

# Structural Health And Rupture Detection (SHARD)

Team members (CSE): Julian Herrera and Matthew Manley

Team members (AEE/MEE): Matthew Meesit, John Bruce, Paul Awad

CSE Faculty: Marius Silaghi

Client: Dr. Willard/AEE Senior Design GSAs

# Goal and Motivation

- The project goal is to create a shielding system that protects against micrometer debris in low earth orbit Our goal is to provide the project with an interface between the human and the system.
- More specifically, we will design and develop an application that utilizes specific data obtained from sensors attached to the outside structure (tiles), and display this data and tiles in a user-friendly way
- The application will also have a microcontroller that activates a mechanism that autonomously repairs the system when there is a breach in the shield.

# 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.
- 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.
- 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.
- 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

- OpenGL using gl3w, a simple OpenGL core profile loading used to load the OS/video cards specific implementation of the OpenGL specifications
- glm, OpenGL Mathematical used for matrix mathematics to manipulate where objects are in 3D space
- wxWidgets, a graphics framework library used to provide our application with a user interface
- C++ programming language used

# Technical Challenges

- Accuracy of tile positioning - current problem
  - Inaccurate updating of a tiles adjacency list (of other tiles) because of how we have our algorithm right now to determine connections when adding hexagons to the scene
- Rotation - foreseen problem
  - Because of how rotation works, it is challenging to fully connect a composite structure when rotations are at different angles (because the angles don't match up, causing disparities when trying to connect the last pieces)
  - An undesired solution is to only rotate at fixed 45 degree increments
- Vibration sensors - current problem
  - Our team is using vibration sensors to determine if an impact occurs on the tile
  - The sensors we have a maximum output voltage of 100V, however when connected to the arduino, only a maximum of 5V can be used since that is how much the Arduino can handle
  - Possible solution is to use a voltage divider, or maybe something called an Arduino 4 Relay Shield

# Design

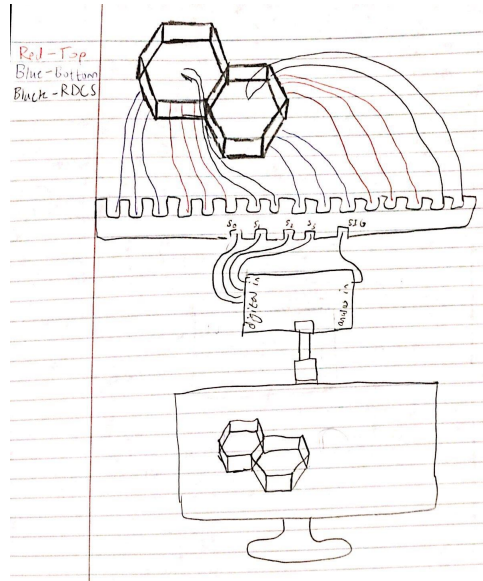


Figure 1: Data Acquisition System

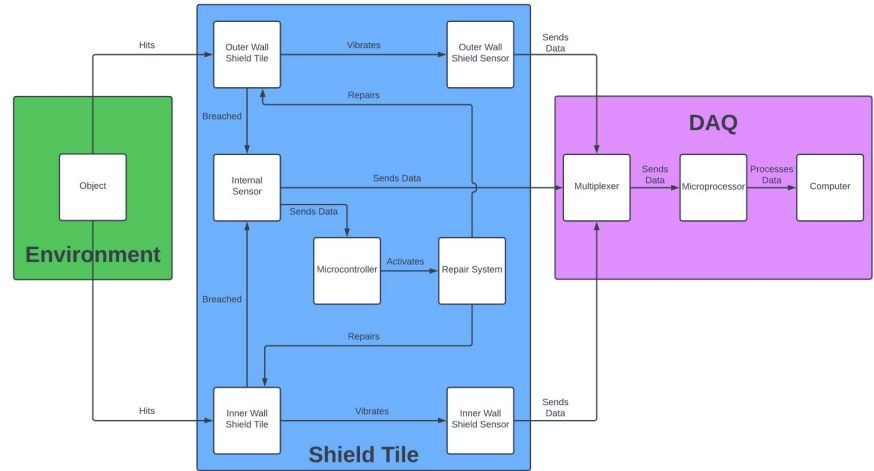


Figure 2: System Architecture Diagram



# Evaluation

- Application is accurate to a great extent except for that issue discussed in the technical challenges with the accuracy of tile positioning in the logic of the program



# Progress Summary

Module/feature	Completion %	To do
User Interface	75%	<ol style="list-style-type: none"><li>1. Implement applicable functionalities for the toolbar</li></ol>
Modeling	80%	<ol style="list-style-type: none"><li>1. Rotation of hexagons</li><li>2. Fixing of some bugs</li></ol>
Sensor To Application Interface	0%	<ol style="list-style-type: none"><li>1. Create a panel/option to switch between modeling mode and connection to sensors mode</li><li>2. Implement the concept of connecting the sensors to the application</li></ol>
Simulation	0%	<ol style="list-style-type: none"><li>1. Create a panel/option to switch between modeling mode and simulation/execution mode</li><li>2. Implement the projectile events</li></ol>

# Milestone 4

- Implement Applicable GUI Functionalities for the toolbar
- Create Welcome Page
- Continue Sensor Experimentation
- Allow for ability to rotate newly added tiles (hexagons)
- Implement sensor to application interface

# Milestone 5

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

# Milestone 6

- Create and finish user manual
- Create and finish demo video
- Finish touch up features

# Milestone 4 Task Matrix

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

- Implement Applicable GUI Functionalities for the toolbar
  - Last semester we added in the toolbar and menu items, now we need to program the functionalities that are designated for these items
- Create Welcome Page
  - Purpose is to provide a launching point for the users to use the application
  - Includes buttons such as creation of a new model, opening of an existing model, an about, and an exit
- Continue Sensor Experimentation
  - Ongoing sensor experimentation with the whole team (non-CSE majors) for use in the system
- Allow for ability to rotate newly added tiles (hexagons)
  - Rotation should be in 45 degree increments
  - Purpose is to emulate a 3D structure of tiles (hollow) so that the user has the ability to create a model that resembles a spacecraft or vessel
- Implement sensor to application interface
  - Purpose is to start work on connecting the sensors to the application, to lay the framework for future work



Questions?