EE91:

Spacecraft Teamwork Game

Dan Pipe-Mazo

dpipemazo@caltech.edu

MSC 690

Winter Term 2014

I. OVERVIEW

The proposed project is a teamwork game with multiple, wirelessly connected game boards. Each board will contain a processing unit, a character display and multiple user input sources such as switches, knobs or an accelerometer. The goal of the game is to have the user perform the action displayed on the character display within a certain amount of time. However, the action shown on one user's display may only be able to be completed on a different user's game board. In this nuance lies the teamwork; users will have to communicate verbally to each other required actions and then perform any actions required on their own boards. This game is loosely based on the concept of the popular application "Space Team" which can be played with multiple connected iPhone or Android cell phones.

The processing for the game will be distributed among the game boards; there will be no central processing unit. The boards will communicate via a wireless channel. Each board being used in the game will display a request to the user via the character display on their game board at all times. Every time that a user triggers an action on their game board (such as pressing a button or flipping a switch), the board will broadcast that the action has been completed. If any other game board is waiting for the particular action to be completed, it will do so until it receives the message that the action has been performed. If an action is not completed within the allotted time, the game ends. The object of the game is to have it last as long as possible. The game will be designed to accommodate at least 8 game boards, and will dynamically adapt to the number of game boards present for each individual session.

The game will be space-themed, as the idea behind the game is that the users are working together to cooperatively pilot a "spacecraft". This game is intended for use in a ditch day stack.

II. OPERATIONAL DESCRIPTION

Inputs

Each board will contain an identical set of gameplay inputs. The challenge of the game lies in that gameplay commands issued on board 1 will potentially relate to inputs on boards 2, 3, 4, etc. This will force users to communicate the commands which they see on their personal displays to other users.

Each board will contain the following gameplay inputs:

- Two pushbuttons
- Two SPST switches
- One knob with greater than or equal to six positions
- One accelerometer capable of determining board orientation in three-dimensional space
- One keypad with greater than or equal to 9 keys
- One standard Radio Frequency Identification Device (RFID) reader

The gameplay inputs will be used as the objects of gameplay commands. In order to use the RFID reader input, users will each be given a passive RFID beacon in addition to their game board. This will force users to switch boards during gameplay, since a command may be issued forcing User X to scan their RFID beacon on User Y's game board.

Each board will also have both a reset and a begin button. The reset button will put the board into a device location mode, and each game board will attempt to connect with the other present game boards. After all boards are recognized, the user can press the begin button on their board to begin the game.

All buttons and switches will be active low, with output pull-up resistors and grounded input terminals.

Outputs

Each game board will have a character display which will display commands to the user. The display will contain at least 20 characters and will allow for scrolling of long messages. It will be capable of showing at least the first 128 characters in the ASCII character set.

Each game board will have a game health status indicator. This indicator will begin at its maximum value and will decrease as the game progresses according to the rules of the game.

Each game board will have K LEDs, where K is the maximum number of allowable game boards per game session. These LEDs will be used to show which boards are networked properly before beginning a game. Whenever a game board networks with another game board, it will light up one of these LEDs to indicate to the user how many boards are present in the game and which boards are active.

Wireless Communication

Game boards will communicate wirelessly via a 2.4GHz channel. Game boards will issue four different kinds of messages: command request messages, command complete messages, game health messages and board identification messages. All messages sent will have a message header which includes a unique board ID and the command type, followed by the command data.

Messages

Command request messages are generated by each game board. Command requests are generated at random from a list of all possible commands given the game boards present for the session. When a command request is generated, the command is displayed to the user via the character display, and then the command is broadcast to all other boards. A new command request is generated at the beginning of each session, after a game board receives a command complete message for its current pending request, or after a a board has been waiting for a command for an allotted amount of time.

Command complete messages are generated by each game board. These messages are generated every time a user completes a pending request by modifying one of the user input devices. Pending requests are identified by each board maintaining a list of all command request messages which it receives. When a user modifies a user input device and this modification matches a pending request, the game board will remove the pending request from the pending requests list and broadcast a command complete message to the other game boards. The other game boards, upon receiving this command complete message will also remove the pending request from their pending requests lists. Any board which was waiting for the command to be completed will generate a new command request.

