Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Matthias Schubert
Felix Borutta

Munich, May 16, 2019

Managing Massive Multiplayer Online Games

SS 2019

Exercise Sheet 3: Conflict Management and Dead Reckoning

The assignments are due May 22, 2019

Assignment 3-1 *Conflicts*

Consider an abstract game with three players in a two-dimensional world. Each player p has a health value $p.H \in \mathbb{N}$. The initial value is 50 for all players, $\forall 1 \leq i \leq 3 : p_i.H = 50$. A player p_i can perform the following actions:

- $Heal(p_j, n)$ increases the health points of player p_j by n points up to a maximum value of 100, i.e., $Heal(p_j, n) = min(p_j + n, 100)$.
- $Attack(p_j, n)$ reduces the health points of player p_j by n points. If $n > p_j.H$ player p_j is *dead* and cannot perform further actions.

The game uses a client-server architecture with the server handling the processing sequence, i.e. the order of execution is determined by the server. For the sake of simplicity you can assume that the latency is two time steps in both directions, i.e. for the transmission of an action to the server and for the transmission of an update from the server to the client.

Consider the following action requests:

Player	Action	Time(Client)
p_2	Attack(p_1,60)	1
p_1	Attack(p_2,30)	2
p_1	Heal(p_1, 80)	3
p_2	Heal(p_2, 60)	4
p_2	Attack(p_3,30)	5
p_3	Attack(p_2,50)	6
p_2	Attack(p_3,30)	7

How does the game proceed on the server, respectively on the clients? In case of conflicts, solve them by using the *reset local actions* approach.

(a) How does the game proceed on the side of the server?

Player	Action	Time(Server)	Result
p_2	Attack(p_1,60)	1+2	p_1 dead
p_1	Attack(p_2,30)	2+2	ignored because p_1 is dead
p_1	Heal(p_1, 80)	3+2	ignored because p_1 is dead
p_2	Heal(p_2, 60)	4+2	$p_2.H = 100$
p_2	Attack(p_3,30)	5+2	$p_3.H = 20$
p_3	Attack(p_2,50)	6+2	$p_2.H = 50$
p_2	Attack(p_3,30)	7+2	p_3 dead

(b) How does the game proceed on the side of the client of player p_1? What anomalies occur?

Actions from the other players arrive with a delay of time steps at player p_1, while own actions are processed immediately.

Player	Action	Time(p_1)	Result
p_1	Attack(p_2,30)	2	$p_2.H = 20$
p_1	Heal(p_1, 80)	3	$p_1.H = 100$
p_2	Attack(p_1,60)	1+4	$p_1.H = 40$
p_1	Undo: Attack(p_2,30)	2+4	$p_2.H = 50$
p_1	Undo: Heal(p_1, 80)	3+4	p_1 dead

Anomaly: At timestamp 5 p_1 receives the message that he has been attacked and survives although he should die. Just at time 7 player p_1 is dead. His attack action has to be undone.

(c) How does the game proceed on the side of the client of player p_2? What anomalies occur?

Actions of other players arrive after 4 ticks at player p_2, while own actions are processed immediately.

Player	Action	Time(p_2)	Result
p_2	Attack(p_1,60)	1	p_1 dead
p_2	Heal(p_2, 60)	4	$p_2.H = 100$
p_2	Attack(p_3,30)	5	$p_3.H = 20$
p_2	Attack(p_3,30)	7	p_3 dead
p_3	Attack(p_2,50)	6+4	$p_2.H = 50$

Anomaly: p_2 is attacked locally by the already dead player p_3.

(d) How does the game proceed on the side of the client of player p_3? What anomalies occur?

Player	Action	Time((p_3))	Result
p_2	Attack(p_1,60)	1+4	p_1 dead
p_3	Attack(p_2,50)	6	p_2 dead
p_2	Heal(p_2, 60)	4+4	$p_2.H = 60$
			$p_2.H = 100$ *
			$p_2.H = 50$ *
p_2	Attack(p_3,30)	5+4	$p_3.H = 20$
p_2	Attack(p_3,30)	7+4	p_3 dead

* Note: At timestamp 8 we also receive the game state from the server (from timestamp 6) which says that $p_2.H = 100$. This means we have a conflict here and need to reset local changes. Therefore, we must set the health value of p_2 to 100. However, the action p_3 attacks p_2 with value 50, which is overwritten locally by resetting the local game state is not lost. The client saves locally processed actions that have not been acknowledged by the server yet, and re-employs them on the local game state, hence $p_2.H = 50$.[1]

(e) Which anomalies would be prevented locally for player p_3 if the clients would communicate via peer2peer and used a lag-mechanism with four time steps delay to solve conflicts? Assume a latency of two time steps for the communication between two clients.

