

Mobile Computing

Exercise 7

Mobile Computing 2005/2006

Administration

- Andi ist weg
 - Fragen zum Multi-Hop Layer an Andi oder Nicolas
 - Fragen zur Theorie an Pascal
 - Fragen zu neuen, praktischen Übungen an Nicolas

Mobile Computing 2005/2006

Ziel der Übung

- Multi-Hop Layer ist (fast) fertig
- Lobby
 - Spiel hosten
 - Offene Spiele suchen
 - Spiel beitreten
 - Spiel verlassen/abbrechen

Mobile Computing 2005/2006

Hilfsklassen

- Player
 - Speichert Daten zum Spieler
 - Serialisierbar
- Game
 - Speichert Daten zu einem Spiel
 - Serialisierbar
- HeartsMessage
 - Nachrichtenformat für die Lobby

Mobile Computing 2005/2006

HeartsMessage

- Auf Multi-Hop Layer aufgesetzt
 - Die ganze **HeartsMessage** ist im **data**-Feld der **MHMessage**
 - Auf **MHSocket** Ebene haben alle Pakete den Typ **TYPE_SRMSG**
- **int type**
 - Typ der **HeartsMessage**
- **Object payload**
 - Parameter abhängig vom Typ der Nachricht

Mobile Computing 2005/2006

Funktionsweise der Lobby

- Suchen von Spielen
- Spiel beitreten
- Spiel verlassen

Mobile Computing 2005/2006

Hosten/Suchen von Spielen

- Spiele werden gesucht, nicht bekannt gemacht
 - Hosten eines Spiels ist nur ein interner Zustand. Keine spontanen Nachrichten.
 - Spiele werden per Flooding gesucht.

Mobile Computing 2005/2006

Suchen von Spielen

- Flooding einer **HeartsMessage** mit
 - **type = LOOK_FOR_GAMES**
 - **payload = null**
 - Empfänger = **BROADCAST_ADDRESS**
- Ergänze MHLayer um Flooding von **TYPE_SRMSG** Nachrichten

Mobile Computing 2005/2006

Reagieren auf eine Spiel-Suche

- HeartsMessage
 - `type = HAVE_GAME`
 - `payload = Game`
 - Empfänger = Absender der Spiel-Suche
- Kein Flooding sondern direkte Nachricht
 - `MHSocket` startet Route Search

Mobile Computing 2005/2006

Spiel beitreten: Neuer Client

- HeartsMessage
 - `type = JOIN_GAME_REQ`
 - `payload = Player` (der beitreten will)
 - Empfänger = Host des Spiels
- Warten auf das `JOIN_GAME_ACK`

Mobile Computing 2005/2006

Spiel beitreten: Host

- Prüfen ob es noch Platz hat für den Spieler
- Falls Nein: Nachricht ignorieren
- Falls Ja: HeartsMessage
 - `Type = JOIN_GAME_ACK`
 - `Payload = Game`
 - Empfänger = der neue Spieler

Mobile Computing 2005/2006

Spiel beitreten: Host

- Die anderen bereits angemeldeten Spieler informieren
- HeartsMessage
 - `type = HAVE_GAME`
 - `payload = Game`
 - Empfänger = Angemeldete Spieler (einzeln)

Mobile Computing 2005/2006

Spiel verlassen: Client

- Host muss informiert werden, wenn ein Client das Spiel verlässt.
- HeartsMessage
 - **Type** = **LEAVE_GAME**
 - **Payload** = **Player**
 - Empfänger = Host des Spiels

Mobile Computing 2005/2006

Spiel verlassen: Host

- Der entsprechende Spieler wird aus dem Spiel gelöscht.
- Die angemeldeten Spieler müssen informiert werden.
- HeartsMessage
 - **type** = **HAVE_GAME**
 - **payload** = **Game**
 - Empfänger = Angemeldete Spieler (einzeln)

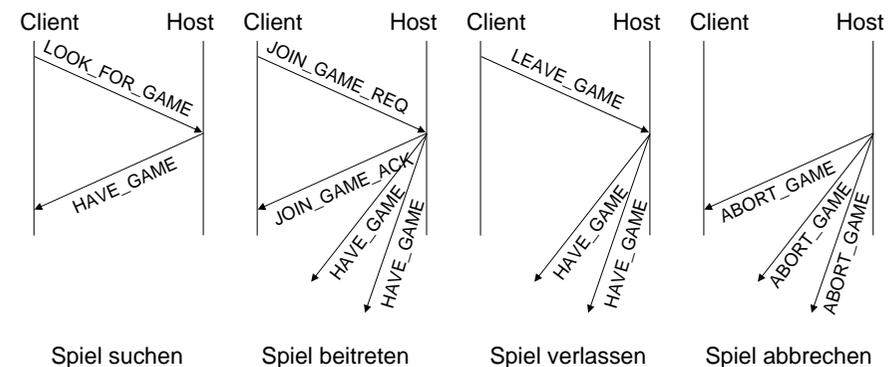
Mobile Computing 2005/2006

Host bricht ab

- Alle betroffenen Spieler müssen informiert werden, dass das Spiel aufgelöst wird
- HeartsMessage
 - **type** = **ABORT_GAME**
 - **payload** = **Game** (vor dem Abbruch)
 - Empfänger = Angemeldete Spieler (einzeln)

Mobile Computing 2005/2006

Zusammenfassung



Mobile Computing 2005/2006

User Interface

- Spieler muss sich entscheiden...
 - ein Spiel zu hosten
 - Spiele zu suchen
 - einem Spiel beizutreten
 - ein Spiel zu verlassen

Mobile Computing 2005/2006

User Interface

- Konsole
 - Pro: Einfach zu programmieren
 - Con: Benutzerunfreundlich
- Graphisch
 - Pro: Benutzerfreundlich
 - Con: Etwas schwieriger zu implementieren

Mobile Computing 2005/2006