Sprint 8

10 day duration (Ends 14th of July) ~116 pomodoros of work.

Product Backlog Tasks

1. As a player, I always want to try and make my transportation network more efficient, and need to do so in order to achieve my goals.
2. As a player, I can understand what my transportation network is doing just by looking at it.
3. The game's UI displays all the information a player needs to know in a primarily visual and acoustic manner.
4. As a player, I have access to a variety of maps that offer me different challenges and opportunities.

Basic Design

Story

The same as in all the previous sprints. It's not clear that developing a cohesive fiction is going to be a good use of my time.

Mechanics

Same as in Sprint 7, with a tweaked score system. Players must maintain the score necessary to win for a certain period of time, so that they do not achieve victory on an unstable configuration.

Aesthetics

I'll try to improve upon the aesthetics of the game. It still relies on simple geometic shapes for the terrain and the blobs; but ideally Societies, ResourceDepots, ConstructionZones, HighwayManagers, and Highways themselves should now look something like the concepts they represent. The aesthetic style will be whatever I can find.

Technology

The primary deployment remains a PC executable, though I'll also try to deploy it as a web application to make it easier to present to prospective employers.

Risk List

1. My design isn't based on rigorous and repetitive playtesting, and it seems unlikely I'll be able to playtest it properly.
    a) The same problem, the same ultimate conclusions. I need to focus on improving what user experience I can and on making maps. I can't make any more progress on improving the underlying mechanics of the game.
2. The game's victory conditions might be so lenient that players can take shortcuts and don't have to construct stable networks, or so cumbersome that players find it frustrating.
    a) Instead of having a certain score that players need to accomplish, I can instead specify a number of societies at various tiers that players need to support to achieve victory. I can also set a victory timer that begins counting down when the victory conditions are satisfied and stops counting down when any society no longer has its needs satisfied. That way, players will need to establish stable connections.
    b) I can keep that timer from getting annoying by fiddling with the production and consumption cycles. Decreasing the frequency of production and consumption gives resources more chance to get from place to place, which should make intermittently stable configurations less likely by making it easier to get the required resources to a society in time. I can also change the rate at which resources travel down highways for the same effect.
3. There might be bugs and unexpected problems with my various builds that don't become immediately apparent in the editor.
    a) I'll need to start testing the builds earlier than I did previously, building a web player no later than tomorrow if I want to deploy it. I'll also need to thoroughly test both types of deployable, looking for problems and inconsistencies to address. Likely those attempts will help populate the sprint backlog.
4. It's not clear to me that ResourceBlob serialization works properly.
    a) Simply put, I'll need to test it and address any problems that arise.
5. There might not be enough maps to demonstrate the complexities of the game.
    a) I need to devote a considerable amount of the development cycle designing at least one more map, ideally two. I'll need to carefully define the priorities of designing these maps so that I don't end up filling the whole sprint with programming and sensory polish tasks.
6. I still don't have a strong theory or intuition about how to design maps, which might make it difficult to design them properly.
    a) I've made at least some progress in designing maps. The only way I can think of to resolve this problem is to play the maps and do so intelligently. Every time I run through a map, I need to have a very specific question that I'm trying to answer and I cannot let myself fiddle around with things that ultimately do not matter. Unfortunately I don't have a more formal mitigation than that. Map design is perhaps my least developed skill at the moment.
7. There might be too great a jump between the last tutorial and the first normal map.
    a) I could consider creating an intermediate map, requiring not the support of one city but of two towns. That's a much simpler task, but by presenting a constrained map with little development I could probably make it an engaging one.
8. The audio for highway drawing isn't well thought out and might make highway drawing feel or sound strange.
    a) I should steal the road-building paradigm from Cities: Skylines and make the drawing of highways click, rather than produce the engine-like noise I have now. I won't steal the assets themselves, mind you, but that paradigm is better-tested than anything I could come up with in a short time. I should use their discoveries to my advantage.

