ELE307 Spring2009 Final Project

The final project will be performed by the team in three week. Every week, before leaving the lab, progress must be checked according to the project milestones. Presentation and final project report is due on the final examination day.

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 this light cycle game, to the best of your ability, on the Altera DE2 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 (lives) 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 first ten steps listed below are to be followed exactly. Every week before your team leave the lab, your progress should be verified by either the TA or the instructor. Each completed step will earn you 10% toward the project baseline score scaling factor. Steps 11 and 12 are your chance to earn 10% bonus each toward your final grade. However, you will earn the bonus points only if you complete all first 10 steps. Remember that you must be ahead of the schedule to have time for the bonus steps.

.

It is important to approach a big project by taking small steps so that each step can be easily debugged and verified.

Week 1

Step 1. Working specifications:

The first task is to complete the game specifications:

- What is the play screen look like?
- What is the size of the playing field? (minimum size is 60X60)
- What is the color scheme? (player1? player2? score? etc.) Minimum color depth is eight colors; or three bits per color.
- What are the controls? (Initially both players will use the same keyboard, later only one player will be on this keyboard)
- What happen when the game ends?
- How to start a new game?
- What will be the RAM usage? (only about 384K bits are available.)

Step 2. Block Digram, Algorithm and Flowchart

- Draw a rough block diagram showing the connection of all important components. You should continue to refine this block diagram as you go. The more detailed block diagram will be included in your final report.
- Derive a rough algorithm or flowchart of your design. You should refine this as you go and include it in your final report.

Step 3. 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) Depends on how the players' scores (or lives) will be displayed, some room outside of this frame may be needed.

Step 4. 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 3 above; make sure that the screen is cleared and that the player 1 is back to its starting position.

Week 2

Step 5. 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. The player could press a key (other than those for controlling direction) and restart the game. This means that the screen needs to be blanked, frame redrawn, and player is back to the initial location.

Step 6. 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 on screen as you previously specified. Alternatively, you may use the 7-segment displays to show the score.

Step 7. 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.

Week 3

Step 8. 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 4, 5 and 6 for Player 2 here. Remember the collision occurs when a player is moving into anything other than background. What happened when both player are moving into the same location at the same time?

Step 9. 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 same keyboard.

Step 10. 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. Alternatively, you could use your creativity and come up with your own fanciful end game feature.

Bonus

You could earn up to 10% toward your final grade for each bonus step below. However, to earn credit for step 11, you must complete steps 1-10. Similarly, to earn credits for step 12, you must complete steps 1-11.

Step 11. Player 2 from the RS-232

Now, move player 2 from the keyboard to the RS-232. At this stage, the second player is played on a Linux PC running "minicom". The local controls (in the form of ASCII codes) should also be sent out. For now, these ASCII codes will be displayed on the minicom screen.

Step 12. Switch player identity

With a switch, the game can switch between host and guest modes. The host is player 1 and the guest is player 2 locally at their own DE2 board, respectively. 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: (40%)

1. Title page: title, class, name of team member, etc.
2. Game specifications (as in step 1). Some descriptions are needed to justify your decisions and/or provide rationales.
3. Block diagram. Just enough details to understand your design.
4. Algorithms or flowcharts of your design. (Need not include the known/given VHDL codes such as keyboard.vhd, vga.vhd, etc.)
5. Discussions on constraints, design philosophy, your general approaches and future outlook.
6. 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: (30%)

You should prepare your presentation in PowerPoint slides (about 10 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 ELE307.

Final Project Grading Policy:

Scaling factor is the percentage of completion of steps 1 to 10.
10% for meeting project milestones each week.
The final project is 40% of the total grade

Total grade=scaling_factor * (3*weekly_verification + written report + Presentation)