NEWCASTLE UNIVERSITY

Implementing Finite State Machines on the Graphics Processing Unit

Andrew Jones (110305163)

Master Computer Science, May 2014

Supervisor: Dr William Blewitt

Word Count XXXXX

**Abstract.**

Declaration

Acknowledgement

Contents

Introduction

Terminology

Project Outline

Aims and Objectives

Aims

Objectives

Changes

Research

Research Strategy

AI within games (finite state machines)

GPGPU

Developing a solution

Planning

Use of old code

Designing the core system

Stuff here

Tools and technologies used

Developing the System

CPU

GPU

Optimisations

Testing

Before and after optimisations

Set test for CPU and GPU

Max test for CPU and GPU

Evaluation

Research and specification

Design

Implementation

Testing and results

Quality of tests

Data collected

Conclusion

Aims and objectives

What has be learnt

Further work

Summery

References

1. Introduction
   1. Terminology
   2. Project Outline

Artificial intelligence is a key element in the majority of modern video games and there has been many different AI systems researched and developed specifically for use within video games. These systems range from pathfinding to non-deterministic decision making and are very effective at what they do. Currently the vast majority of these systems are designed for use on a single CPU with very little parallelism in mind. Even though AI plays such an important role in these games it is slowly receiving less and less of the CPU’s processing time [1] to allow for more to be spent on graphics and physics processing. With these two limitations, AI is not advancing at the same rate within games as other aspects. Some games may require a large quantity of intelligent agents and with the current state of games AI this would not be possible.

If these systems could be made to be run in parallel, they would be able to be run on the GPU and it would make them much more scalable. The GPU has a parallel architecture and utilizing this would give an enormous performance increase, allowing for a large number of agents to be run at once. Over the years, there has be an ever growing interest in moving parts of video games onto the GPU but the main areas of research have been into implementing physics, complex mathematics and search algorithms on the GPU [2]. In comparison there has been very little research into implementing AI decision making onto the GPU.

Massively Multiplayer Online Role Playing Games (MMORPG) have a massive number of agents within them, with many needing to be run at the same time. These agents generally run simple Finite State Machines (FSM) and rarely need communication between agents which makes them ideal candidates to try and implement onto the GPU. Shared memory is limited on the GPU and accessing shared memory can lead to less parallelism if used incorrectly so the lack of communication helps. Even though there is no communication there are still problems that may arise with just a simple implementation.

In this project, we will attempt to implement a modern day MMORPG agent on the GPU, see if it would give any performance improvements and what techniques could be used to give even further performance improvements. Firstly we will look at older and current attempts to implement FSMs on the GPU as well as what modern day MMORPGs AI systems involve. Then we look at the development of an AI system on the CPU being converted to a GPU system and what possible improvements can be made. Lastly we look at and compare the results of these different versions of the GPU implementation compared the CPU implementation and in which situations they are better or worse.

* 1. Aims and Objectives

1. Research

This section looks at an overview of modern AI within games as well as current programming on the GPU before specifically looking at previous attempts at implementing FSMs on the GPU.

* 1. AI Within Games

There are many different aspects to games and many different goals that games are created for however; the main aim of a game should be the give the player an enjoyable and fun experience. If there is a feature that takes a lot of resource but doesn’t increase this factor it will normally not be implemented or removed from the games to make room for elements that satisfy this factor more. This constraint is the main cause for the divide between research AI and game AI because game AI does not necessarily solve very complex tasks and needs to solve its problems in real time while taking as little resources as possible. AI still plays a key role in video games, however it has been placed on a lower priority for resources as realistic graphics and physics can usually give a bigger increase to player enjoyment in most cases. Although this is not the case for all games where very simple, or “stupid”, AI would ruin the player’s immersion. Using the GPU may allow for very complex AI systems to be implemented onto the GPU and still run in real time without hindering other areas of the game and in some cases could potentially allow completely new types of games to be created.

Another direction that GPGPU could take AI is instead of having more complex AI, is to have a massive number of simple AI all running in parallel. MMORPG need to have a lot of rather simple entities running at all time and sampling data from a high number of players and if there was a strong GPGPU implementation this could potentially vastly improve the number of agents they could use. One of the foundations of game AI is the FSM which is a power yet simple took for giving an agent in a game the illusion of artificially intelligence [3]. A classic example of a game that uses FSMs is Pac-man, in which each one of the ghosts has its own FSM to govern its plan on how to beat the player and where to move to accomplish this [4]. This simple yet efficient technique can be modified to emulate a variety of different agent types for use in a variety of games. As well as this, more complex AIs can be created using modified versions of this simple AI system such as hierarchical finite state machines or nondeterministic variants. These form a foundation for a high quantity of AI models in video games.

The AI for MMORPGs is normally some form of FSM that is ran on the server side thus it does not need to render anything at all which the GPU would normally do. If the AI could be implemented onto the GPU this would allow many more agents per server or even potentially more servers to be hosted as well as freeing up CPU processing power for other aspects of the game. This potentially could allow more room for increasing the players experience within the game.

* 1. Programming on the GPU

References

[1] Anderson, E. F. (2003). Playing Smart - Artificial Intelligence in Computer Games.

[2] Carneiro, T. (2001). A New Parallel Schema for Branch-and-Bound Algorithms Using GPGPU . 23rd International Symposium on Computer Architecture and High Performance Computing.

[3]

[4] Thompson, T. (2008). An Evaluation of the Benefits of Look-Ahead in Pac-Man. IEEE Symposium on Computational Intelligence and Games.