9. HighwayManagerDisplay does not update itself in anything resembling real-time, which could make HighwayManager difficult and irritating to interact with.
    a) I can underpin HighwayManagerSummary with the HighwayManager it's referencing, so that HighwayManagerDisplay can update itself from current information. It'll then be a simple matter of periodically refreshing (maybe once every second) the display with a simple coroutine.
10. I have not used Highway priorities in several months, which means that no part of my design takes such priorities into account. They are currently a redundant element that might confuse players.
    a) The easiest way to extract priorities from the game would be to remove the UI that sets them. That way, every highway will have the same priority and players will have no way of setting them. I don't think there's a strong case for removing highway priorities from the codebase more thoroughly.
11. With the terrain hex grid in place, the interaction for selecting a map node has unusual affordances. Players cannot select a MapNode by the region defined by it, which might feel unintuitive. The terrain grid might also make the central sphere around the MapNode redundant, that it only contributes to visual clutter.
    a) I don't think that this is avery high-priority event. If I do want to address it, I can simply place appropriate OnClick callbacks on all of the terrain hexes, and then tell MapNode to listen to them and push OnClick events when any of its associated tiles are selected. I'm sure there's a more intelligent way to do things, but it's probably not worth my time to pursue it.
12. The meshes and textures associated with Societies, HighwayManagers and whatnot are not up to new aesthetic standards of quality. It might be unpleasant or unintuitive for players to reason about upright and sideways cylinders, and poor model quality might negatively impact user experience.
    a) The simplest solution would be to find a single mesh for societies in general, some sort of building, and use that for every complexity of society, applying the color to represent the exact complexity. Another simple option would be to find a mesh for each tier of complexity and color it appropriately. I could also use sprites and tint them with the appropriate color.
13. The game still has performance issues, which might negatively impact player experience during configurations of even moderate complexity.
    a) Though it will likely remain a low-priority event, I can always do a more extensive profile of the game's performance and find ways to improve it. The biggest candidates for change in my mind are object pooling for ResourceBlobs and more efficient blob movement.
14. ConstructionZones do not provide good feedback when they are created, completed, or destroyed, which might negatively impact player experience.
    a) Adding sounds to the various ConstructionZone events should suffice here. I wouldn't even need to incorporate other modules. I could apply an audio source to ConstructionZone that triggers when the Zone completes its project, and I could handle creation and destruction through UI sound.
15. Societies do not provide on-map evidence that their needs are not being satisfied, which might make it harder for players to tell, at a glance, which societies are having problems.
    a) The easiest solution here would be to add a UI element, a small symbol that represents unsatisfied needs, that hovers near societies when their needs are unsatisfied. I should be able to place a canvas underneath every society with the appropriate symbol, toggling it on and off at appropriate times.
16. Societies do not make noises when they complexify or decomplexify, which might negatively impact user experience and also make it harder to tell when parts of the current configuration are changing, especially off-screen.

    a) That'll be a simple matter of adding a pair of audio sources to each Society that play complexification and decomplexification sounds respectively. It'll be only a small amount of code in Society proper, though finding appropriate sounds will take some time.
17. Map 1 is not sufficiently tested and might not support the desired aesthetics of play.
    a) I'll need to test Map 1 at a reasonably high priority (possibly even as a critical priority), to see how challenging it is to set up a stable configuration.
18. It's difficult to pick out edges over certain terrains, which might make it difficult for players to judge the topology of a given map.
    a) I can fiddle with the appearance of MapEdges, especially playing around with their color and the color of the various pieces of terrain that obscure it. I might consider making MapEdge a lighter gray to stand out more on darker colors, and then change Grassland's material to be a bit darker. I could also make the edges thicker.

Sprint Backlog

To Do

Critical (0 pomodoros of work remaining)

1.

Important (0 pomodoros of work remaining)

1.

Desirable (0 pomodoros of work remaining)

1.

In Progress

Critical

1.

Important

1.

Desirable

1.

