

Development of a 3-D Multiplayer Racing Game

JOGI, JOAL and Java .NIO for Game Development

Evangelos Pournaras
University of Surrey
1041

AGENDA

- > JautOGL Racing Game
- > Supported Functions
- > Architecture
- > Elements of the Game Engine
- > Future Work Extensions
- > Research and P2P Networking Models

AGENDA

- > **JautOGL Racing Game**
- > Supported Functions
- > Architecture
- > Elements of the Game Engine
- > Future Work Extensions
- > Research and P2P Networking Models

Why a Racing Game?

- > A very popular category of games
- > A challenging application for programmer
- > Trade off issues between efficiency and visual experience
- > Real-time requirements
 - In networking
 - In game playing (actions) and interactivity
- > Multidimensional application
 - Graphics
 - Sound
 - Artificial Intelligence
 - Networking
 - Advanced interactions
 - Advanced physics

Why Java?

- > Java is quite fast as the dominant C++ in game industry
- > Both high and low level programming solutions
- > Application accessibility through *Java Web Start* and *install4j*
- > Existing commercial games
- > Sun Microsystems strongly supports the game orientation of Java
- > Foundations are set in game consoles industries for Java support
 - Networking services orientation
 - Phantom console (<http://www.phantom.net/>)
 - Sony Playstation 3 can provide java support through some Linux versions
- > More steps are required but the most promising fact is the networking orientation of the game services which customers tend to follow

JautOGL Racing Game

- > Started as my undergraduate dissertation in University of Piraeus, Greece (
<http://www.unipi.gr>)
- > Part of the work published on java.net (
<http://today.java.net/pub/a/today/2006/10/10/development-of-3d-multiplayer-racing-game.html>)
- > Project under-development on java.net (
<http://jautogl.dev.java.net>)
- > It needs further work



AGENDA

- > JautOGL Racing Game
- > **Supported Functions**
- > Architecture
- > Elements of the Game Engine
- > Future Work Extensions
- > Research and P2P Networking Models

Game Engine Elements

- > Initial objective
 - Development of some fundamental game elements
 - A running demo version of the game
- > Supported functions
 - Screen management (Swing and Full Screen Support)
 - Keyboard interactivity
 - 3-D Models Loading
 - Simple 3-D sound playing
 - Multiplayer networking mode
 - Simple game playing (Car movement, camera change)

AGENDA

- > JautOGL Racing Game
- > Supported Functions
- > **Architecture**
- > Elements of the Game Engine
- > Future Work Extensions
- > Research and P2P Networking Models

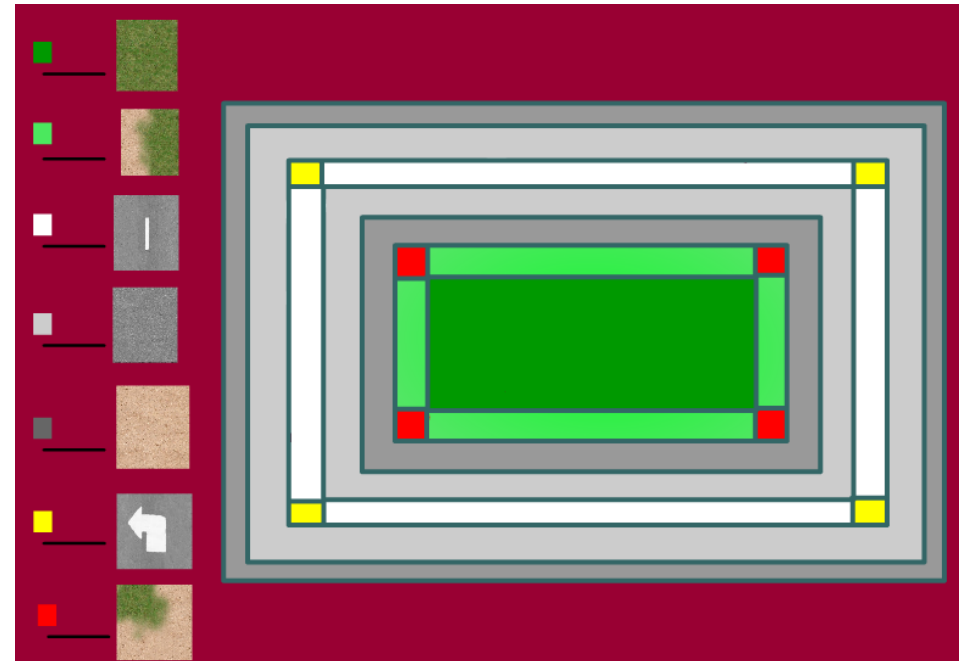
- [illegible]

