert er doch unter bestimmten gegebenen Voraussetzungen den optimalen Weg zum Ziel und das in einem akzeptablen Zeitrahmen. [A*]

3.2.3.1 Vorgehensweise beim A*-Algorithmus

Es wird immer der geschätzte kürzeste Weg zum Ziel bevorzugt. Das Schätzen eines Weges basiert auf einer Heuristik, also einem Wert der die ungefähre Entfernung zum Ziel angibt. Man

bezeichnet diese Vorgehensweise auch als eine informierte Suche. Im Gegensatz zu einer uninformierten Suche werden zunächst die Wege untersucht, welche am wahrscheinlichsten und am schnellsten zum Ziel führen.

Der A*-Algorithmus verwendet zwei Listen, eine „OPEN“-Liste und eine „CLOSED“-Liste. In der „OPEN“-Liste stehen alle bisher erreichbaren und noch nicht besuchte Wegpunkte, in der „CLOSED“-Liste alle bereits besuchten Wegpunkte. Zu jedem Wegpunkt werden ausserdem die Kosten gespeichert, also z.B. die Dauer, dieses Feld zu erreichen. Für eine „OPEN“-Liste verwendet man als Implementierung üblicherweise eine Prioritätswarteschlange, da hier die Wegpunkte mit den geringsten Kosten ganz vorne stehen.

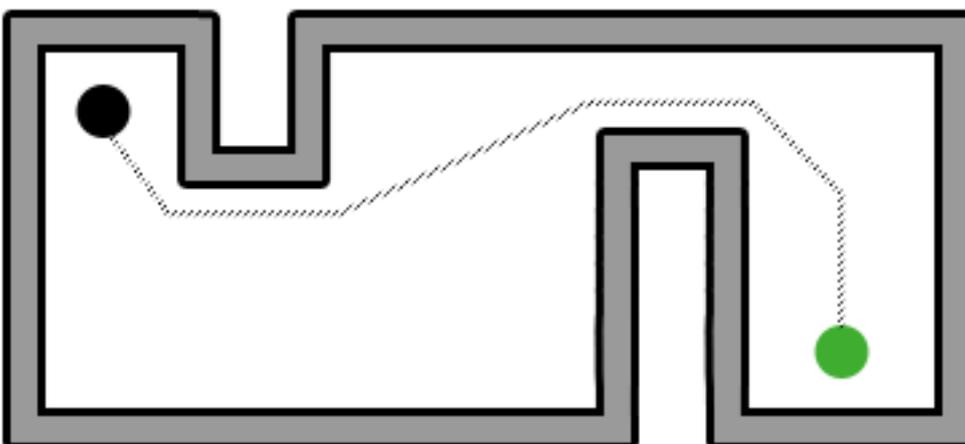
3.2.3.2 Cooperative Pathfinding

Es gibt ein Problem bei der Wegfindung, welches sicherlich schon zu Frustration geführt hat. Einheiten versperren sich gegenseitig den Weg, während man versucht den Angriff einer gegnerischen Armee abzuwehren. Anstatt sich gegenseitig Platz zu machen, blockieren sie den Weg. [AIGPW3]

