Ludwig-Maximilians-Universität München
Institut für Informatik
PD Dr. Matthias Schubert
Gregor Jossé

München, 04.05.2016

Managing Massive Multiplayer Online Games

SS 2016

Übungsblatt 3: Konfliktmanagement und Dead Reckoning

Besprechung: 12.05.2016

Betrachten Sie im Folgenden ein abstraktes Spiel, bei dem sich Spieler in einer virtuellen zwei-dimensionalen Welt bewegen. Jeder Spieler S besitzt eine positive Anzahl an Lebenspunkten $S.L \in \mathcal{N}$. Ein Spieler S_i kann in diesem Spiel folgende Aktionen durchführen:

- $Heilen(S_2, n)$ erhöht die Lebenspunkte eines Spieler S_2 um den Wert n. Falls $S_2.L + n > 100$, so wird $S_2.L$ auf den Wert 100 gesetzt.
- $Hauen(S_2, n)$ reduziert die Lebenspunkte eines Spieler $S_2 \neq S_1$ um den Wert n. Falls $n > S_2.L$, so gilt Spieler S_2 als *tot* und kann in diesem Spiel keine weitere Aktionen mehr durchführen.

Aufgabe 3-1 *Konflikte*

Betrachten Sie nun eine Instanz dieses Spieles, in der die folgenden Aktionsrequest gesendet werden. Initial habe jeder Spieler 50 Lebenspunkte, d.h. $\forall 1 \leq i \leq 3 : S_i.L = 50$. Zur Kommunikation verwende dieses Spiel eine Client-Server Architektur mit zentraler Zeitverarbeitung, d.h., die Ausführungsreihenfolge der Aktionen wird vom Server entschieden. Zur Vereinfachung nehmen wir an, des die Latenzzeit, sowohl für die Übertragung einer Aktion zum Server sowie zum Schicken eines Updates vom Server zu einem Client zwei Ticks beträgt.

Spieler	Aktion	Zeit(Client)
S_2	Hauen(S_1,60)	1
S_1	Hauen(S_2,30)	2
S_1	Heilen(S_1, 80)	3
S_2	Heilen(S_2, 60)	4
S_2	Hauen(S_3,30)	5
S_3	Hauen(S_2,50)	6
S_2	Hauen(S_3,30)	7

Zur Lösung von Konflikten soll der Lösungsansatz *Rücksetzen von lokalen Aktionen* verwendet werden.

(a) Wie läuft das Spiel auf Seite des Servers ab?

Spieler	Aktion	Zeit(Server)	Ergebnis
S_2	Hauen(S_1,60)	1+2	S_1 tot
S_1	Hauen(S_2,30)	2+2	ignoriert weil S_1 tot
S_1	Heilen(S_1, 80)	3+2	ignoriert weil S_1 tot
S_2	Heilen(S_2, 60)	4+2	$S_2.L = 100$
S_2	Hauen(S_3,30)	5+2	$S_3.L = 20$
S_3	Hauen(S_2,50)	6+2	$S_2.L = 50$
S_2	Hauen(S_3,30)	7+2	S_3 tot

(b) Wie läuft das Spiel auf Seite des Clients von Spieler S_1 ab? Welche Anomalien treten auf?

Aktionen anderer Spieler kommen erst nach 4 Ticks bei Spieler S_1 an, während eigene Aktionen sofort verarbeitet werden.

Spieler	Aktion	Zeit(S_1)	Ergebnis
S_1	Hauen(S_2,30)	2	$S_2.L = 20$
S_1	Heilen(S_1, 80)	3	$S_1.L = 100$
S_2	Hauen(S_1,60)	1+4	$S_1.L = 40$
S_1	Rückgängig: Hauen(S_2,30)	2+4	$S_2.L = 50$
S_1	Rückgängig: Heilen(S_1, 80)	3+4	$S_1 tot$

Anomalie: Zum Zeitpunkt 5 erhölt S_1 das Ergebnis vom Server dass er gehauen wird und überlebt obwohl er eigentlich tot ist. Erst zu Zeitpunkt 7 ist Spieler 1 tot. Aktion Hauen wird Rückgängig gemacht.

(c) Wie läuft das Spiel auf Seite des Clients von Spieler S_2 ab? Welche Anomalien treten auf?

Aktionen anderer Spieler kommen erst nach 4 Ticks bei Spieler S_1 an, während eigene Aktionen sofort verarbeitet werden.