Apparently the game proceeds as calculated by the server since all latencies are lower than the lower lag-delay-bound.

[1]More precise information can be found here: `https://www.cs.cornell.edu/~wmwhite/papers/2009-ICDE-Scalability.pdf`

Player	Action	Time(p_3)	Result
p_2	Attack(p_1,60)	1+2+2	p_1 dead
p_1	Attack(p_2,30)	2+2+2	ignored
p_1	Heal(p_1, 80)	3+2+2	ignored
p_2	Heal(p_2, 60)	4+2+2	$p_2.H = 100$
p_2	Attack(p_3,30)	5+2+2	$p_3.H = 20$
p_3	Attack(p_2, 50)	6+0+4	$p_2.H = 50$
p_2	Attack(p_3,30)	7+2+2	p_3 dead

The anomaly which caused that player p_3 was killed by a phantom is eliminated.

(f) Discuss the advantages and disadvantages of these solutions!

- Both solution can cause anomalies (if the latency of one player is higher than the latency of the other player)
- Local-lag approach is perfect, if the local lag-interval corresponds exactly to the latency of all players.
- Advantage of client/server model: Errors can be recognized and fixed by the server with the help of synchronization.
- Disadvantage of client/server model: The state of the server can be unfair if players have different latencies.

Assignment 3-2 *Dead Reckoning*

To save bandwidth positions of players are not transmitted every tick. Consider the client of player p_1 who perceives actions of another player p_2. The client of p_1 receives the following position updates of player p_2 from the server:

Player	x	y	Time
p_2	100	100	0
p_2	110	90	15
p_2	130	90	30
p_2	160	50	40

At which position is player p_2 displayed at time 45? Use the following prediction models:

(a) The last known position is used as prediction.

$x = 160, y = 50$

(b) The position is predicted by assuming a linear movement with constant velocity.

The current velocity is calculated with help of the last two updates:

Generic formula:

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

In this case: $t_0 = 30, t_1 = 40, \Delta t = 5$, thus:

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

(c) The position is predicted by assuming a linear movement with constant acceleration.

The acceleration is calculated by using the last three updates. More precisely the last movement (between the last two updates), the current velocity, and the change of velocity between the last three updates are used.

Generic formulas:

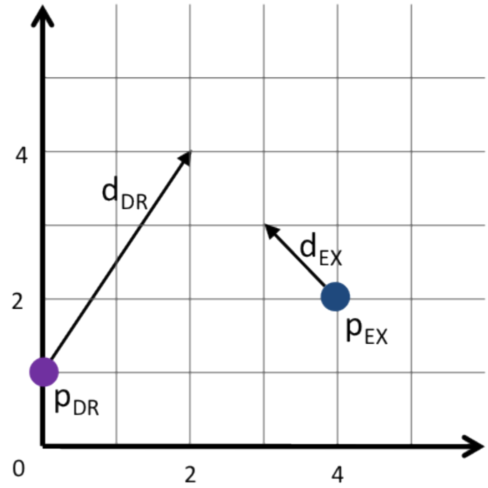
$$\begin{aligned}
p(t_i + \Delta t) &= \frac{1}{2} a(t_i) \Delta t^2 + v(t_i) \Delta t + p(t_i) \\
a(t_i) = \frac{\Delta v}{\Delta t} &\approx \frac{v(t_i) - v(t_{i-1})}{t_i - t_{i-1}} \\
v(t_i) = \frac{\Delta p}{\Delta t} &\approx \frac{p(t_i) - p(t_{i-1})}{t_i - t_{i-1}}
\end{aligned}$$