3.2.4 Realistisch wirkende Wegfindung

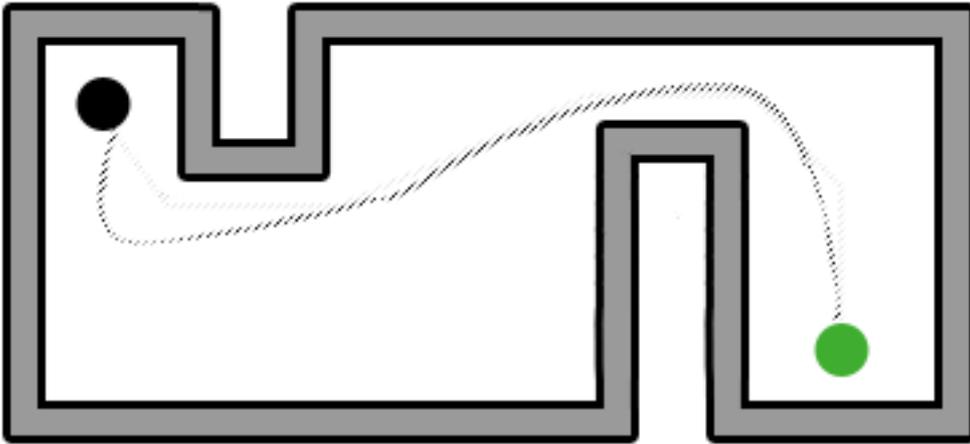
Hat eine Einheit einen Weg zum Ziel erst einmal gefunden, möchte sie ihn natürlich auch bestreiten. Damit das ganze natürlich aussieht verwendet man häufig eine so genannte Catmull-Rom Interpolation zwischen den einzelnen Wegpunkten.

Folgendes Beispiel macht den Unterschied deutlich: Eine Einheit soll sich in einem Raum von einem Startpunkt zu einem Zielpunkt bewegen. Ein Wegfindungsalgorithmus generiert folgendes Ergebnis.



[Bild 4] Harte Kanten geben den Ton an

Dieser Weg sieht sehr kantig aus und wirkt keinesfalls realistisch. Interpoliert man nun aber zwischen den einzelnen Punkten mit Hilfe einer Catmull-Rom-Interpolation, sieht das ganze schon viel natürlicher aus:



[Bild 5] Ein Weg, geglättet mit Hilfe der Catmull-Rom-Interpolation

Dies ist nur ein Ansatz, um computergenerierte Wege realistischer erscheinen zu lassen. Natürlich darf es, auch nicht passieren, dass Einheiten durch Gegenstände durchlaufen können, bzw. dass einzelne Körperteile z.B. einfach durch eine Tür hindurch gehen. Des Weiteren muss man darauf achten, dass – gerade wenn man menschliche Computergegner simulieren will – sich die Einheiten mit angemessener Geschwindigkeit durch die Spielwelt bewegen. [AIGPW3]

3.3.0 Sportspiele

Sportspiele tanzen nicht so sehr aus der Reihe, wie man vielleicht im ersten Moment vermuten mag. Natürlich bilden Sportspiele ein komplett eigenständiges Genre, allerdings muss auch hier geprüft werden, wie beispielsweise Fußballspieler von einem zum anderen Tor kommen, ohne dabei eigene Spieler über den Haufen zu rennen. Oder es müssen Taktiken und Formationen angewandt werden, um dem menschlicher Spieler ein möglichst spannendes und anspruchsvolles Spiel bieten zu können.

Ich möchte Anhand des Fußballspiels Fifa 2007 die Anforderungen erklären, welche an die KI für Fußballspiele gestellt werden.

3.3.1 FIFA 2007

Das von EA Sports entwickelte Fußballspiel ist durch seine komplette Lizenzierung durch die FIFA das Fußballspiel schlechthin. Seit 1994 gibt es fast jedes Jahr ein neues FIFA-Spiel.



[Bild 6] FIFA 2007

Folgende Situation beschreibt ganz gut, welche Anforderungen darüber hinaus noch an eine KI für Fußballspiele gestellt werden müssen:

- Anpfiff: Die Aufstellung beschreibt die grundlegende Strategie
- Pass zum Mitspieler: Der Pass muss so ausgeführt werden, dass er
- nicht von einem gegnerischen Spieler abgefangen werden kann
- der Taktik gerecht wird (offensives bzw. defensives Spiel)
- in Sachen Schussstärke so gewählt wurde, dass ein Mitspieler den Ball erreichen kann
- Entwicklung des Spiels, passen an Mitspieler, in Tor-Schussweite bringen
- Wahl einer geeigneten Taktik für Torversucht
- Alle Spieler in Position bringen
- Torschuss, wobei dieser in Stärke, Winkel und Richtung so ausgeführt werden muss, um entweder als Flanke einem Mitspieler eine bessere Gelegenheit zu bieten ein Tor zu erzielen, oder so, dass der gegnerische Torwart nicht an den Ball heran kommt und das Runde möglichst im Eckigen landet
- Der menschliche Spieler steuert genau einen der 11 Spieler. Der Computer muss also das Zusammenspiel und die Steuerung der restlichen 10 Spieler koordinieren und aufeinander abstimmen

