

Rethinking Game Architecture with Immutability – Milestone 4

(February 19th)

Student: Jacob Dufault (jacobdufault@gmail.com)

Faculty sponsor: Professor Bernhard

Progress Matrix

Task	Completion
Change project focus	100%
Update milestone goals	100%
Continue work on entity framework	100%
Create spatial module	100%
Separate Unity integration into a custom library	100%

Task Summaries

Change project focus

Forge is a more academically interesting project than creating a new game. It contains numerous innovative features with respect to game engines, such as automated multithreading. It has been designed to support programmer productivity above all else, while still having good performance.

The previous project, a tower defense game, was interesting but as the senior design project has progressed much of the work has gone into Forge. It makes sense to switch on numerous accounts.

Update milestone goals

The milestone goals have been updated to reflect the new project direction.

Milestone 5

- Create poster
- Implementation automatic synchronization error detection
 - This will detect “de-syncs”, or when two computers have diverged into two different game states. This will be implemented primarily by comparing hashes of the current game state. This technique is relatively fast and widely used in the industry.
- Continue work on Unity integration
 - Forge is an engine independent entity component system framework. It needs to be associated with a rendering engine to be useful. For this senior design project, there will be two bindings: Unity and XNA. After milestone 5, the Unity bindings will be completed.
- Begin work on XNA/MonoGame integration
 - Work will begin on the XNA and MonoGame bindings. These are not going to be as sophisticated as the Unity bindings as they will not provide a content creation experience. The XNA/MonoGame bindings only need to provide runtime support, such as rendering.

Milestone 6

- Work on XNA/MonoGame bindings
 - The XNA/MonoGame bindings will need to be finished by the end of this milestone, as the demo game depends on them.
- Demo a simple game running in both XNA and Unity
 - One of the key features of Forge is engine dependence. The game will not be terribly complex, but it will demonstrate how to use Forge, serving a critical part of the documentation. Current thoughts are towards a simple platformer game, though this may change depending on time constraints. A cool demo would be having the XNA client and the Unity client connect to each other and play multiplayer together, though again, this depends on time constraints and is not critical for the project.
- Write user manual and documentation
 - Forge is a complex system with many moving parts. It solves many issues in game development and can be intimidating. Tutorials and overall documentation are required. The demo game can serve as an advanced tutorial, but more documentation will need to be written.

Continue work on entity framework

There were a number of bug fixes within Forge that landed in the past month. A few breaking public API changes occurred, mainly dealing with deserialization error diagnostics.

Create spatial module

Work has begun on a spatial module that provides optimization spatial querying and monitoring. This is implemented via a quadtree. Additionally, the spatial module will include support for locomotion of predefined paths and velocity based movement.

Separate Unity integration into a separate library

The Unity related code was completely separated into a separate repository. This helps increase code modularity as accidental dependencies cannot be introduced.

Sponsor Feedback

Signature and Date: _____

Feedback: