SHARD

Structural Health and Rupture Detection

Team Members:

Julian Herrera, jherrera2020@my.fit.edu

Matthew Manley, mmanley2018@my.fit.edu

Matthew Meesit, mmeesit2019@my.fit.edu

John Bruce, jbruce2020@my.fit.edu

Paul Awad, pawad2021@my.fit.edu

CSE Faculty Advisor:

Dr. Silaghi, msilaghi@fit.edu

Client:

Dr. Willard/AEE Senior Design GSAs

Client Meeting Dates:

Every Friday at 3pm

Goal and Motivation

The overall goal of the project is to provide the client with a system that protects against micrometer debris in low earth orbit. Our goal is to provide the system with an interface between the human and the system. More specifically, to design and develop an application that utilizes specific data obtained from sensors attached to the outside structure (tiles), to display said data and tiles in a user friendly way, and to program a microcontroller to activate a mechanism that autonomously repairs the system when the data is not acceptable.

Approach

Visualize the real world structure of a shield tile system. In the application, the user will be able to create various model templates of a structure that is representative of what it should and does look like in the real world. Having the user create the model allows the user to have modularity when it comes to designing and visualizing the system. The user will be able to see a model of the system, making life easier when a rupture occurs because the user will know on which tile it happened.

Interact with real time data that is fed into the application from outside sensors. Interaction of the data is key when it comes to the system because the user must know what is happening around the structure in order to be made aware of the situation outside. And when a breach occurs, the user will want to know where that happens. The real time data is visualized through the model on the application in a way that is intuitive.

Choose what to do when sensor data reveals a breach in the structure. When a breach occurs, the user will want to take necessary actions in order to save his or her life. From the application, the user can take the necessary steps to ensure safety.

Save the data that is collected to a storage device for later use. This is useful for the user because he or she can know what happened during the lifetime that the application (and presumably the mission) was ongoing.

Algorithms and tools

One tool used is OpenGL. More specifically, gl3w (simple OpenGL core profile loading) with glm (OpenGL mathematics), alongside the core OpenGL specification provided by the OS. These tools/libraries/specifications were used in the code for 3D rendering of various objects.

Another tool used is wxWidgets, which is used for the user interface. The C++ programming language is used to implement the application.

Novel features/functionalities

None

Technical Challenges

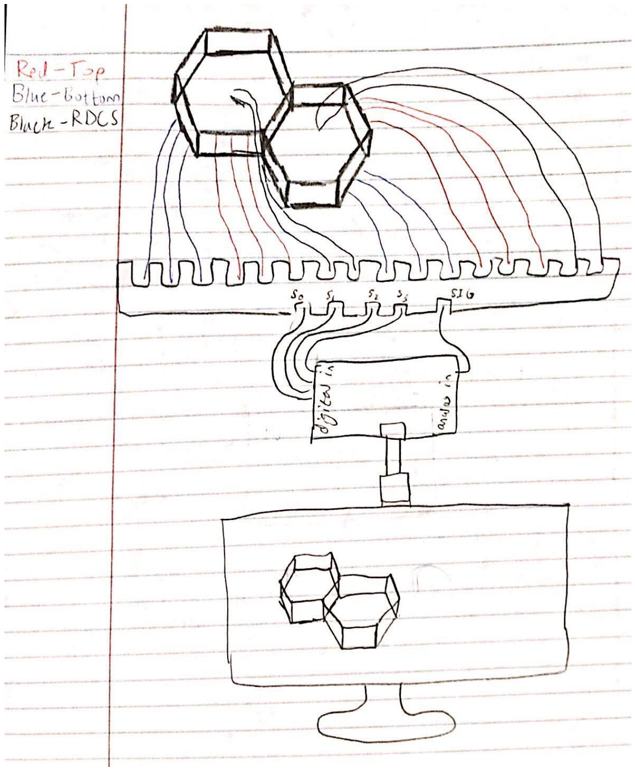
Collecting data from the sensors and turning it into understandable information.
Knowing which tile in the physical model corresponds to a tile in the virtual model. This will be a challenge because all that the sensors will be providing us is with the pressure applied to it.
Learning more about microcontrollers (or other electronics that interface with sensors) and how they interact with computers. We have never interfaced with sensors before.

One challenge faced last semester was the sensitivity of the vibration sensors. The sensors were too sensitive and maxed out everytime on the Arduino microcontroller. To remedy this problem, silicone nano gel adhesive tape is going to be wrapped around the sensor to lessen the sensitivity.

A challenge that was faced last semester was figuring out how to translate a mouse click on the 2D screen to a selection of a hexagon in the 3D space. This challenge was overcome by doing research into the problem and performing some tests.

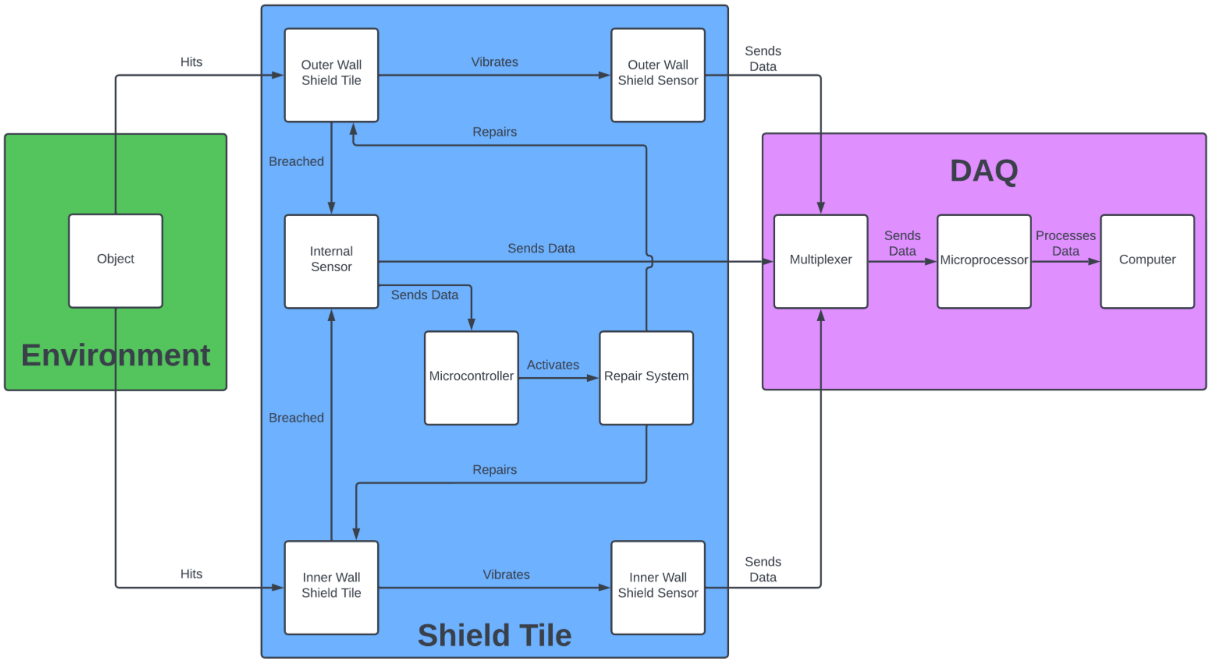
A challenge that we face this semester is the accuracy of tile positioning. For example, when adding the tiles in a circular fashion and then closing the last gap by adding a tile, the tile being added registers correctly the face (and corresponding tile) which was clicked on by the user. However, the newly added tile does not register the face behind it because there is no one-level connection between the newly added tile and the tile that was being added from. This is because of the code that was written, which is similar to that of a linked list except with six links to adjacent tiles. This issue must be addressed going forward.

Design



The above diagram shows an overview of the Data Acquisition System. The hexagonal tiles are each connected to 8 sensors. Three sensors on the top, three sensors on the bottom, and two sensors for the RDCS system. The sensors are connected to the multiplexer which feeds into the microcontroller. The microcontroller connects to the computer program interface, shown at the bottom of the diagram. The connection configuration of a virtual tile is set up by a specific UUID that relates to the physical tile. The UUID is determined by the position in the multiplexer, which multiplexer, and which microcontroller.