Bei einem Fußballspiel kommt es vor allen Dingen darauf an, dass man den gegnerischen Spielern ausweicht, dass man sie umspielt um sie so gerade nicht an den Ball kommen zu lassen. Dieses Ausweichen, bzw. Umspielen wird in Fußballsimulationen häufig in Form von Negativ-Gewichten realisiert, welche in Matrixform auf das Spielfeld gelegt werden. Man kann hier auch sehr gute Parallelen zu der Terrainbewertung in beispielsweise Strategiespielen erkennen.

[FUBAKI]

Zusätzlich finden sich meist Implementierungen von sogenannten Zustandsautomaten in den Fußballspielen wieder. Jeder Zustand beschreibt dabei die aktuelle Aufgabe des Spielers. Es gibt beispielsweise Zustände wie „Ich habe den Ball“, „Ich bin Ball-nächster Spieler“ oder „Ich bin verletzt“. Gerade bei dem letzten Beispielzustand sieht man, dass es noch weitere Anforderungen gibt. So muss der Computer „überlegen“, wann er eine Auswechslung vornimmt, denn es stehen ihm in einem Spiel üblicherweise nur 3 zur Verfügung.

Darüber hinaus muss es nicht nur für verschiedene andere Spielsituationen noch ganz andere Taktiken geben, z.B. für einen Freistoß oder einen Eckstoß, sondern auch Unterscheidungen in der Wahl der Taktik in Abhängigkeit vom Ballbesitz und Spielweise der gegnerischen Mannschaft gemacht werden.

3.4.0 Adventures bzw. RPGs

Gerade die Rollenspiele (engl.: Role-Playing-Games, RPG) nehmen ein immer größer werdendes Marktsegment ein, wie sich an dem MMORPG (engl.: Massive Multiplayer Online Role Playing Game) World of Warcraft erkennen lässt. Was sich ebenfalls erkennen lässt, ist, dass der Trend von klassischen Adventures (wie z.B. „Monkey Island“, „Sam & Max“, „Day of the Tentacle“, alles Spiele der Firma „Lucas Arts“) weg geht. Bei den klassischen Adventures schlüpfte man in die Rolle des Helden und erlebt dessen Geschichte nach. Bei den MMORPGs schlüpft man zwar auch in die Rolle eines Helden, ist aber direkter mit ihm verknüpft. So entwickelt man den Charakter über Monate, manchmal sogar Jahre hin, und erlebt gemeinsam mit anderen Spielern gleichzeitig eine Geschichte.

Besonders in MMORPGs ist die künstliche Intelligenz meist nicht so vordergründig wie in anderen Genres, z.B. in Strategiespielen. Das liegt hauptsächlich daran, dass die Spielehersteller Ressourcen sparen möchten. Würden zusätzlich zu den tausenden gleichzeitig gespielten Accounts auch noch eine KI hinzu kommen, die alle Spieler berücksichtigt, würde das Unmengen an Server Ressourcen verbrauchen. Also hat man hier gespart.

Ich habe mir als Beispiel das zurzeit immer noch erfolgreichste MMORPG rausgesucht, World of Warcraft.

3.4.1 Künstliche Intelligenz in World of Warcraft

World of Warcraft wird von der Firma Blizzard entwickelt und hat schon in den ersten 24 Stunden seit Verkaufsbeginn neue Rekorde aufgestellt. Mittlerweile gibt es über 7 Millionen Menschen weltweit, die dieses Spiel aktiv spielen.



