

NUNY: Ninja University in the City of New York

Kshitij Bhardwaj, Van Bui, Vinti, and Kuangya Zhai
{kb2673, vb2363, vv2236, kz2219}@columbia.edu

Overview

In this project, we will design and implement a Fruit Ninja like video game on the FPGA. Fruit Ninja is a popular video game where the player slices fruit using their finger(s) on the touch screen. The theme of our game will be based on undergraduate/graduate school life so that rather than slicing fruit, the object of the game will be to slice assignments, exams, qualifiers, advisor meetings, paper deadlines, thesis writing, food (like pizza), coffee, books, etc. The game will generate several moving objects on the screen and the goal is to destroy as many objects as possible using an on screen blade or sword controlled by either the PS2 mouse and/or hand motion captured by a video camera.

Description

The goal of the game is to destroy as many objects as possible under certain constraints such as time and number of lives remaining. When the moving track of the player's hand or the PS2 mouse intersects with an object, the object will be either partially damaged or completely destroyed. Some objects will require several slices to be completely destroyed. An object may appear on the screen multiple times to allow the player several chances to completely destroy it.

The player can chose the difficulty of the game based on the number of classes and desired degree (bachelors, masters, or doctorate). In the hard mode, there will be a large number of objects moving and disappearing quickly on the screen. The objects will move randomly, and possibly rotate to present a better visual effect. The moving and changing graphics on the screen are expected to be very computation intensive. Since the behaviors of different objects are independent, we will leverage the parallelism of the FPGA to visualize the moving objects.

Algorithms

The trajectory of moving, falling, and splitting objects on the screen will be based on game physics. We will compute physical properties such as linear/angular velocity and linear impulse.

The algorithm that we plan to implement for hand recognition will be color-based human skin pixel detection. For hand-tracking, the algorithm will be based on distance and mass measurements.

Inputs and Outputs

Inputs: Our program will sample the position of the PS2 mouse or the hand from the camera,

together with the time of each sampling value. The program can then determine the moving track, the real time speed of the cursor, and which objects the mouse/hand intersected with.

Outputs: The program will display several moving objects on to the screen and the moving track of the mouse. When the mouse is moving at a high speed, the width of the track will be bolder, so as to mimic an effect of the cut by a blade on the screen. The program will report the performance of the player. There will also be audio effect accompanying the movement of the mouse/hand, the cutting of the object, etc.

Hardware/Software Split

Hardware:

1. PS2 mouse action control and capability to detect the movement of the cursor and keystrokes.
2. Game console development: VGA graphic display using RGB color, game graphics such as different shapes and colors for different homeworks, projects, exams, etc.
3. Hand tracking module consisting of three stages: subsampling of each video frame, color based skin pixel detection, and hand detection/tracking.

Software:

1. Program and test the algorithms for mouse/hand detection and tracking the trajectory of objects.
2. Develop a GUI for the entire game and check for correctness.

Project Milestones

Milestone 1 (April 1)

1. Implement algorithms to detect the cursor position of the mouse and compute its speed.
2. Design all the graphics such as shapes and colors of floating objects in software.
3. Implement subsampling for each video frame for hand recognition.

Milestone 2 (April 15)

1. Integrate the algorithm and graphics on the FPGA.
2. Implement skin pixel detection and hand tracking.
3. Add audio effect for movement of the blade.

Milestone 3 (April 29)

1. Test that the game console works properly via simulation and real-time testing.
2. Integrate hand detection/tracking to control the sword or blade.
3. Test that the game console works correctly with hand detection/tracking.