AGENDA

- > JautOGL Racing Game
- > Supported Functions
- > Architecture
- > **Elements of the Game Engine**
- > Future Work Extensions
- > Research and P2P Networking Models

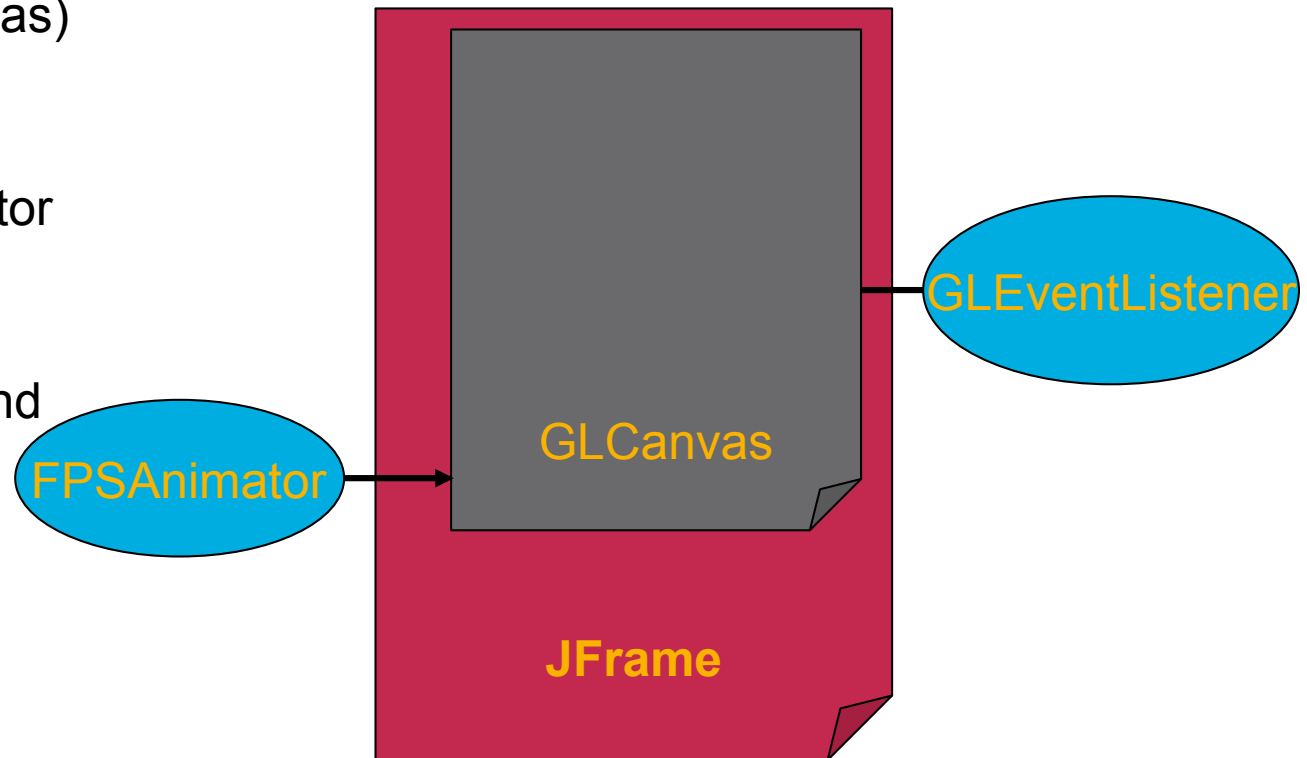
3-D Graphics

- > JOGL API
- > 3-D graphics hardware acceleration
- > Draw a simple racing terrain
 - Triangle strip, mipmapping
- > Texture support
- > Set the lighting in the scene
- > Load the 3-D car models
- > Camera changes
 - DRIVER_MODE, FAR_MODE, NORMAL_MODE
 - Modification of visibility range and properties
(`GLU.gluPerspective`, `GL.glFrustum`)



Screen Management

- > Full Screen Mode
- > Join OpenGL (GLCanvas) with the Java Swing (JFrame)
- > Animation / FPSAnimator
- > Define and choose the best display mode (resolution, bit depth and refresh rate)
- > Anti-flickering
- > Anti-tearing



3-D Models Loading

> Use of Wavefront format specification

- .obj file format

`v float float float -> gl.glVertex3f(x,y,z);`

- A vertex in 3-D space

`vn float float float -> gl.glNormal3f(x, y, z);`

- A vertex normal in 3-D space

`vt float float -> gl.glTexCoord3f(x,y,z);`

- **A texture vertex**

`f int int int ...`

`f int/int int/int int/int . . .`

`f int/int/int int/int/int int/int/int ...`

- A polygonal face. The numbers refer to vertex positions, texture coordinates and normals respectively

3-D Models Loading (cont.)