[Bild 7] World of Warcraft

Natürlich kommen in World of Warcraft auch die gleichen grundlegenden Techniken zum Einsatz, wie sie auch in anderen Spielen Verwendung finden. So spielt die Wegfindung eine große Rolle (auch wenn diese in World of Warcraft nicht so ausgereift ist, wie in anderen Spielen, so bekommt man des Öfteren Gegner zu sehen, welche bei der Verfolgung der Spieler in Bäumen hängen bleiben o.ä.). Wenn Gegner z.B. auf Patrouille gehen, laufen sie im Vorwege festgelegte Wegpunkte ab. Befindet sich beim Patrouillengang ein menschlicher Spieler in Reichweite (hier kommen wieder einmal die Agenten zum Einsatz), greifen sie ihn an. Auch kommt es vor, dass Ansätze von sozialen Algorithmen zum Tragen kommen, so eilen einer gegnerischen Kreatur Artgenossen zur Hilfe falls diese um Hilfe schreit und sich die Verbündeten in Reichweite befinden. Man kann also nicht sagen, dass es kaum künstliche Intelligenz in World of Warcraft gibt, allerdings sind fast alle Ereignisse in World of Warcraft gescriptet, d.h. der Ablauf der Ereignisse ist vorher festgelegt..

Als Beispiel für so ein gescriptetes Ereignis lässt sich ein Boss Kampf anbringen, der Kampf gegen einen „Endgegner“ in einem bestimmten Gebiet. Der Kampf läuft immer nach demselben Schema ab, hat der Computergegner einen bestimmten Prozentsatz an Lebensenergie erreicht, oder haben die vom Menschen gesteuerten Widersacher ein bestimmtes Ereignis ausgelöst, begibt sich der Boss in eine nächste Phase und verändert beispielsweise die Art des Angriffs.

[GPD]

Die logische Konsequenz hieraus ist ganz klar ersichtlich: Hat eine eingespielte Gruppe von Spielern erst einmal raus, wie sie den Boss besiegen kann, läuft der Kampf nach immer dem gleichen Schema ab und wird schnell zur Routine. Der Vorteil an der Sache: Der Kampf ist für die menschlichen Spieler einfach und es fällt der Gruppe nicht schwer an die Beute zu gelangen. Aber auch der Nachteil liegt auf der Hand: Solche derartigen Kämpfe werden schnell langweilig, einfach weil nichts Aufregendes passiert, was den Spielern neue Herausforderungen bietet.

Dies ist auch ein Grund dafür, warum häufig auf neuen Content gewartet wird, dass neue Gebiete eingeführt werden mit neuen Herausforderungen. Hat man einmal eine gewisse Stufe im Spiel erreicht und ist allen Gefahren erfolgreich begegnet, weiß man häufig nicht mehr was man sinnvolles tun kann und wartet auf Updates des Spiels.

Des Weiteren gibt es in vielen MMORPGs auch so genannte World Events, also Ereignisse die einen Großteil der Spielwelt – und somit auch viele Spieler – betreffen. Aber selbst solche Ereignisse sind alle schon im Vorwege geskriptet und laufen genau so ab, wie die Programmierer es vorherbestimmt haben.

3.4.2 Ausblick

MMORPGs mit KIs, welche auch dem Namen gerecht werden. Irgendwann in naher Zukunft wird die Technik soweit ausgereift sein, dass so etwas möglich ist. Vom Computer gesteuerte Einheiten, die selbstständig agieren, sich Taktiken „ausdenken“, um den Spieler zu überlisten, in Hinterhalte zu locken oder mit immer neuen Angriffsmustern zu besiegen.