In this case: $t_0 = 15, t_1 = 30, t_2 = 40$ and $\Delta t = 5$, thus:

$$\begin{aligned}
p(t_2 + \Delta t) &= \frac{1}{2} a(t_2) \Delta t^2 + v(t_2) \Delta t + p(t_2) \\
&\approx \frac{1}{2} \frac{v(t_2) - v(t_1)}{t_2 - t_1} \Delta t^2 + v(t_2) \Delta t + p(t_2) \\
&\approx \frac{1}{2} \frac{\frac{p(t_2) - p(t_1)}{t_2 - t_1} - \frac{p(t_1) - p(t_0)}{t_1 - t_0}}{t_2 - t_1} \Delta t^2 + \frac{p(t_2) - p(t_1)}{t_2 - t_1} \Delta t + p(t_2) \\
&= \frac{1}{2} \frac{\frac{\begin{pmatrix}160\\50\end{pmatrix} - \begin{pmatrix}130\\90\end{pmatrix}}{40-30} - \frac{\begin{pmatrix}130\\90\end{pmatrix} - \begin{pmatrix}110\\90\end{pmatrix}}{30-15}}{40 - 30} 5^2 + \frac{\begin{pmatrix}160\\50\end{pmatrix} - \begin{pmatrix}130\\90\end{pmatrix}}{40 - 30} 5 + \begin{pmatrix}160\\50\end{pmatrix} \\
&= 12.5 \cdot \frac{\frac{1}{10}\begin{pmatrix}30\\-40\end{pmatrix} - \frac{1}{15}\begin{pmatrix}20\\0\end{pmatrix}}{10} + \frac{1}{2}\begin{pmatrix}30\\-40\end{pmatrix} + \begin{pmatrix}160\\50\end{pmatrix} \\
&= 1.25(\begin{pmatrix}3\\-4\end{pmatrix} - \begin{pmatrix}4/3\\0\end{pmatrix}) + \begin{pmatrix}15\\-20\end{pmatrix} + \begin{pmatrix}160\\50\end{pmatrix} \\
&= \frac{5}{4}\begin{pmatrix}5/3\\-4\end{pmatrix} + \begin{pmatrix}15\\-20\end{pmatrix} + \begin{pmatrix}160\\50\end{pmatrix} \\
&= \begin{pmatrix}177 1/12\\25\end{pmatrix}
\end{aligned}$$

Assignment 3-3 *Hermite-Interpolation*

Consider the following situation: The locally assumed position of a player and his direction of movement at time t are given by dead reckoning with position vector p_{DR} and movement vector d_{DR}. At the same time the client receives an update that consists of the actual position vector p_{EX} and movement vector d_{EX} from the server.



Now, the client has to transfer position and movement which were calculated with dead reckoning to the actual data within a time window Δt. For the sake of simplicity you can assume that a player moves exactly the length of a movement vector within time window Δt. In other words, at time $t + \Delta t$ the player should be at position $p_{\text{EX}} + d_{\text{EX}}$.[2]

The following vectors are given:

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

Illustrate the idea of position correction with linear combination of Hermite-functions as described in the script (chapter 3, page 20). Calculate the value of the linear combination function $\hat{p}(x)$ (see below) for $x \in \{\frac{1}{2}, \frac{7}{8}\}$. Mark these points in the plot and sketch your idea of the corresponding connecting curve based on these.

$$h_1(x) = 2x^3 - 3x^2 + 1 \qquad h_2(x) = -2x^3 + 3x^2$$
$$h_3(x) = x^3 - 2x^2 + x \qquad h_4(x) = x^3 - x^2$$

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

where $x \in [0, 1]$ describes the progress of movement between time t and time $t + \Delta t$.[3]

$\frac{1}{2}$:

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

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

[2]Theoretically, the correction of the position could also be done by two straightforward approaches: either teleport the player from position p_{DR} to p_{EX} at time t and apply the movement along vector d_{EX}, or teleport the player from position $p_{\text{DR}} + d_{\text{DR}}$ to $p_{\text{EX}} + d_{\text{EX}}$ at time $t + \Delta t$. Both would result in undesired teleportations of the player from one position to another. To avoid this, we employ the Hermite interpolation to get a smoothed correction of the position, i.e., we use the time window Δt to smoothly move the player from position p_{DR} to position $p_{\text{EX}} + d_{\text{EX}}$ (cf. the blue line in the Figure on the next page).

[3]For instance $\hat{p}(\frac{1}{2})$ gives us the position where the player should be displayed after 50% of the time window Δt have passed (cf. the dot in the middle of the blue line in the Figure on the next page). By using various values for x, we would get various coordinates that correspond to a smooth interpolation (cf. the entire blue line).

$\frac{7}{8}$:

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

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

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