

Vision Document

Dual State Framework - Software
Development Project 2014 - IT Carlow



Introduction	3
<i>What is Parallelism?</i>	3
<i>Why, When, and How?</i>	3
Requirement	3
<i>Skills Required</i>	3
<i>Environment</i>	3
Stakeholders and User Description	4
<i>Stakeholder Summary</i>	4
<i>User Summary</i>	4
Main Risk	4
<i>Business Risk</i>	4
<i>Technology Risk</i>	4
Main Functionalities	4
<i>Simulation Based</i>	4
<i>High Level</i>	4
<i>Predictable</i>	5
<i>No Deadlock or Livelock</i>	5
Reference	5

Introduction

The project is going to create a framework which implements parallel computing. It helps game developers be easier to parallelize.

What is Parallelism?

"Parallel computing is the science and art of programming computers that can do more than one operation at once, concurrently, during the same cycle, often via having more than one processor." [Parallel Computing]

Why, When, and How?

"Why?"

- *Problems "too" costly to be solved with the classical approach*
- *The need of results on specific (or reasonable) time*

When?

- *Are there any sequences which are better suited for parallel implementation*
- *Are there situations when is better NOT to parallelize*

How?

- *Which are the constraints (if any) when we need to pass from sequential to parallel implementation*
- Requirement" [Parallel Processing why when how?]*

Requirement

Skills Required

- C++ static and dynamic libraries
- Parallel Programming Mode
- OpenMP & Intel Threading Building Blocks

Environment

Operating System	Mac OS X
Compiler	Apple LLVM
Programming Languages	C++
IDE	Xcode
Libraries Required	STL, Intel Threading Building Blocks

Stakeholders and User Description

Stakeholder Summary

Name	Nominations	Responsibilities
Software Develop	Yu Chen	Develop the DualStateFramework library
Supervisor	Joseph Kehoe	Contact to customers

User Summary

Name	Discription
Game Developers	Who wants to create game with parallelism

Main Risk

Business Risk

- A lot of game development frameworks are already existed.
- Value proposition has not been validated yet.

Technology Risk

- Memory allocation
- Use of pointer
- Memory leak
- OpenMP

Main Functionalities

“

Simulation Based

The model should map naturally onto the object based paradigm used in simulations. The closer the fit the easier it will be to incorporate into the games domain.

High Level

The model should not involve the user in the low level details of the implementation of concurrency. It should also be easily incorporated into existing high level scripting languages.

Predictable

The model should, as far as possible, allow for predictable run times so that guarantees can be given as to the amount of time required to process the behavior generating scripts during each frame.

No Deadlock or Livelock

The model should remove deadlock and livelock. Deadlock is where a system locks up because two or more processes are waiting for each other to finish. A livelock is similar to a deadlock, except that the states of the processes involved in the livelock constantly change with regard to one another, none progressing. Reasoning about deadlock and livelock requires a high level of programming skill that scriptures cannot be assumed to have. The user of the model, the scripter, should not have to worry about these issues.

“[A Concurrency Model for Game Scripting]

Reference

http://www5.in.tum.de/forschung/simlab/course2010_files/course_mat/SimLab10Potolea.pdf
[Parallel Processing why when how?]

http://www.dmoz.org/desc/Computers/Parallel_Computing [Parallel Computing]

Kehoe. J. “A Concurrency Model for Game Scripting” [A Concurrency Model for Game Scripting]