3.5.0 Simulationen

Simulationen bilden wie der Name schon vermuten lässt einen (begrenzten) Teil der Realität im Computer ab, er wird simuliert. Häufig ordnet man auch Spiele anderer Genres den Simulationen zu, so zum Beispiel „Action-Simulationen“ oder „Sportsimulationen“. Da diese allerdings meiner Meinung nach besser in die jeweiligen oben beschriebenen Genres passen, gehören meiner Meinung nach in den Bereich der Simulationen die „Wirtschaftssimulationen“ und die „Life-/God-Simulationen“.

Die „Wirtschaftssimulationen“ bilden den meiner Meinung nach aus Sicht der künstlichen Intelligenz weniger interessanten Teil des Genres ab. So brauchen sie zwar auch grundlegende Algorithmen wie z.B. Wegfindungs-Techniken oder ähnliches, allerdings passiert auch hier meistens nichts spannendes, keine Taktik die sich die KI selber „hat einfallen lassen“, weil sie beobachtet hat, wie der Spieler spielt. Zu den klassischen Wirtschaftssimulationen gehören Spiele wie „Anno 1703“, „Die Siedler“ und ähnliche Spiele.

„Life-/God-Simulationen“ beinhalten Spiele wie „Black & White 2“ oder auch „Die Sims 2“. Gerade bei „Black & White 2“ kommen Algorithmen zum Tragen, welche es den vom Spieler gesteuerte Kreaturen ermöglichen zu lernen, wie der Spieler sich in bestimmten Situationen verhält, beispielsweise Strafen, um so etwas in Zukunft zu entgehen.

Ich werde anhand des Beispiels „Die Sims 2“ vorstellen, welche Anforderungen eine „Life-/God-Simulation“ an die künstliche Intelligenz stellt.

3.5.1 Die Sims 2

Die Sims 2 wurde von Maxis entwickelt und von EA Games vertrieben. Es handelt sich um die Nachfolge Version des Klassikers Die Sims, welche zu damaliger Zeit einzigartig war. Bei den Sims geht es darum, das Leben einer Familie vom Computer simulieren zu lassen und zu beeinflussen. Die Sims haben Bedürfnisse (unter anderem wollen sie essen, arbeiten, ihre Freizeit genießen und Freundschaften schließen. Der Benutzer hat Einfluss auf alle diese Aktivitäten und Bedürfnisse. Neu an „Die Sims 2“ ist unter anderem, dass nun jeder Sim über Gene verfügt und phänotypische Eigenschaften an seine Nachkommen weitergibt. Es werden also auch genetische Algorithmen mit ins Spiel gebracht.



[Bild 8] Eine typische Szene aus „Die Sims 2“

Neu ist das so genannte „Smart Terrain“ bzw. „Message Passing“, Algorithmen zur künstlichen Intelligenz welche von Maxis entwickelt wurden. Diese Algorithmen bieten eine gute Möglichkeit die in einem Simulations-Spiel wie „Die Sims 2“ auftretenden Anforderungen an eine künstliche Intelligenz zu erfüllen.

Diese Techniken bieten die Möglichkeit, Charaktere welche bestimmte Bedürfnisse haben in einer virtuellen Welt zu steuern, dies geschieht indem der Spieler die Prioritätenliste eines Sim anpasst. Ist der Mensch einmal passiv, so werden die Geschehnisse durch vorbestimmte Algorithmen weitersimuliert, wobei die aktuellen Bedürfnisse eines Sim „beobachtet“ werden und entsprechend ihrer Dringlichkeit zu entsprechenden Handlungen führen. [UNIMAGDE]

Folgendes Beispiel: Ein Sim hat das Bedürfnis zu trinken. Ein Schrank in der Küche signalisiert, dass er etwas zu trinken beinhaltet. Der Sim geht also zum Schrank, öffnet ihn, holt sich etwas zu trinken heraus und trinkt. Hat der Sim erst einmal getrunken, stellt sich schnell das Bedürfnis ein, die Toilette aufzusuchen. Die Toilette im Badezimmer signalisiert „Hier kannst du hingehen“, also geht der Sim auf die Toilette. So wird es dem Sim möglich gemacht etwas zu trinken, und auch anschließend den körperlichen Bedürfnissen nachzukommen.

