

# 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

# Client and Advisor Meeting Dates

## Client: Dr. Willard

- September 9, 2022 - Discussed the overall project (including non-CSE) and its goals and objectives
- September 16, 2022 - Discussed the overall project's system requirements and the sensors involved.
- September 23, 2022 - Discussed more about the sensors and hardware capabilities.

## Advisor: Dr. Silaghi

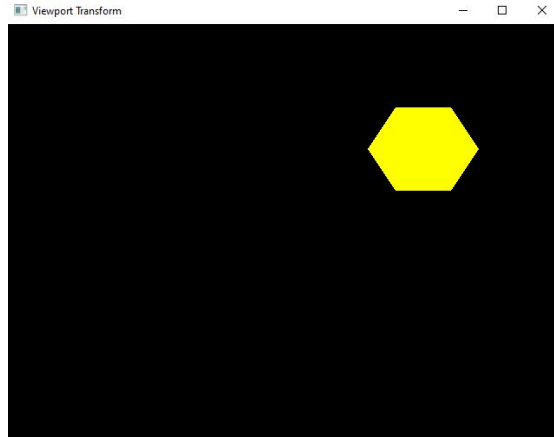
- September 20, 2022 - Met with Dr. Silaghi to discuss the necessary components of the project.
- October 4, 2022 - Met with Dr. Silaghi to discuss our progress.

# Overview of accomplished tasks

- GUI API decision
- Microcontroller/hardware system design
- Requirements
- Design
- Test Plan

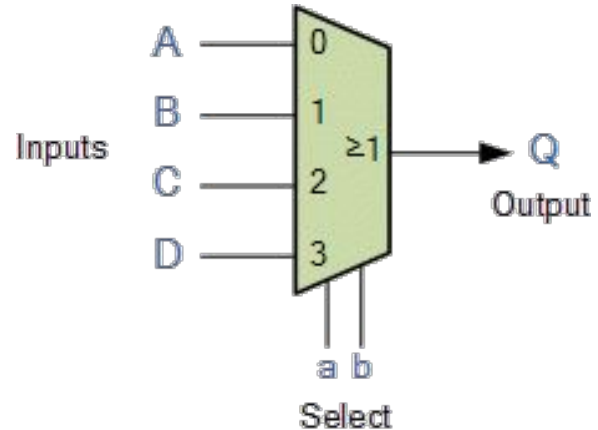
# GUI API Decision

```
9  /* Initialize OpenGL Graphics */
10 void initGL() {
11     // Set "clearing" or background color
12     glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque
13 }
14
15 void display() {
16     glClear(GL_COLOR_BUFFER_BIT); // Clear the color buffer with current clearing color
17
18     glBegin(GL_POLYGON); // These vertices form a closed polygon
19     glColor3f(1.0f, 1.0f, 0.0f); // Yellow
20     glVertex2f(0.4f, 0.2f);
21     glVertex2f(0.6f, 0.2f);
22     glVertex2f(0.7f, 0.4f);
23     glVertex2f(0.6f, 0.6f);
24     glVertex2f(0.4f, 0.6f);
25     glVertex2f(0.3f, 0.4f);
26     glEnd();
27
28     glFlush(); // Render now
29 }
30
31 /* Main function: GLUT runs as a console application starting at main() */
32 int main(int argc, char** argv) {
33     glutInit(&argc, argv); // Initialize GLUT
34     glutInitWindowSize(640, 480); // Set the window's initial width & height - non-square
35     glutInitWindowPosition(50, 50); // Position the window's initial top-left corner
36     glutCreateWindow("Viewport Transform"); // Create window with the given title
37     glutDisplayFunc(display); // Register callback handler for window re-paint event
38     initGL(); // Our own OpenGL initialization
39     glutMainLoop(); // Enter the infinite event-processing loop
40     return 0;
41 }
```

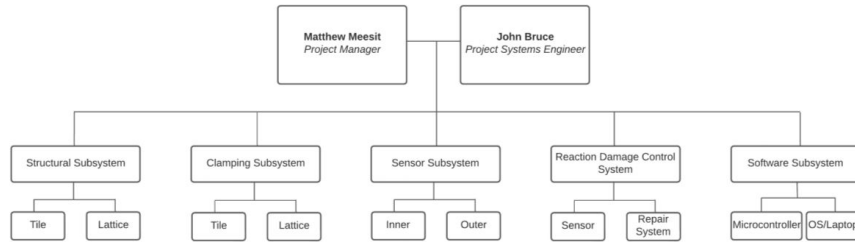


# Microcontroller/Hardware System Design

- 16 sensors max connected to one multiplexer
- One multiplexer connected to raspberry pi
- Raspberry pi connected to laptop
- Each sensor will be connected to a specific pin number on the multiplexer, which allows the user to identify which tile on a virtual model corresponds to the tile on the physical structure

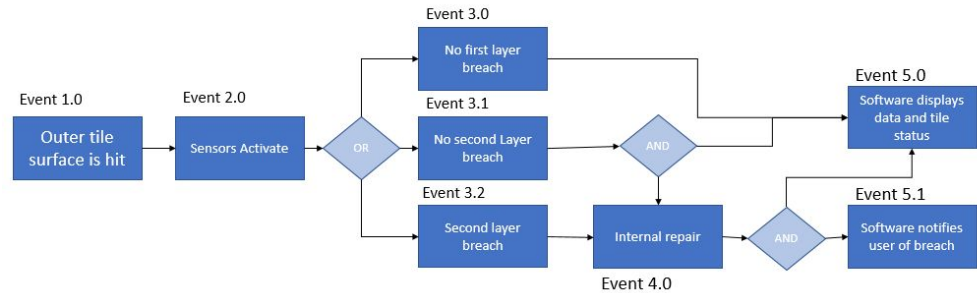


# Design



## System Architecture

## Concept of Operations



# Sample Test Cases

Code	TC012
Description	Simulate using projectile events
Preconditions	Model exists
Procedure	Create a projectile event by entering in meteoroid speed, angle, and location.
Valid Input	User enters in meteoroid velocity, density, starting position, and time of collision
Invalid Input	User enters in non-numeric data
Expected Output	Simulation shows projectile event three-dimensionally and whether or not the collision created a dent or rupture

# Milestone 2 Tasks

Task	Julian	Matthew
1. Increase understanding of the correlation between the physical sensors with the virtual tiles	50%	50%
2. Experiment with multiplexer and raspberry pi connection with each other and with the laptop.	50%	50%
3. Create the basic structure of GUI	50%	50%





# Questions?