Spieler	Aktion	Zeit(S_2)	Ergebnis
S_2	Hauen(S_1,60)	1	S_1 tot
S_2	Heilen(S_2, 60)	4	$S_2.L = 100$
S_2	Hauen(S_3,30)	5	$S_3.L = 20$
S_2	Hauen(S_3,30)	7	S_3 tot
S_3	Hauen(S_2,50)	6+4	$S_2.L = 50$

Anomalie: S_2 wird vom lokal toten Spieler S_3 noch gehauen.

(d) Wie läuft das Spiel auf Seite des Clients von Spieler S_3 ab? Welche Anomalien treten auf?

Spieler	Aktion	Zeit((S_3))	Ergebnis
S_2	Hauen(S_1,60)	1+4	S_1 tot
S_3	Hauen(S_2,50)	6	S_2 tot
S_2	Heilen(S_2, 60)	4+4	$S_2.L = 60$
S_2	Hauen(S_3,30)	5+4	$S_3.L = 20$
S_2	Hauen(S_3,30)	7+4	S_3 tot

S_3 wird vom lokal toten Spieler S_2 zweimal gehauen und getötet. Außerdem sind die Lebenspunkte von S_2 lokal inkorrekt, und müssen vom Server neu synchronisiert werden.

(e) Welche Anomalien würden bei lokal bei S_3 vermieden, wenn die Clients via Peer2Peer miteinander kommunizieren, und zur Lösung von Konflikten ein Lag-Mechanismus mit vier Ticks Delay verwendet wird? Nehmen Sie an, dass die Latenzzeit zur Kommunikation zwischen Clients zwei Ticks betrage.

Offenbar läuft das Spiel wie gewünscht, also wie vom Server berechnet, ab, da die Latenzen alle unter der Lag-Delay-Grenze liegen. Erst wenn Spieler 2 (als einziger Spieler) beispielsweise ein Lag von 8 Ticks hätte, wäre das Ergebnis entschieden anders. (Er würde von Spieler 3 getötet, bevor seine Heilung lokal auf Spieler 3s Rechner ausgefüht werden könnte.)

Spieler	Aktion	Zeit(S_3)	Ergebnis
S_2	Hauen(S_1,60)	1+2+2	S_1 tot
S_1	Hauen(S_2,30)	2+2+2	ignoriert
S_1	Heilen(S_1, 80)	3+2+2	ignoriert
S_2	Heilen(S_2, 60)	4+2+2	$S_2.L = 100$
S_2	Hauen(S_3,30)	5+2+2	$S_3.L = 20$
S_3	Hauen(S_2, 50)	6+0+4	$S_2.L = 50$
S_2	Hauen(S_3,30)	7+2+2	S_3 tot

Die Anomalie die verursacht hat, dass S_3 von einem Phantom getötet wurde, ist behoben.

(f) Diskutieren Sie Vor- und Nachteile dieser Lösungen.

- Beide Lösungen können Anomalien verursachen (falls Latenz von einem Spieler höher)
- Lokal-Lag Ansatz perfekt, falls das lokale Lag-Interval genau der Latzent aller Spieler entspricht.
- Vorteil Client/Server: Fehler können durch synchronisation mit dem Server erkannt und behoben werden.
- Nachteil Client/Server: Der Zustand auf dem Server kann auch unfair sein, falls unterschiedliche Spieler unterschiedliche Latenzzeiten haben.
- usw... bisserl diskutieren

Aufgabe 3-2 *Dead Reckoning*

Um Bandbreite zu sparen, werden die Positionen von Spielern nicht zu jedem Tick an den Server übermittelt. Betrachten Sie den Client eines Spieler S_1, der einen anderen Spieler S_2 wahrnimmt. Der Client von S_1 erhält vom Server folgende Positionsupdates von S_2:

Spieler	x	y	Zeit
S_2	100	100	0
S_2	110	90	15
S_2	130	90	30
S_2	160	50	40

An welcher Position sollte Spieler S_2 zum Zeitpunkt 45 angezeigt werden? Verwenden Sie dazu folgende Vorhersagemodelle:

(a) Die letzte bekannte Position wird unverändert als Vorhersage gewählt.

$x = 160, y = 50$

(b) Es wird zur Vorhersage eine lineare Bewegung mit konstanter Geschwindigkeit angenommen.

