ELE306 Spring2008 Final Project

Final Project Outlines:

I. Background

Tron is a movie, produced by Disney in 1982, which was notable for its pioneering use of early computer graphics. Although primitive by today's standards, Tron managed to include around thirty minutes of rendered graphics, as well as merged scenes including both real actors and virtual scenes. The reason for this seemingly odd combination is that during the movie, one of the actors is digitized and pulled into the computer system, where programs are sentient entities. There, the human and many of the programs are forced to compete in games until they are eliminated and deleted.

Most notable of these games is a sequence using virtual vehicles called light cycles, devices which are infinitely thin, leave impassable trails, and travel at incredible speeds. The participants then race around a large square arena attempting to force their competitors to crash into a wall or a trail left behind by one of the cycles. You may find many examples of TRON game on the web.

II. Project Description

Your goal is to implement the light cycle game, to the best of your ability, on the Altera Flex 10k FPGA board, using VHDL and your design experiences from previous labs and exercises. Since we will not have the luxury of virtual modeling and rendering, the game will be displayed as two-dimensional graphics on the provided VGA output. Two players will be allowed to compete at a time, and their light cycles will leave trails which will become obstacles. For the sake of playability, each player should not die after a single impact, but should have a maximum of 5 chances to play. Additionally, each player should be allowed their own keyboard, which implies that projects should be able to play against each other via serial communication, and send enough data to play on a remote screen.

III. Specifications

Besides the constraints mentioned above (each player has five lives; and use only the facilities available in the laboratory), you are to work out the rest of the specifications of the games yourself as the first step of your project. These steps are listed in the next section: project milestones.
Final Project Milestones:

The ten steps listed below are to be followed exactly. You should report the completion of each step or milestone to either Mr. Parys or me so we can record your progress. Each completed step will earn you 10% toward the project baseline score; which will in turns apply to the 25% overall score allocated for the project along.

It is recommended that you take the following steps as a working plan and proceed with the given sequence. It is important to approach a big project by taking small steps so that each step can be easily done.

You may do things out of sequence as you see fit. Demonstration of the completion of each and every step is still required. For instance, you may attempt to complete steps 3, 4 and 5 in one try. At the completion, if you can demonstrate what were described in steps 3, 4 and 5, individually, you will earn the 30% together. The risk is of course, if you fail, you will have nothing to show for and thus no point earned.

I. Working specifications
   • The first task is to complete the game specifications:
   • What is the screen look like?
   • What is the size of the playing field? (minimum size is 20X20)
   • What is the color scheme? (player1? player2? score? etc.)
   • What are the controls?
   • What happen when the game ends?
   • How to start a new game?
   • What will be the RAM usage? (only 18K bits are available.)

II. Frame and Background
   Show the frame (or borders) of the empty playing field. When “reset” or “start a new game” this empty playing field should show up again. (Remember that the RAM contents will not “reset”; you will have to fill it with “background” color to clear the field)

III. Player 1 moves
   Show only the first player. Player 1 should be controlled by the four keys on the keyboard as you’ve specified. In Tron game, Player 1 will leave a trace as it moves. At this time, you may double check on your “reset” capability implemented in step II above; make sure that the screen is cleared and that the player 1 is back to its starting position.

IV. Collision detections
   Now that player 1 is moving, add the collision detection feature to the game. Player 1 “dies” whenever it goes into
something other than background. At this point, the game stops when player 1 collided into something.

V. Scoring
At the beginning of the game give Player 1 five lives. Each time player 1 “dies” a live is taken away. The number of lives should be clearly displayed outside the playing field as you previously specified.

VI. End game & new game
The game ends when all five lives have gone. Implement your “end game” and “new game” features here. The “end game” feature can be as simple as freeze everything on screen and wait for the “new game” key to be hit. Fanciful end game feature will be implemented in step 9. You now have a single player game.

VII. Add Player 2 via keyboard
Add the second player by allocating another set of four keys on the keyboard for Player 2 controls. Repeat steps III, IV and V for Player 2 here. Remember the collision occurs when a player is moving into anything other than background.

VIII. Two players on keyboard
Proceed to implement scoring for player 2. Now you should have a two players TRON game with both players using the keyboard.

IX. Interesting end game features
When the game ends, the point of collision should flash continuously until a new game starts. You may determine the flashing rate and pattern yourself.

X. Player 2 from the RS-232
Now, move player 2 from the keyboard to the RS-232 (see Lab 6 for RS-232 hardware interface design). Remember that you do not need to convert the keyboard scan codes to ASCII codes in order to use RS-232.
With a switch, the game can switch between host and guest modes. This will allow two versions of your design to play together, via RS-232, while displaying the correct screen on individual machine.
Final Project Written Report:
The final project written report should include:

- Game specifications (as in step 1)
- Algorithms or flowcharts of your design. (Need not include the known/given VHDL codes such as keyboard.vhd, vga_sync, etc.)
- Screen shot of game plays (I will bring in a digital camera)
- Discussions on constraints, design philosophy, your general approaches and future outlook.
- Appendices should include VHDL codes of your design, compilation report summary on the FPGA hardware usage, and/or RTL level diagrams.

Final Project PowerPoint Slides:
You should prepare your presentation in PowerPoint slides (about 10-15 slides) covering exact topics as you have included in your written report. Do not include a detailed VHDL code listing. Instead, when necessary, show a segment of VHDL codes to help illustrate your point. You should email the PowerPoint files to me at least one day before the presentation.

Final Project Demonstration and Presentation:
The demonstration and presentation of your final project will occur on the scheduled final exam day for ELE306.