Design (cont.)



The above diagram is the System Architecture Diagram. It shows the three main systems which are the Environment, Shield Tile, and DAQ, and how they interact with each other. It also shows the various components inside each system. For example, an object/micrometeoroid from the environment hits the “Outer Wall Shield Tile” of the shield tile. This vibrates the “Outer Wall Shield Sensor” which sends data to the multiplexer of the DAQ system. This data is then processed by the microprocessor and sent to the computer interface.

Evaluation

Accuracy:

The application is accurate so far in most ways except for when tiles are connected in such a way that the tiles are closed. This is discussed in more detail in the technical challenges section.

Reliability:

- The application correctly adds an adjacent tile to the correct face 10 out of 10 times
- The application correctly closes 10 out of 10 times when the user clicks the exit button
- The application correctly opens multiple file tabs 10 out of 10 times

Progress Summary

Module/feature	Completion %	To do
User Interface	75%	1. Implement applicable functionalities for the toolbar
Modeling	80%	1. Rotation of hexagons 2. Fixing of some bugs
Sensor To Application Interface	0%	1. Create a panel/option to switch between modeling mode and connection to sensors mode 2. Implement the concept of connecting the sensors to the application
Simulation	0%	1. Create a panel/option to switch between modeling mode and simulation/execution mode 2. Implement the projectile events

Milestone 4 (Feb 13): itemized tasks

1. Implement Applicable GUI Functionalities for the toolbar
2. Create Welcome Page
3. Continue Sensor Experimentation
4. Allow for ability to rotate newly added tiles (hexagons)
5. Implement sensor to application interface

Milestone 5 (Mar 20): itemized tasks

1. Create and finish poster
2. Create and finish e-book page
3. Implement, test, and demo the simulation mode of the application
4. Start touch up features

Milestone 6 (Apr 17): itemized tasks

1. Create and finish user manual
2. Create and finish demo video
3. Finish touch up features

Task matrix for Milestone 4

Task	Julian	Matthew
Implement Applicable GUI Functionalities for the toolbar	10%	90%
Create Welcome Page	10%	90%
Continue Sensor Experimentation	50%	50%
Allow for ability to rotate newly added tiles (hexagons)	90%	10%
Implement sensor to application interface	90%	10%

Milestone 4 Description

Task 1: Implement Applicable GUI Functionalities for the toolbar

Last milestone, the toolbar was created and implemented into the GUI. In this task, functionalities will be implemented for the toolbar. Actions such as go forward, go backward, zoom in, zoom out, add tile, remove tile, save as, exit application, etc. will be added to the corresponding icon on the toolbar.

Task 2: Create Welcome Page

The welcome page will pop up upon opening the application. The welcome page will contain the S.H.A.R.D logo in the center with a space background. The welcome page will also include “New File”, “Open File”, “About”, and “Exit” buttons.

Task 3: Continue Sensor Experimentation

Experimentation to improve sensor functionality will continue this milestone. Silicone nano gel adhesive tape will be wrapped around the vibration sensors to reduce sensitivity. The arduino microcontroller will continue to be used to test the sensor and determine if the sensitivity is reduced by the silicone layer.

Task 4: Allow for ability to rotate newly added tiles (hexagons)

The purpose of this task is so that when the user is modeling on the software, the user can rotate the hexagons so as to emulate what the real world structure should or does look like. The rotation as of now will only be in 45 degree rotations. The reasoning for this is because if the angles went from anywhere between 0 and 360 degrees, some scenarios could occur wherein the hexagons do not look connected (because the angles are not in even increments).

Task 5: Implement sensor to application interface

The purpose of this task is to start implementing the connection interface between the sensors and the application. This involves the microcontroller (Arduino) to be connected to the program, the program to be running, and at least one sensor connected to the microcontroller. In the application, the user can switch from modeling mode to connection configuration mode wherein he or she will select tiles to specify the connection. These connections will be utilized later when the simulation/execution portion of the application is finished.