Die aktuelle Geschwindigkeit berechnet sich aus den letzten beiden Updates:

Allgemeine Formel:

$$p(t_1 + \Delta t) = p(t_1) + \Delta t \cdot \frac{p(t_1) - p(t_0)}{\|p(t_1) - p(t_0)\|} \cdot \frac{\|p(t_1) - p(t_0)\|}{t_1 - t_0}$$

In diesem Fall: $t_0 = 30, t_1 = 40, \Delta t = 5$, also:

$$\begin{aligned} p(45) &= p(40) + 5 \cdot \frac{p(40) - p(30)}{\|p(40) - p(30)\|} \cdot \frac{\|p(40) - p(30)\|}{40 - 30} \\ &= \begin{pmatrix} 160 \\ 50 \end{pmatrix} + \frac{1}{2} \cdot \begin{pmatrix} 30 \\ -40 \end{pmatrix} \\ &= \begin{pmatrix} 175 \\ 30 \end{pmatrix} \end{aligned}$$

(c) Es wird zur Vorhersage ein eine lineare Bewegung mit konstanter Beschleunigung angenommen.

Die aktuelle Geschwindigkeit berechnet sich aus den letzten drei Updates. Genauer: aus der letzten Bewegung (zwischen den letzten beiden Updates), der aktuellen Geschwindigkeit sowie der Geschwindigkeitsveränderung zwischen den letzten drei Updates. Allgemeine Formel:

$$p(t_2 + \Delta t) = p(t_2) + \Delta t \cdot v(t_2 + \Delta t) \cdot \frac{p(t_2) - p(t_1)}{\|p(t_2) - p(t_1)\|}$$

wobei die Geschwindigkeitsvorhersage $v(t_2 + \Delta t)$ wie folgt definiert ist

$$\begin{aligned} v(t_2 + \Delta t) &= v(t_2) + \Delta t \cdot \frac{v(t_2) - v(t_1)}{t_2 - t_1} \\ &= \frac{\|p(t_2) - p(t_1)\|}{t_2 - t_1} + \Delta t \cdot \frac{\frac{\|p(t_2)-p(t_1)\|}{t_2-t_1} - \frac{\|p(t_1)-p(t_0)\|}{t_1-t_0}}{t_2 - t_1} \end{aligned}$$

In diesem Fall: $t_0 = 15, t_1 = 30, t_2 = 40\Delta t = 5$, folglich:

$$\begin{aligned}
v(45) &= \frac{\|p(40) - p(30)\|}{40 - 30} + 5 \cdot \frac{\frac{\|p(40)-p(30)\|}{40-30} - \frac{\|p(30)-p(15)\|}{30-15}}{40 - 30} \\
&= \frac{\|\begin{pmatrix} 30 \\ -40 \end{pmatrix}\|}{10} + 5 \cdot \frac{\frac{\|\begin{pmatrix} 30 \\ -40 \end{pmatrix}\|}{10} - \frac{\|\begin{pmatrix} 20 \\ 0 \end{pmatrix}\|}{15}}{10} \\
&= \frac{\sqrt{30^2 + (-40)^2}}{10} + 5 \cdot \frac{\frac{\sqrt{30^2+(-40)^2}}{10} - \frac{\sqrt{20^2}}{15}}{10} \\
&= 5 + \frac{1}{2} \cdot (5 - \frac{4}{3}) \\
&= \frac{41}{6} \\
p(45) &= p(40) + 5 \cdot v(45) \cdot \frac{p(40) - p(30)}{\|p(40) - p(30)\|} \\
&= \begin{pmatrix} 160 \\ 50 \end{pmatrix} + 5 \cdot \frac{41}{6} \cdot \frac{\begin{pmatrix} 30 \\ -40 \end{pmatrix}}{\|\begin{pmatrix} 30 \\ -40 \end{pmatrix}\|} \\
&= \begin{pmatrix} 160 \\ 50 \end{pmatrix} + \frac{41}{60} \cdot \begin{pmatrix} 30 \\ -40 \end{pmatrix} \\
&= \begin{pmatrix} 160 \\ 50 \end{pmatrix} + 0.6833 \cdot \begin{pmatrix} 30 \\ -40 \end{pmatrix} \\
&= \begin{pmatrix} 180,5 \\ 22,67 \end{pmatrix}
\end{aligned}$$