Game health messages are generated by each game board. The game health initially begins at its maximum value. Every time that a command request is not completed within the allotted time, the game board which was waiting for the command will issue a game health message indicating all boards to decrease their health by a set value. When game health reaches zero, the game is over.

Board identification messages are generated by each board. After power-up or reset, each board will generate an identification message. This will allow all of the boards present to synchronize before the game starts.

Gameplay

Each game will begin with N or greater game boards for N users. When users decide to begin the game, they will press the reset buttons on their boards. After pressing the reset button, each board will emit a board identification message, allowing the boards to network. When each game board has discovered

another game board, it will light up one of its K status LEDs, where K is the maximum number of allowable game boards, indicating to the user how many game boards will be playing in the game. Once the user is satisfied with how many game boards have been recognized and are in the game, he or she will press the begin button on their board to begin the game. Once a single one of the networked users presses the begin button, the game will begin for all users.

Once the game has begun, each user will immediately see a command on their character LCD. Assuming that we are user W, the command may be similar to "Press button 1 on user X's board" or "Turn the knob to 5 on user Y's board" or "Flip user Z's board upside-down". The user will then have to communicate to the appropriate player the instructions which they see on their character display within a set amount of time. If the command is successfully executed within the amount of time, then the game proceeds and the user's board generates a new command. If the command is not successfully executed, every player's game health is decreased and the game continues until game health reaches 0.

The objective of the game is to last as long as possible without running out of health. As time progresses, the amount of time allotted for commands to be executed in will decrease, making the game more difficult over time.

III. HARDWARE

Each game board will require the hardware listed in the inputs and outputs sections above.

Each game board will require a microprocessor capable of serial communications, either through SPI or I^2C with the accelerometer, RFID and wireless devices. This microprocessor will ideally be clocked upwards of 10MHz.

In order to reduce the I/O line load on the microprocessor, the system may use a CPLD to help manage I/O. This CPLD would communicate with the microprocessor over a serial connection and would have enough I/O ports to monitor all of the user inputs and outputs. The CPLD will monitor user inputs for state changes, and will send serial commands to the microprocessor indicating the input which changed and its new state. The CPLD could also also handle driving the character display, the game health status indicator and the networking LEDs. Whether or not the CPLD will be used and the extent to which it may be used will be known once a microprocessor is selected.

Each game board will require a 2.4GHz wireless communications module. The module will communicate with the process via either SPI or I^2C and will alert the microprocessor when data has been successfully sent. This will allow for some amount of error-checking in the system and will increase system robustness.

Each game board will require a custom printed circuit board which will be designed to minimize total circuit footprint.

All hardware will be designed and chosen to minimize total parts cost.

IV. POWER

The game board will be designed to run on a battery pack which outputs between 3 and 5 volts. The battery pack will either allow for swapping of standard-sized batteries or for removal and recharging. The system will have enough stored power to allow for several hours of continuous gameplay.

V. PACKAGING

The game board PCB will be made as small as possible in order to minimize cost. The packaging for the PCB will be a single laser-cut panel which houses all of the user inputs. The panel will have sides and a back, allowing for the electronics to be enclosed within a thin box. The box will be made to be hand-held, and will have the potential to be mounted on a wall through mounting joints on the back of the box. The packaging will have a space-like feel, as the game is intended to simulate users cooperatively flying a "spacecraft".

All of the input devices will be mounted on the front panel in an organized manner. Next to each input will be the input's name, allowing users to translate commands from the character display to the input panel. Different game boards will have different names for the same inputs, i.e. a pushbutton on board 1 could be named "Fire laser" while on board 2 the same pushbutton could be named "Release waste".

The top of the panel will contain the character display, the game health status indicator, the K networking LEDs, the reset button and the begin button. The remainder of the panel will contain the gameplay user inputs, arranged in a fashion to be decided.

The back of the box will be removable to allow the swapping out of batteries.

VI. SCHEDULE

The schedule for the project (by week) is proposed as:

1. Choose and order all coponents
2. Create schematics
3. Prototype design on breadboard
4. Lay out, route and send PCB for fabrication
5. Write software
6. Assemble processor on PCB, get processor to boot
7. Assemble I/O on PCB, get processor to interact with I/O
8. Assemble wireless communications on PCB, get board-to-baord communications to work
9. Design and create packaging. Debug software.
10. Final assembly and test