



Project Status report

Name:	Mason Lane
Community (UN SD goal):	SDG 14
MVP #	1
Sprint cycle dates:	Nov 3 - Nov 16 (Shorter 1 wk sprint due to personal time off)

Project Name	aeS
Blurb	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.
For Week Ending	Nov 16 MVP 2
Project Status	Yellow
Status Description	<p>Popup works as intended for trivia and quota.</p> <p>Refactored some code that was causing a major performance drop. FPS is still low and needs to be looked at.</p> <p>camera panning issue persists no "net" animation.</p> <p>Worried about leaving enough time for websocket exploration. Websockets are the best solution for this kind of project, but I've never done one - so I may have to default to a simpler solution at the cost of performance (to meet mvp3).</p>

Activities—During the past sprint cycle

Created trivia and quota popup overlay
Reads json file for trivia
Manually tested behavior.
Performance fixes
Changed cursor (as a demo for the possible fishing-net pointer)

Project Issues

Cannot pan the game yet

Performance is sluggish.

Given the aforementioned issues, I worry that I may not have left sufficient time for the database side of things.

Project Changes



No changes since last time. If websockets become too complicated to achieve, I may resort to a non-standard handshake solution.

Aforementioned time constraints may bar me from including IP tracking for a player's unique score/turn timer. Perhaps cookies would offer a faster solution - though one that is easily bypassed.

Activities—Planned for Next Week

Work on implementing websockets. I will start with the objects themselves, and then move on to ip tracking. IP tracking is not as important as the actual object tracking, as object tracking will enable the possibility of a multiplayer demo.

Make panning possible (otherwise, fit objects to screen).

Add new mouse pointer if possible.

Examine performance issues.

Add fade-in effect for popups

Reflection

Do you feel "on track"?

I feel less than on-track, but within expected margins. I voiced my anxieties about this mvp in previous communication. Luckily, I got a lot of the work done in mvp1 to offload some of the expectations in mvp2.

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

Reading json for the popup was an exceptionally difficult task. I learned a lot about how to handle future<type> properties. In my project, I use a Future<List> to create the popup contents only AFTER the json is read. With the way overlays work, the json cannot be read until the user has clicked a tappable at least once.

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

Given my inexperience, I am anxious about the game's performance. I have a fairly good PC for development, so I need to test this application on weaker machines.

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

Resources on websockets, tutorials - especially ones that involve Flame, Flutter or just the language Dart would be useful I imagine.

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

N/A

DISCUSSION BELOW

DISCUSSION BELOW

DISCUSSION BELOW



Quick (It's not quick I lied) note on communities of practice;

As it turns out, it is exceptionally easy to embed flutter applications within existing HTML websites. Namely by using an IFrame. Wordpress - for example - has its own tutorial on how to do exactly this.

Knowing this, I've put more thought into whether my application should be a platform or a tool. It was originally envisioned as a platform (offering its own blog, etc). However, this would require a lot of expert curation. On the other hand, there are existing resources that offer the same learning opportunities. As such, I believe it would be ideal for my application to be an embeddable tool.

I could then create my own website/blog that also embeds the game, phone apps that consist of *just* the game (or perhaps a blog). In this way, members of the COP may use my application as supplemental material for their own content while not having to borrow or depend on the blog/info-sheet features. I would reach more users this way, while still being able to offer my own blog in the future.

Additionally, this direction separates concerns and enables re-use throughout the community. It is more in-line with Domain-Driven-Design, and Object Oriented principles. In short, users should not *need* my proposed blog to enjoy the game with other tools.

For example, if miniclip wanted to embed my game as it was originally proposed, they would need to suffer including my side-blog features.