Completed

Critical

1. Alter the scoring system so that it considers the number of societies belonging to each tier, rather than some common score, for victory.
    a) Estimated 2 pomodoro duration.
    b) Took 4 to complete.
2. Add a timer to the victory system that provides victory only if a player sustains a stable configuration without any unsatisfaction of needs for a certain period of time.
    a) Estimated 2 pomodoro duration.
    b) Took 2 to complete.
3. Make MapEdges stand out more on the terrains they run over.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
4. Run through the PC standalone build, looking for inconsistencies and errors that need to be fixed.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
5. Change the startup procedures, disabling camera controls while TitleScreen is up and providing TitleScreen with a background that covers the whole screen.
    a) Estimated 1 pomodoro duration.
    b) Took < 1 to complete.
6. Add Exit Game functionality to the escape menu.
    a) Estimated 1 pomodoro duration.
    b) Took < 1 to complete.
7. Add Load Session functionality to the escape menu.
    a) Estimated 3 pomodoro duration.
    b) Took 2 to complete.
8. Add Exit Session functionality to the escape menu.
    a) Estimated 3 pomodoro duration.
    b) Took < 1 to complete.
9. Find a new model for each of the Society complexity tiers.
    a) Estimated 5 pomodoro duration.
    b) Took 6 to complete.
10. Color and incorporate the various Society sprites into the game.
    a) Estimated 5 pomodoro duration.
    b) Took 3 to complete.

11. Modify the audio for highway drawing, aligning it with the suggestions of mitigation 8.
    a) Estimated 3 pomodoro duration.
12. Create a rough draft of a map that requires the construction of two towns, whose complexity is somewhere between the last lesson and the map currently named Map 1.
    a) Estimated 3 pomodoro duration.
    b) Took 1 1/2 - 2 to complete.
13. Test and iterate upon the rough draft of the two-town map.
    a) Estimated 5 pomodoro duration.
    b) Took 1 1/2 to complete.
14. Give players the ability to permit individual ascensions to make accidental ascensions harder.
    a) Estimated 2 pomodoro duration.
    b) Took 3 to complete.
15. Test and iterate upon Map 1 to make sure that it supports the desired aesthetics of play.
    a) Estimated 8 pomodoro duration.
    b) Took 3 to complete.
16. Create a credits and attribution page.
    a) Estimated 2 pomodoro duration.
    b) Took 2 to complete.

Important

1. Change ConstructionPanel so that all buttons always display their cost even when disabled.
    a) Estimated 1 pomodoro duration.
    b) Took < 1 to complete.
2. Remove the UI that permits the modification of highway priorities.
    a) Estimated 1 pomodoro duration.
    b) Took a fraction of a pomodoro to complete.
3. Find a new model and texture for BlobTubes.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
4. Find a new model and texture for ResourceDepots.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
5. Find a new model and texture for ConstructionZones.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
6. Find a new model and texture for HighwayManagers.
    a) Estimated 2 pomodoro duration.

b) Took 2 to complete.

7. Add sounds for the destruction, complexification, and decomplexification of Societies.
    a) Estimated 5 pomodoro duration.
    b) Took 8 to complete.
8. Add sounds for the creation, destruction, and completion of ConstructionZones.
    a) Estimated 5 pomodoro duration.
    b) Took 1 to complete.
9. Apply sounds to the destruction of ResourceDepots and HighwayManagers.
    a) Estimated 1 pomodoro duration.
10. Deploy a web application, seeing how such a deployment affects the execution of the game, that we might determine how viable that method of deployment is.
    a) Estimated 5 pomodoro duration.
    b) Took 8 to complete.
11. Create a rough draft of a second map that addresses one of the specified concepts.
    a) Estimated 5 pomodoro duration.
    b) Took 6 to complete.
12. Test and iterate upon the rough draft of the second map to iron out any problems.
    a) Estimated 8 pomodoro duration.
    b) Took 2 to complete.
13. Change the ScoreTracker so that it provides more information about what is needed in order to win and why a given configuration isn't ticking down (providing the society types that are unstable, for instance).
    a) Estimated 2 pomodoro duration.
    b) Took 2 to complete.
14. Figure out how to bound the camera so that it never gets too far away from the map.
    a) Estimated 1 pomodoro duration.
    b) Took 2 to complete.
15. Fix various UI bugs scattered throughout the codebase.
    a) Estimated 3 pomodoro duration.
    b) Took ~6 to complete.
16. Check each of the existing maps and fix any inconsistencies in their structure.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
17. Update the highway-selection code to work with Highways visualized by sprites.
    a) Estimated 1 pomodoro duration.
    b) Took < 1 to complete.
18. Return all configurations to a reasonable state and set up the MapPermissionManager with the appropriate values.

