edX (https://www.edx.org)

UTAustinX: UT.6.01x Embedded Systems - Shape the World

KarenWest (/dashboard) ▼

Courseware (/courses/UTAustinX/UT.6.01x/1T2014/courseware) **Course Info (/courses/UTAustinX/UT.6.01x/1T2014/info)**

Discussion (/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum) **Progress (/courses/UTAustinX/UT.6.01x/1T2014/progress)**

Questions (/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/)

Syllabus (/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/)

Embedded Systems Community (/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/)

Help

2. Function Description

2.1. Functionality: What will the system do precisely? You will design, implement and debug an 80's or 90's-style video game. You are free to simplify the rules but your game should be recognizable as one of these six simple games: Space Invaders, Asteroids, Missile Command, Centipede, Snood, or Defender. Buttons, and the slide pot are inputs. The Nokia5110 LCD, LEDs, and sound (Lab 13) are the outputs. The slide pot is a simple yet effective means to move your ship. Interrupts must be appropriately used control the input/output, and will make a profound impact on how the user interacts with the game. You could use an edge-triggered interrupt to execute software whenever a button is pressed. You could create two periodic interrupts. Use one fixed-frequency periodic interrupt to output sounds with the DAC. You could decide to move a sprite using a second periodic interrupt, although the actual LCD output should always be performed in the main program.

2.2. Scope: List the phases and what will be delivered in each phase. The first phase is forming a team and defining the exact rules of game play. You next will specify the modules: e.g., the main game engine, a module to input from switches, a module to output to LEDs, a module to draw images on the Nokia5110, and a module that inputs from the slide pot. Next you will design the prototypes for the public functions. At this phase of the project, individual team members can develop and test modules concurrently. The last phase of the project is to combine the modules to create the overall system.

2.3. Prototypes: How will intermediate progress be demonstrated? In a system such as this each module must be individually tested. Your system will have four or more modules. Each module has a separate header and code file. For each module create a header file, a code file and a separate main program to test that particular module. After the game is completed, you will create a single C file that compiles within the starter project to be uploaded for others to play.

2.4. Performance: Define the measures and describe how they will be determined. The game should be easy to learn, and fun to play. You will have the option to share your game with other students in the class.

2.5. Usability: Describe the interfaces. Be quantitative if possible. In order to allow other students to download, compile and run your game you must follow strict requirements for the pins used for input and output. These requirements are detailed at the top of the `SpaceInvaders.c` file within the starter project.

2.6. Safety: Explain any safety requirements and how they will be measured. To reduce the chance of spreading viruses we will only allow `SpaceInvaders.o` files with TM4C123 object code to be uploaded. The usual rules about respect and tolerance as defined for the forums apply as well to the output of the video games.

3. Deliverables

3.1. Reports: How will the system be described? You will create and upload a 60-sec YouTube video describing how the game works. Additional comments can be added to the description section of the YouTube upload explaining the purpose and functionality of your game.

3.2. Audits: How will the clients evaluate progress? There will be a discussion forum that will allow you to evaluate the performance (easy to learn, fun to play) of the other games.

3.3. Outcomes: What are the deliverables? How do we know when it is done? There are three options for Lab 15. First, Lab 15 is completely optional and you are not required to develop or share your game.

Second, if you wish to show your game to others, you will create a YouTube video, and upload a YouTube link to a Goggle form hosted on the class Google Drive. We will create a YouTube channel of all the videos submitted. This will allow others to watch your video and upvote the ones they like.

Third, if you wish to allow others to play your game, you must follow strict hardware specifications listed in the stater project. Next, you must store your `SpaceInvaders.o` somewhere on the web and provide a URL to the `SpaceInvaders.o` file. Other students can download your object file and play your game on their hardware. We will disable uploads and downloads on the last class day.

Help

About (https://www.edx.org/about-us) Jobs (https://www.edx.org/jobs)
Press (https://www.edx.org/press) FAQ (https://www.edx.org/student-faq)
Contact (https://www.edx.org/contact)

EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.

(http://www.meetup.com/edX-Global-Community/)

(http://www.facebook.com/EdxOnline)

(https://twitter.com/edXOnline)

(https://plus.google.com/108235383044095082735/posts)

(http://youtube.com/user/edxonline)

Terms of Service and Honor Code -
Privacy Policy (https://www.edx.org/edx-privacy-policy)