- Also more complex geometrical shapes and elements can be supported (Bezier curves, B-Splines, Taylor polynomial curves and complex surfaces)

- .mtl file format

`newmtl name`

- New material definition (used by `usemtl` in .obj file)

`Ka r g b`

- Ambient RGB color float values

`Kd r g b`

- Diffuse RGB color float values

`Ks r g b`

- Specular RGB color float values

`d alpha` or `Tr alpha`

- The alpha value specifying the material transparency

- `gl.glColor4f(r, g, b, alpha);`

3-D Models Loading (cont.)

`illum n`

- Illumination mode of the material

`map_Ka filename`

- The texture for the material

- > Wavefront format 3-D models are produced by 3-D Studio Max 8 (and above) and Blender
- > Development of Java .obj and .mtl parsers
 - Use of the basics of the specification
 - Information is stored in array lists (`ArrayList`)
 - `StringTokenizer`
 - `trim()`
 - `startsWith()`
 - `charAt()`

3-D Models Loading (cont.)

- > How the information is stored in 3-D scene?
 - From array lists to OpenGL pipeline
 - Display lists (`glGenLists()`, `glNewList()`)
- > Alternative OpenGL storage of information: Vertex arrays
- > Display Lists vs Vertex Arrays
 - Display lists are compiled in VRAM. CPU resources are not consumed during display lists calls. It is a good solution for mostly static data in the 3-D environment
 - Vertex arrays benefit from DMA memory transfers and low degree of CPU calls. A good solution for dynamic data which change rapidly

Interactivity

- > Keyboard/mouse interactions for game playing
- > Interactivity patterns
 - User could configure the buttons and perform the keyboard mapping
- > Interactions for:
 - Car navigation
 - Stopping the game
 - Change camera view
- > Detections of different button press profiles (e.g. initial press)
- > Player's interactions provide game feedback (sounds, and graphics transformations)
 - Sounds for acceleration and ready-to-go
 - Calculations of the new cars coordinates

3-D Sound

- > JOAL API
- > OpenAL entities:
 - **Buffer:** Loads sounds, requires data streaming due to limited JVM default buffer size
 - **Source:** The position of a sound source in the 3-D space (cars, environment)
 - **Listener:** Create an association between the camera position and listener position (mainly the driver - player). This can provide the illusion of the distances and the different hearing ranges
- > JautOGL sound engine uses only ambient and spot sounds
- > The sound engine runs in its own thread
- > Manage loading, sound attributes and exiting issues of sound engine