Aufgabe 3-3 *Hermite-Kurven*

Gegeben sei folgende Situation in einem abstrakten Spiel auf einer zweidimensionalen Karte: Die Position eines Spielers und dessen Bewegungsrichtung zum Zeitpunkt t seien durch Dead Reckoning gegeben, in Form eines Positionsvektors p_{DR} sowie eines Bewegungsvektors d_{DR}. Vom Server komme zum selben Zeitpunkt ein Update mit den echten Positions- und Bewegungsdaten $p_{\text{EX}}, d_{\text{EX}}$. Nun soll der Client die durch Dead Reckoning errechnete Position und Bewegung in die tatsächliche überführen, und zwar im Zeitfenster Δt. Der Einfachheit halber kann angenommen werden, dass sich der Spieler im Zeitfenster Δt genau um die Länge eines Bewegungsvektors fortbewegt. Das heißt, zum Zeitpunkt $t + \Delta t$ soll sich der Spieler an der Position $p_{\text{EX}} + d_{\text{EX}}$ befinden.

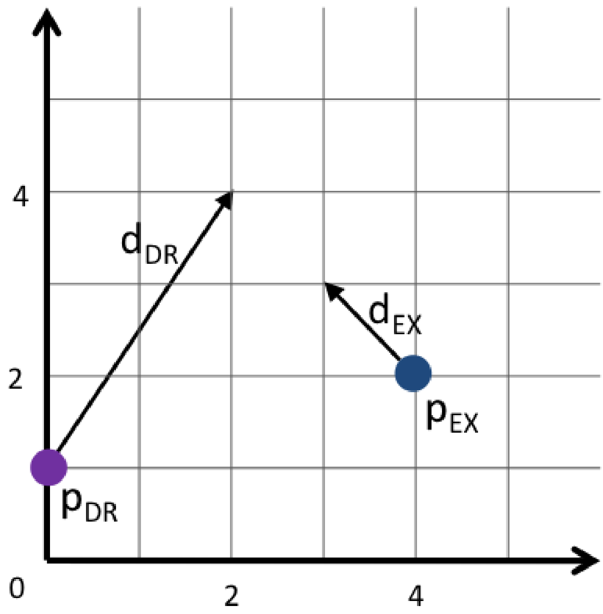
Dabei seien

$$p_{\text{DR}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad d_{\text{DR}} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad p_{\text{EX}} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} \quad d_{\text{EX}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

wie unten dargestellt.

Verdeutlichen Sie sich die Idee der Positionskorrektur per Linearkombination von vier Hermite-Kurven wie im Skript (Kapitel 3, Seite 20) beschrieben. Rechnen Sie dazu den Wert der Linearkombinationsfunktion $\hat{p}(x)$ (siehe unten) für $x \in \{1/2, 7/8\}$ aus. Zeichnen Sie anschließend diese Punkte ein und skizzieren Sie auf dieser Basis ihre Vorstellung der entsprechenden Verbindungskurve.

$$\begin{aligned}
h_1(x) &= 2x^3 - 3x^2 + 1 \qquad & h_2(x) &= -2x^3 + 3x^2 \\
h_3(x) &= x^3 - 2x^2 + x \qquad & h_4(x) &= x^3 - x^2
\end{aligned}$$



$$\hat{p}(x) = p_{\text{DR}} \cdot h_1(x) + (p_{\text{EX}} + d_{\text{EX}}) \cdot h_2(x) + d_{\text{DR}} \cdot h_3(x) + d_{\text{EX}} \cdot h_4(x)$$

wobei $x \in [0, 1]$ die Bewegung vom Zeitpunkt t zum Zeitpunkt $t + \Delta t$ beschreibt.

$1/2$:

$$\begin{aligned} h_1(1/2) &= 1/4 - 3/4 + 1 = 1/2 \\ h_2(1/2) &= -1/4 + 3/4 = 1/2 \\ h_3(1/2) &= 1/8 - 1/2 + 1/2 = 1/8 \\ h_4(1/2) &= 1/8 - 1/4 = -1/8 \end{aligned}$$

