Selection in 3D Graphics Environments - Final Report

CREU Final Reports 2015

Student Researchers: Christine Albert
                      Lindsey Press

Advisor: Dr. Mirela Damian

Institution: Villanova University

Webpage: http://www.csc.villanova.edu/~creu/selection

Goals of the Project

The level of detail portrayed in graphics environments is steadily increasing and so is the challenge to enable users to accurately select 3D objects from a graphics scene in real time. Thousands of players may compete against one another in a multi-player game for instance and they frequently select objects by pointing to them with the mouse cursor or other pointing devices. Selection in a 3D scene is addressed by picking algorithms, which are the subject of this research.

The problem we address is as follows: in a 3D graphics scene displayed on the screen, which object is under the mouse? By using two different picking algorithms and comparing them with regard to speed and accuracy, this research provides an empirical answer to determine the faster of the algorithms to find which object is under the mouse. Knowledge of the faster algorithm contributes to the field of graphics by allowing video games design to use the faster and more accurate algorithm to increase efficiency and user game enjoyment.

The focus in this project is on analysis and implementation rather than development of new picking algorithms. The work is motivated by the lack of literature on general implementation of picking algorithms not tied to the graphics pipeline of a particular application programming interface (API), such as OpenGL.

Process

Our approach to this research entails studying the inverse of the graphics pipeline transformations and the way the GPU converts 3D scenes into 2D images on the screen. To determine which object in a 3D scene is under the mouse, we investigated two different methods: (1) ray casting, designed to run on the CPU, and (2) color picking, designed to run on the GPU. We computationally analyzed these algorithms and compared them across several levels of scene complexity and object velocity to determine the conditions under which one picking algorithm outperforms the other.

When a 3D object is rendered, it undergoes a series of transformations. The pipeline traverses from object space to world space to camera space to viewport space, with each of these transformations involving a series of matrix manipulations. The ray picking approach utilizes the inverse of this pipeline so that when the mouse clicks in the viewport space, reverse transformations are applied to discover the location of the ray in object space. Once in object space, the object under the mouse can be determined by testing for intersection.
The color picking approach assigns a unique color to each object to identify which object has been selected. The screen is rendered twice, once on screen using normal colors, and once off-screen (in a hidden frame buffer) using unique colors based on a unique object identifier. When the user performs a selection, the pixel at the mouse position is read back from the hidden frame buffer. The object with the color matching the pixel color is selected and the scene is rendered directly on the screen with the selected object highlighted. Color picking does not require any intersection tests; however it does not provide any feedback on the exact point of intersection.

**Conclusions and Results**

We implemented the two algorithms (ray picking and color picking) and collected timing data for 100 selection test runs per screen image. We varied the scene complexity between 100 and 1500 different objects per scene. We recorded the CPU and GPU times on three different machines.

Our experimental results indicate that the GPU color picking algorithm is consistently faster and slightly more accurate than the CPU ray picking algorithm. The GPU performance is independent of the number of objects in the input scene; we attribute this to the large number of GPU units in operation. The CPU performance is highly dependent on the number of objects; we attribute this to the overhead introduced by the intersection tests, which is linear in the number of objects.

Although the two algorithms are similar in terms of accuracy, the color picking method seems slightly more accurate than the ray picking method, especially for distant objects. We suspect that this is mainly due to the floating-point errors induced by the graphics pipeline operations employed by the ray picking algorithm.

In our future work we plan to implement the ray picking algorithm on the GPU, to take advantage of the many parallel GPU units. Each GPU unit could potentially execute a different intersection test between the selection ray and a scene object. We conjecture that in this case the two picking algorithm would be more balanced in terms of selection time.

**Presentations and Publications**

Christine Albert and Lindsey Press. "Selection in 3D Graphics Environments." Research poster, ACM Student Research Competition, SIGCSE, Kansas City, Missouri, March 2015. This poster was awarded 3rd place in the undergraduate research competition.

Christine Albert and Lindsey Press. "Selection in 3D Graphics Environments." Research poster, Villanova Chapter of Sigma Xi Student Research Competition, Villanova University, April 2015. This poster was awarded 1st place in the undergraduate computer science research competition.