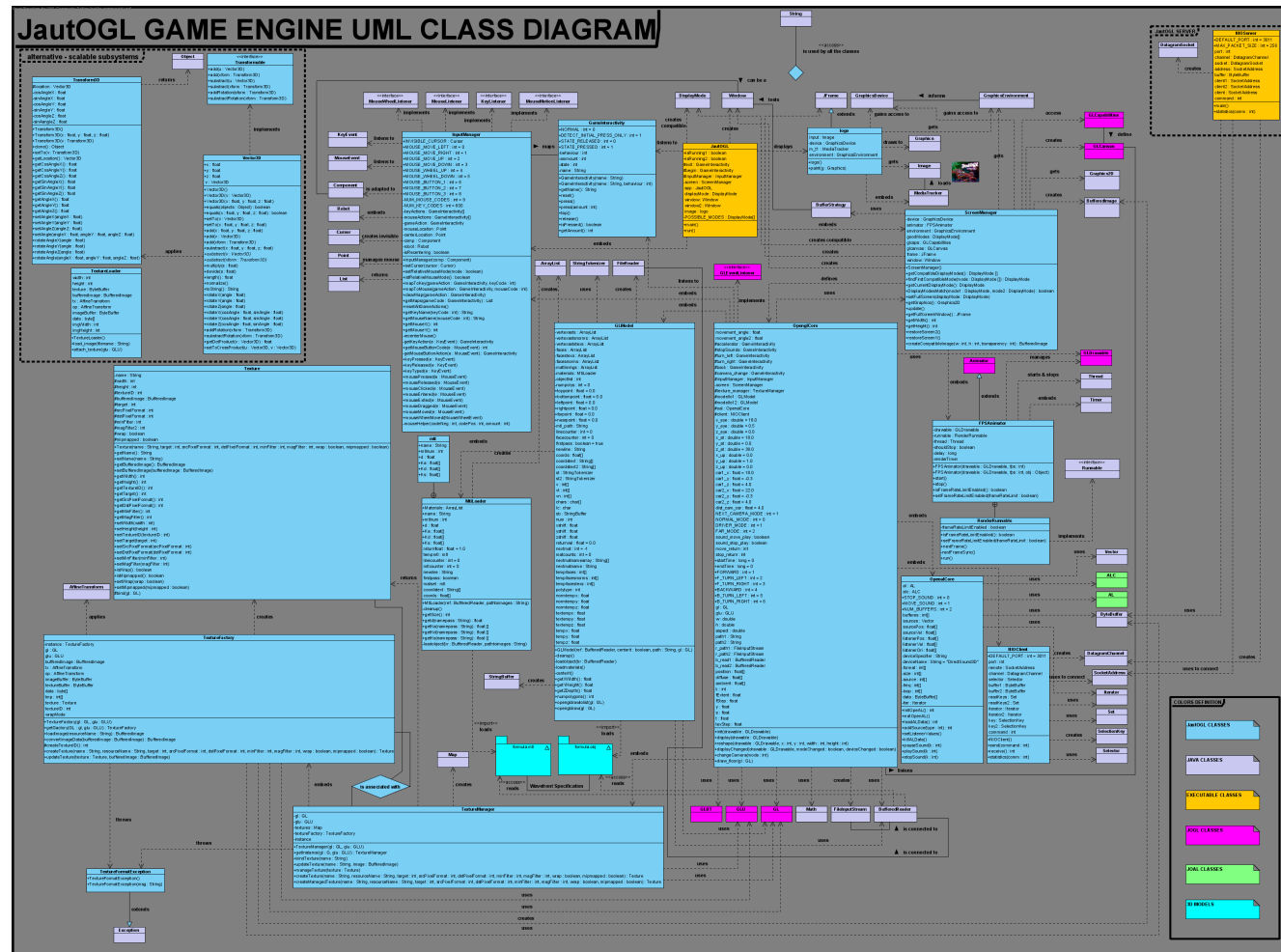
Multiplayer mode and Networking

- > Client – server model
- > UDP protocol. Why?
 - Racing game requires fast data transfers
 - Reliability can be easily satisfied in application level
 - It is a widely used protocol for real-time applications
- > Many networking data sources must be handled in the game
 - Many threads
 - Synchronization
 - Complication
 - Instability
 - Overhead
 - **Problems!**

Multiplayer mode and Networking (cont.)

- > Java .NIO. It provides:
 - Non-blocking communication
 - Reading and writing in one thread
 - Communication is established over channels
- > JautOGL networking model
 - At the beginning (logging in the game) some simple handshake datagrams are exchanged
 - Server knows the players
 - Each client receives data (opponent's car coordinates)
 - At the same time it transmits its own state

Game Engine Overview



AGENDA

- > JautOGL Racing Game
- > Supported Functions
- > Architecture
- > Elements of the Game Engine
- > **Future Work Extensions**
- > Research and P2P Networking Models

Future Work Extensions

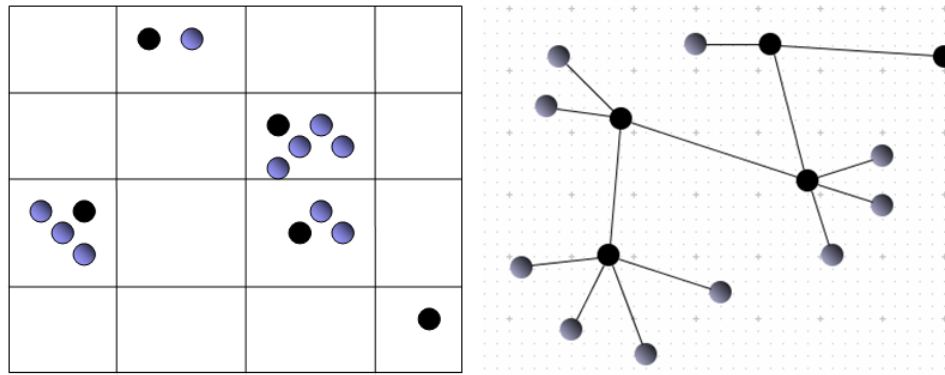
- > A more advanced sound engine
 - Cone sound
 - Doppler effect simulation
- > Collision detection system
- > Artificial Intelligence
 - Neural Networks for creating automatic driven opponents
- > Interactivity
 - JInput
- > GUI
- > Physics
- > Networking
 - A more advanced networking protocol
 - Manages multiple .nio channels

AGENDA

- > JautOGL Racing Game
- > Supported Functions
- > Architecture
- > Elements of the Game Engine
- > Future Work Extensions
- > **Research and P2P Networking Models**

Research and P2P Networking Models

- > There are diverse research dimensions in this field
- > This work inspired me on a P2P networking orientation
- > My potential PhD
- > Virtual regions are mapped to a P2P network topology (Gnutella 2 Network)
- > In such a network there is a growing complexity
- > Self-organized and self-management issues are raised
- > Autonomic P2P computing



Evangelos Pournaras
University of Surrey

ev.pournaras@yahoo.com