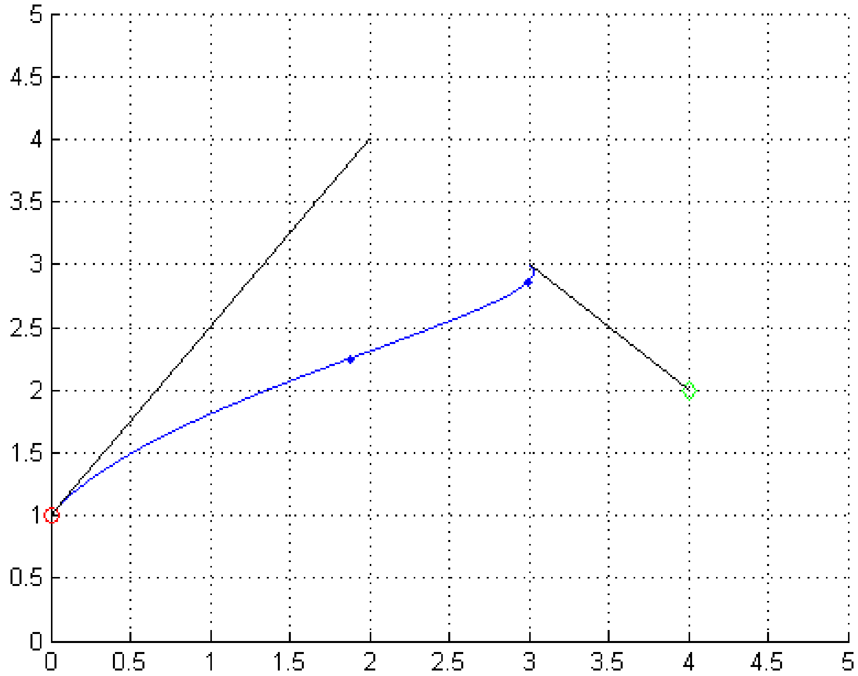
$$\hat{p}(1/2) = 1/2 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 1/2 \cdot \begin{pmatrix} 3 \\ 3 \end{pmatrix} + 1/8 \cdot \begin{pmatrix} 2 \\ 3 \end{pmatrix} - 1/8 \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{15}{8} \\ \frac{18}{8} \end{pmatrix}$$

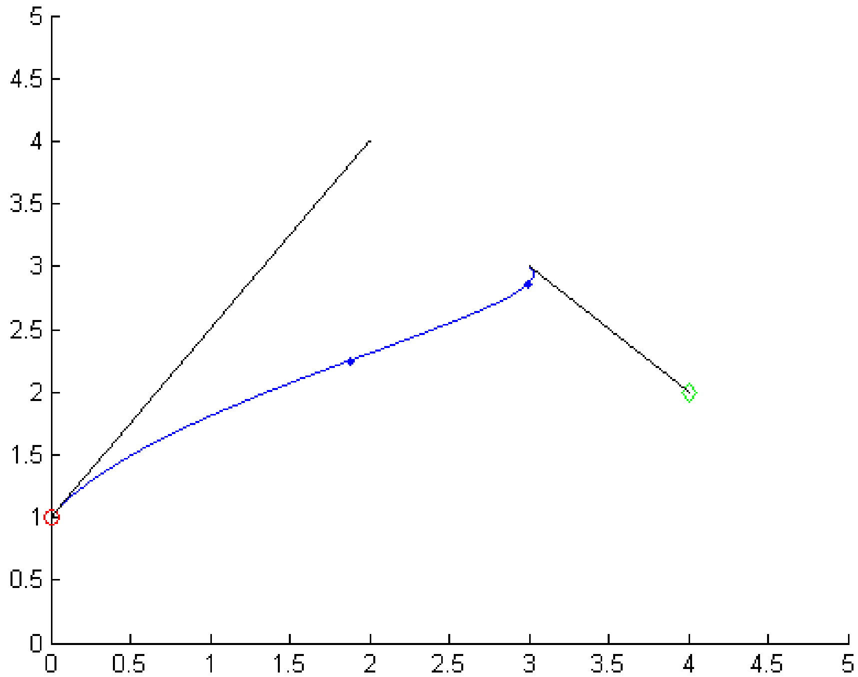
$7/8$:

$$\frac{7}{8}^2 = \frac{49}{64} \quad \frac{7}{8}^3 = \frac{343}{512}$$

$$\begin{aligned} h_1(7/8) &= 343/256 - 147/64 + 1 = 11/256 \\ h_2(7/8) &= -343/256 + 147/64 = 245/256 \\ h_3(7/8) &= 343/512 - 98/64 + 7/8 = 343/512 - 784/512 + 448/512 = 7/512 \\ h_4(7/8) &= 343/512 - 49/64 = 343/512 - 392/512 = -49/512 \end{aligned}$$

$$\hat{p}(7/8) = 11/256 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 245/256 \cdot \begin{pmatrix} 3 \\ 3 \end{pmatrix} + 7/512 \cdot \begin{pmatrix} 2 \\ 3 \end{pmatrix} - 49/512 \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2,9941 \\ 2,8594 \end{pmatrix}$$


0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5