In „Die Sims 2“ kommen natürlich auch grundlegende Algorithmen der künstlichen Intelligenz zum Tragen, wie z.B. Wegfindungsalgorithmen, was für den Benutzer aber besonders reizvoll und faszinierend ist, ist es die Beziehungen zwischen den Sims zu beeinflussen. Denn hat es der Benutzer erst einmal geschafft, und mit seinem Sim das Herz eines anderen Sim erobert, können mit es Glück die genetischen Algorithmen zur Verwendung gelangen. [WIKI:SIMS2]

Genetische Algorithmen benutzen im Gegensatz zu anderen Algorithmen ein Optimierungsverfahren. Das heißt, es wird von einer schlechten Lösung eines Problems ausgegangen und optimiert, bis man eine optimale Lösung erreicht hat. Dies ist ein Vorteil von genetischen Algorithmen gegenüber anderen Algorithmen, welche ein Problem bis ins kleinste Detail hin modellieren, um eine Lösung zu finden. Diese Modellierung kann oft sehr aufwendig und zeit-intensiv sein.

Natürlich ist in „Die Sims 2“ keine komplette Evolution bzw. Vererbung nachgebildet, wie sie in der Natur Verwendung findet, es wurde vielmehr ein Modell entwickelt, welches dazu dient grundlegende phänotypische Eigenschaften genauso weiterzuerben wie „seelische“ (womit die Charaktereigenschaften eines Sim gemeint sind).

4.0 Abschluss

Ich hoffe ich konnte euch einen kleinen Einblick in die breitgefächerte, bunte Welt der Spiele bieten und Lust darauf machen, die verschiedenen Algorithmen der künstlichen Intelligenz zu verstehen. Denn natürlich bereiten immer neue Genres und immer neue Ansprüche der Spielehersteller möglichst realistische Spiele zu erstellen, auch neue Probleme. Aber was wäre ein Problem ohne eine entsprechende Lösung. Für jede Anforderung wurden entsprechende Algorithmen entwickelt und eine Auswahl an Möglichkeiten die geschaffen wurden, werden in den kommenden Seminarvorträgen vorgestellt.

5.0 Anhänge

5.1 Bildverzeichnis

Bild 1: <http://www.relentlesslyoptimistic.com/>

Bild 2: <http://www.hthiele.de/>

Bild 3: <http://www.keller-kinder.de>

Bild 4, Bild 5: Eigenproduktion

Bild 6: <http://www.juegosblog.com>

Bild 7: <http://download.net.pl>

Bild 8: <http://www.4players.de>

5.2 Literaturverzeichnis

Fast alle gesammelten Fakten wurden eigenhändig gesammelt und zusammengetragen. Wo doch Zitate oder die Gedanken andere als Grundlage dienen, ist dies entsprechend gekennzeichnet.

[UNISTUTT] - <http://www.iis.uni-stuttgart.de/lehre/ws06-07/ki/>

[WIKI:KI] - http://de.wikipedia.org/wiki/Künstliche_Intelligenz

[UNIKOELN] - <http://www.uni-koeln.de/ew-fak/psycho/Petzold/>

[UNIBONN] - <http://www.informatik.uni-bonn.de/~alda/presentations/Vergleich.ppt>

[A*] - <http://www.geosimulation.de/Umsetzungen.html>

[AIGPW3] - AI Game Programming Wisdom 3, Itps Thomson Learning, ISBN-13: 978-1584504573

[FUBAKI] - http://fai80.informatik.uni-erlangen.de/inf8/en/alte_vorlesungen.html

[UNIMAGDE] - <http://www.wisg.cs.uni-magdeburg.de/%7Ehartmann/CS2/control.pdf>

[GPD] - Game Programming für Dummies, Wiley-VCH, ISBN-13: 978-3527700974

[WIKI:SIMS2] - http://de.wikipedia.org/wiki/Sims_2