**INFO6044: Game Engine Frameworks & Patterns**

**Detailed content:**

|  |  |  |
| --- | --- | --- |
| Week 1: | C / C++ basic concepts: strong types, unmanaged memory (stack vs. heap), pointers (arrays), basics of standard library (string, string stream) and the STL (vector, map, list), file I/O |  |
| Week 2, 3: | C++ classes: basics, inheritance, polymorphism, interfaces, overloading, “pimpl” pattern |  |
| Week 4: | Abstract factory design pattern | Project #1 due |
| Week 5-6: | Pointers and handles, abstract factory/manager, mediator pattern |  |
| Week 6-7: | Inheritance vs. aggregation (“is-a” vs “contains”), data designed (array of structures vs. structure of arrays) |  |
| Week 8: |  | Mid-term, Project #2 due |
| Week 9: | Data persistence (load, save, etc.) |  |
| Week 10-11: | Timing and control (way points, linear speed, ease in/out, acceleration curves, key frame animation & transitions) |  |
| Week 12: | Hierarchical animation systems (aka “scene graphs”) for rigid bodies (i.e. *not* skinned mesh) | Project #3 due, |
| Week 13: | Basic animation and control scripting |  |
| Week 14: | Additional topics | Project #4 due (week 14) |
| Exam Week |  | Final Exam |

**Major Project Descriptions**

Production of a number of (approximately 2 to 4) example applications of application components. A possible project sequence would be:

Project #1 (Approx. Week 4): Basic implementation of game engine/animation framework

Project #2 (Approx. Week 6): Enhancement of game engine/animation framework

Project #3 (Approx. Week 11): Addition of timing and animation controls (way-points, curves, etc.)

Project #4 (Approx. Week 14): Addition of basic “animation and control” scripting

**Teaching Methodologies:**

Lecture: 30%

Project: 40%

Individual and/or group projects: 30%

**Method Of Evaluation:**

The final mark/grade for this course will be determined as follows:

* Projects and Assignments: 60%
* Exams and tests: 40%

Exact date of each test will be announced during the first class of the semester.

*This course is* ***NOT*** *upgradeable under college policy 2-C-04.*

**Suggested Learning Resources:**

Jason Gregory, Game Engine Architecture, A.K. Peters Ltd., 2009, ISBN-10:1568814135, ISBN-13: 978-1568814131, or

Jason Gregory, Game Engine Architecture, Second Edition, A.K. Peters Ltd., 2014, ISBN-10: 1466560010, ISBN-13: 978-1466560017

Erich Gamma, et al, Design Patterns: Elements of Reusable Object-Oriented Software Hardcover, Addison-Wesley Professional, 1994, ISBN-10: 0201633612, ISBN-13: 978-0201633610