a) Estimated 1 pomodoro duration.

b) Took < 1 to complete.

19. Figure out how to, in NewGameDisplay, put all of the permitted maps above all the unpermitted maps, and how to clearly indicate in the map selection section which maps can and cannot be played.
    a) Estimated 1 pomodoro duration.
    b) Took 1 to complete.
20. Identify precisely what is causing performance issues at moderately complex configurations.
    a) Estimated 5 pomodoro duration.
    b) Took 4 to complete.
21. Run through the game and its UI, cleaning everything up.
    a) Estimated 3 pomodoro duration.
    b) Took 3 to complete.

Desirable

1. Add a UI element to Society that communicates unsatisfied needs to the player in an obvious but unobtrusive way.
    a) Estimated 2 pomodoro duration.
    b) Took 1 to complete.
2. Create a rough draft of a third map that addresses one of the specified concepts.
    a) Estimated 5 pomodoro duration.
    b) Took 3 1/2 to complete.
3. Test and iterate upon the rough draft of the third map to iron out any problems.
    a) Estimated 8 pomodoro duration.
    b) Took 2 to complete.
4. Greenlight all existing unit tests.
    a) Estimated 5 pomodoro duration.
    b) Took ~2 to complete.

Abandoned

Critical

1. Change HighwayManagerDisplay so that it periodically refreshes its display while active.
    a) Estimated 1 pomodoro duration.
    b) Task was already complete.
2. Make sure that changes to BlobDistributor haven't broken the game.
    a) Estimated 2 pomodoro duration.
    b) Task was already complete.

Important

1. Discover and fix the various inconsistencies that have arisen in the web deployable.
    a) Estimated 5 pomodoro duration.
    b) Decided that it was better to ensure the proper behaviour of one platform than produce an additional and incomplete version that needs special attention.
2.

Desirable

1. Permit the selection of a MapNode by selecting one of its associated hexes.
    a) Estimated 2 pomodoro duration.
    b) Was deemed an irrelevant modification that adds little to the game.
2. Modify the mesh for MapNode to make it flesh with the aesthetics of the map better.
    a) Estimated 2 pomodoro duration.
    b) Was deemed an irrelevant modification that adds little to the game.
3. Test ResourceBlob serialization into sessions, resolving any problems that arise.
    a) Estimated 2 pomodoro duration.
    b) Was essentially already done by the time I got to it.

Review

Strategy Blobs, now in its final presentable form, is a much more aesthetically polished and UX-friendly game than it was at the beginning of the sprint. The sheer number of improvements to the game is actually quite surprising to me. Every object, rather than being represented by a cylinder or a box, now has an icon associated with it that is at least somewhat reminiscent of its true purpose (with the exception of villages, which have icons that don't quite make sense for them). There are also now a lot more non-lesson maps than there were at the outset, and which cover a larger segment of the game's mechanical base. One of the maps makes considerable use of forest clearing, for instance, which means that mechanic is now more thoroughly integrated into the game than it has ever been.

I also managed to substantially improve the performance of the game by addressing a serious bottleneck. I managed to bring the framerate of the game (at a hypothetical complex configuration) from a consistent 25-30 FPS to a reasonably consistent 60 FPS, all by making optimizations to BlobDistributor.

While the game in its current state is far from perfect, I think it's in a reasonable enough state for presentation.

Retrospective

What went well?

