



## Project Status report

Name:	Mason Lane
Community (UN SD goal):	SDG 14
MVP #	3
Sprint cycle dates:	Nov 16 - Dec 2

Project Name	aeS
Blurb	<p>aeS aims to be a new web/mobile platform seeking to provide daily interaction/learning through the form of learning activities/games that the user must frequent. One issue with many environmental learning games is that they are quite often one-and-done activities. You browse the game, learn and never come back. It's very hard to appreciate the importance of an endeavor with these types of games. Infact, many mobile learning games are more "fun" than effective. I hope to foster a community that brings back visitors, who wish to observe the fruits of the game communitie's labors. "winning" the game will be a group effort, and will encourage students to return. Upon returning, we hope to draw their attention back to the SDG at hand with new news/information.</p>
For Week Ending	Dec 2 MVP 3
Project Status	Green/yellow
Status Description	<p>Worked mainly on the backend to enable multiplayer gameplay.</p> <p>Did not achieve ip tracking.</p> <p>Game currently does not do websockets with subscriber patterns. Instead, the game follows a typical get/post system. This is not efficient and I cannot depend on its scalability. Would need to change this in other MVPs</p> <p>I think a generated perlin noise image might better store information about the trash location. Each pixel could correlate to a piece of trash. Black = trash, white = water. From here, selected trash items would switch from black to white pixels, until all noise has been removed and we have a white plane. In this way, we can easier track pixels without having to store thousands of IDs and metadata.</p> <p><a href="https://pinetools.com/random-bitmap-generator">https://pinetools.com/random-bitmap-generator</a></p> <p>Did not enable panning</p> <p>No pointer - however, I think an effect would be more effective - like waves. Particle effects are certainly more possible, but I did not have time to do them</p> <p>Examined performance issues. Made some improvements</p> <p>no fade-in effect for popups</p>



## Activities—During the past sprint cycle

Backend for multiplayer gameplay - standard get and post

Looked into using noise as a replacement for multiple DB rows  
Addressed some performance issues

MySQL db server

NodeJS api

Frontend sends requests to api, which then queries the db.

## Project Issues

Cannot pan the game yet

Performance is still sluggish, but with marked improvements.

no ip tracking

no pointer/net/particle effects.

jarring popups. UI feels blunt.

Not following a subscriber pattern for websockets

## Project Changes

No changes since last time. I think particle effects for clicks would be better than pointers. I can't really see a place for unique pointers, as mobile users would not benefit from it. Further, changing the pointer would deviate from the user's familiarity. The most I would do is change the color or arrow - but retain the familiar arrow shape. A "net" is hard to see what your pointer is pointing at. Is it the tip of the net? The middle? The end of the stick? From my experience, I have never enjoyed a fancy pointer because most times it just makes clicking confusing. With oddly-shaped pointer, you think your pointer is on a button, but its not.

## Activities—Potential future

Change to event listeners with a subscriber pattern. Move towards websockets instead of handshakes. Instead of asking the api for updates, the api should report to its subscribers of updates

implement panning

add more trivia

performance improvements

look into using a bitmap to track waste instead. This would likely improve performance as we would need to iterate far less.

Move from local dev environment, and host a live version on the web.

Convert my PC into a server using docker



## Reflection

Do you feel "on track"?

Despite not being a websocket, I'm quite happy with the multiplayer demo I was able to achieve.

What progress do you particularly feel good (great) about?

Creating a nodeJS api with express which queries from mysql db. This was an excellent refresher, and much needed in my opinion. I think we need more full-stack projects. I have observed that students (myself included) are often exclusively familiar with frontend development. Most internships are full stack or frontend. I think we need more backend classes, especially following capstone.

What barriers (if any) do you feel is/are a current impediment to success?

I will need to re-work my current multiplayer solution, so I question if the existing solution is of any value.

What help (if any) do you require to move positively forward?

N/A

What questions or concerns do you have (if any)?

N/A

---