

Interactive Crowd Simulation guided by a Kinect Sensor User Interface

Grant Adam Mulitz-Schimmel

Lu Yang

Rafael Farias Marinheiro

Sam Luo Nelson

Tae Hoon Kim

Cornell University*



Figure 1: *World War Z (2013). Zombie agents were computationally simulated.*

Abstract

We intend to create a interactive Crowd Simulation application using the approach described in [Kim et al. 2012]. The user will be able to interact with the application with a commodity depth sensor [Zhang 2012], modifying the environment in real time.

CR Categories: I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent systems; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: crowd simulation, user interaction, real-time rendering

1 Overview

When trying to create highly detailed virtual environments, such as those present in movies and games, it is necessary to add extra elements to the scene to enrich the user experience. One important element that is almost always present in modern virtual environments is the crowd: a set of agents that are part of the scene and that must behave according to the environment.

In interactive applications, such as in games, the crowd must behave

according to the user’s actions. Therefore, the application must simulate the behavior of those agents in runtime. In non-interactive applications, such as in movies, the artists have the freedom to create the scene beforehand. However, when trying to create large scale scenes, it is impractical to use artists to model every agent, so the director may choose to use computational methods to create the scene (see Figure 1).

2 Software Deliverable

We will deliver a interactive Crowd Simulation application. The user will be able to interact with the agents of the Crowd by using a Kinect Sensor. He will be able to assign orders to the crowd and also to modify the environment (by throwing fireballs, modifying the terrain, adding obstacles and so on).

3 Techniques

In order to implement the Crowd Simulation, we will use the Crowd Simulation method described in [Kim et al. 2012]. This method allows the user to add special elements called Stressors, that affect directly the way that agents choose their path.

We intend to use Microsoft Kinect Sensor to allow the user to interact directly with the crowd. To add realism to the scene, we intend to use modern Real-Time rendering technique such as Deferred Shading [Hargreaves and Harris 2004], Shadow Mapping [Stamminger and Drettakis 2002] and etc.

The software will be implemented using C++ and OpenGL.

4 Software Architecture

We will split this project in 3 different modules.

4.1 Crowd Simulation

This module is responsible for controlling the behavior of the agents. Grant and Lu will work in this module.

4.2 Rendering

This module is responsible for rendering the virtual environment to the user. Rafael and Sam will work in this module.

4.3 User Interaction

This module is responsible for managing the input data that comes from the Kinect Sensor. Tae will work in this module.

5 Milestones

We intend to complete each module separately by the first Milestone (November 14th). After that, we will start to integrate all theses modules together into a single application, which will be presented by the end of the course.

*{gam244, ly77, rf356, sln58, tk536}@cornell.edu

References

- HARGREAVES, S., AND HARRIS, M. 2004. Deferred shading. In *Game Developers Conference*, vol. 2.
- KIM, S., GUY, S. J., MANOCHA, D., AND LIN, M. C. 2012. Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 55–62.
- STAMMINGER, M., AND DRETTAKIS, G. 2002. Perspective shadow maps. *ACM Trans. Graph.* 21, 3 (July), 557–562.
- ZHANG, Z. 2012. Microsoft kinect sensor and its effect. *IEEE MultiMedia* 19, 2 (Apr.), 4–10.