STE6274
Game Design

By Pål Solberg Trefall

Introduction

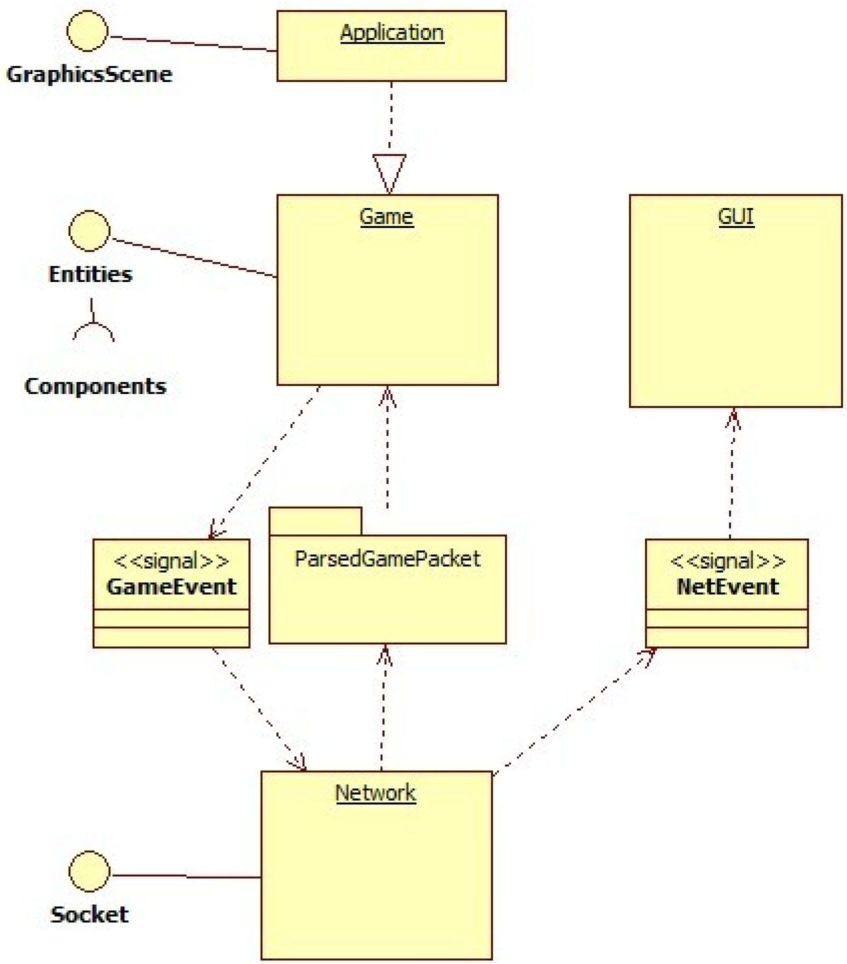
- Approach to the assignment
  - Architecture
    - CMake project definition
    - Network, Gui and Game layer separation
    - Component-Oriented Programming of Entities
  - C++
    - Use of C++0x/C++11 features
    - Templates, Inline, Preprocessor
  - Graphics / OpenGL
  - Client responsibilities

Architecture

- CMake
  - Define project structure and set up build options
  - Find and bind third-party dependencies
  - Copy third-party dlls into execution folder
  - Build Qt types (MOC and UIC)

Architecture

- Network, GUI and Game layer separation



Architecture

- Component-oriented programming
  - Totem Entity Development Kit
    - Shared Properties emiting signals on value-changed
    - Hashed strings for optimized string lookup
    - Template-based event system using signal/slots
    - Delegates
    - Serialization
    - Doxygen documentation
    - Components
    - No third-party dependencies

C++

- C++0x / C++11
  - SharedPtr
  - Unordered Map
  - Hash_value
- Templates
  - Typedef problem
- Inline
  - Optimizing compiled code
- Preprocessor
  - Simplify configurable code
  - Simplify copy/paste-oriented sections

Graphics / OpenGL

- RenderSystem
- Renderable (component)
- MeshSystem
- MeshGeometry (component)
- MaterialSystem
- Material (component)
- SkyboxGeometry (component)
- ParticleSystem
- EngineFlameParticleEngine (component)

Client responsibility

- Client dumbness
- Respect the protocol
- Input handling for games

Conclusion

- Qt and multithreading
- Thoughts on assignment
- Project result

DEMO