- I managed to address every single task in the sprint backlog, either completing it, discovering that it was already done, or removing it from consideration in the final build. That has not happened up to this point. While this could mean that I underbudgeted my time, it also means that I addressed in one form or another pretty much every concern that arose.
- I managed to address some of the cross-platform compatability issues that arose when trying to deploy the game as a WebGL application, something I haven't had to do before. I ended up applying alternate code to FileSystemLiaison, which made me very happy that I decided to separate that module from other components of the codebase.
- I decided that, in order to ensure consistent quality, I needed to focus my deployment efforts to a single platform. While I'm glad that I had to grapple with HTTP requests and asynchronous loading, the considerable differences between a standalone executable and a web application (persistence issues, the browser environment, performance, inconsistent input models) meant that putting too much effort into the web application would've compromised the quality of my game in other areas. Better to deploy a good product on a single platform than a mediocre one on several.
- I did serious performance and efficiency work for perhaps the first time in my life, and I did so in a guided and data-driven way. I used profiling tools to find the critical pieces of code that were slowing down the game, figured out why they were running so slowly, and then made changes to the logic in order to vastly improve efficiency. This is an indication not just that I can perform optimization tasks, but that my original plan (build for comprehensibility until performance becomes an issue, then optimize the critical locations) paid off. There were only really two problematic classes (BlobDistributor and HighwayFactory), which means that all of the inefficient but comprehensible solutions I deployed to other parts of the codebase simply didn't matter. It shows, at the very least, that my workflow was sane.

- I've finally gotten halfway decent at reasoning about map design in the game. It turns out an elaboration of an earlier point (determine the necessary societies and build outwards from there) works acceptably once you get used to it. I was able to craft two fairly interesting and reasonable maps, and was able to perform some guided iteration on their design. They would've benefitted substantially from actual targeted playtesting, I'm sure, but at least I wasn't at a complete loss as to what I needed to do most of the time.
- I did a solid job at reasoning about sound, despite my lack of experience. I managed to use a relatively small number of samples to add a reasonable series of noises to most activities in the game. It went pretty smoothly for a first foray into the world of audio design.

What could be improved?

- Not only did I continue not to playtest the game with other people, I actively refused offers to playtest from my immediate family. I could sit here and try to justify that decision, claiming that the people close to me aren't good testers because they aren't unbiased. I could also claim that the playtests wouldn't do any good because I didn't plan on working on the project anymore. But those are excuses. I should've accepted their feedback, and I didn't. Part of, I think, an ongoing problem that I'll need to address more pointedly in the future. I can't keep avoiding playtests, in this project or any other.
- There were several days in which I cut short my work, particularly the last day where I nixed 5 whole pomodoros. To be fair, there were no tasks in the backlog left for me to address, but I almost certainly could've found new tasks to do if I was looking for them. Being so close to the end of the project seems to have led to something of a breakdown in discipline that I will need to assert more strongly in the future. I don't really have an actionable plan for that assertion other than "schedule more tasks" or "be better," so I suppose I'll have to think on that more.
- There were a huge number of misestimated durations, but over and under, to the point where the estimates may not have been a reasonable tool by which to budget time. I think next time I do task estimation, I need to provide some justification, some reason why I think the task will take as long as it does. Then, when the task inevitably goes over or under that estimation, I can state why it took as long as it did and compare the two answers. That way, I should be able to uncover some of the false assumptions that are leading me to inaccurately assign durations to my tasks.
- My TDD discipline broke down a little bit towards the end of development. I know for a fact there are at least two sections of the codebase that are missing tests, and there are likely many more. To some degree that breakdown of discipline makes sense, since I don't need the tests once the project is done. And a lot of the work I was doing can't reasonably be unit tested in an efficient manner (the user-facing edge of the UI, for instance). Yet that seems like a dangerous habit to get into. I need to more intelligently assert the test-driven paradigm for core modules, especially when those modules are subject to change.
- I completely forgot to communicate the controls of the game to players. I'll need to go back at some point in the future and add a controls page to things. I also forgot to clean up the launcher (adding a logo and changing my company's name, for instance). There are still tasks that I need to address, then.