cs121 - software development
 software architecture
 sai patterns

alexandre r.j. françois
visiting associate professor of computer science

© arjf 2010
outline

sai primitives and organizing principles

rendering
dynamic system
network receive and send
user interaction controls
music mixer and player
sai primitives & organizing principles

cell
- processing unit (no state)

repository
- shared persistent data

stream
- flow of volatile data
- process dependency
- process trigger

pulse
- active (volatile) vs. passive (persistent)
graphics rendering

**pulsar** generates empty pulses that trigger the Render cell at regular time intervals.

**render** cell creates a graphical depiction of the model held in the repository, and places the frame on the stream.

**display** cell places the input frame on the screen.
**pulsar** generates empty pulses that trigger the update cell at regular time intervals.

**update** cell computes the state of the system after time increment:

- example: \( ma = \Sigma F \)
- accel.: \( a \leftarrow g \)
- velocity: \( v \leftarrow v + \Delta t \ a \)
- position: \( p \leftarrow p + \Delta t \ v \)
network receive and send

NwReceive

network input interface

+ message

Update

game state

Can trigger other parts of the system in response to update

NwSend

network output interface

Trigger e.g. pulsar

Sample

+ message
user interaction controls

Event-based devices: mouse, keyboard, etc.

State-based devices: gamepad, sensors, etc.

Receive event
- device input
- interface
  + device event

Make command
- + command

Poll device
- device input
- interface
  + device state

Make command
- + command

Update
- game state

Update

Pulsar
- sampling rate

© arjf 2010
music mixer and player

- Trigger
- Mix
  - Samples/loops database
  - other mixing parameters and attributes
  - + audio frame
- Audio track buffer
- Update
- Pulsar
  - sampling rate
- Sample
  - + audio frame
- Play
  - Audio output interface

© arjf 2010
project architecture: a first draft...

client-server architecture
  game server
  player clients
    platformer client
    builder client
  maybe others...

game state
  description
  communication protocol for updates

other communication concerns
  chat? voice? video?
project architecture: servers and clients

Game server
- maintains game consistency
- computes physics
- enforces game rules
- manages communications with clients

Player client
- maintains local game state (from server)
- displays game state (graphics and sound effects)
- handles player input
- dispatches commands to game server
- Creates contextual music and sound effects

Spectator client
- maintains local game state (from server)
- displays game state (graphics and sound effects)
- handles spectator input
- creates contextual music and sound effects
modules

graphics
  spectator / platformer / builder
physics and collision detection
controls
  spectator / platformer/ builder - mouse / keyboard
game logic
game AI
music & sound
networking
player client
architecture draft – conceptual level

Server Update

Graphics

+ Music&sound

Control

NwReceiveFromServer

network input interface(s)

+ message

N-update

Local game state

Rendering camera and parameters

Pulsar

frame rate

+ empty pulse

Render

Updates rendering parameters only

MakeCommand

device input interface

+ device event

MakeMessage

+ command

+ message

C-update

Server closes the game loop!

NwSendToServer

network output interface(s)
music&sound engine
architecture draft – conceptual level

Local game state

Pulsar
sampling rate

PollGameState

Mix
+ audio frame

Update
Samples/loops
database
other mixing
parameters
and attributes

Sample
+ audio frame

Play
Audio output
interface

Audio track
buffer

Alternate trigger?
Communications

NwReceiveFromClients

network input interface(s)

+ message

N-update

N-sample

Game state

Render

network output interface(s)

Is this necessarily synchronized?

Pulsar

Δt

+ empty pulse

G-update

Game AI

Trigger?

Communications

NwSendToClients

Physics and collision detection

G-update

Game logic

PC-update

Graphics (debugging)

Pulsar

frame rate

+ empty pulse

Communications

NwSendToClients

network output interface(s)

Game state

Rendering camera and parameters

Communications

NwSendToClients

network output interface(s)