BSc in Computer Science

Starborne Map Editor

Hallgrímur Snær Andrésson
Hjörtur Jóhann Vignisson
Mikael Sigmundsson

May, 2020
1. The Team 2
2. The Company: SolidClouds 2
3. The Project: Starborne Map Editor 3
4. Work Agreement 4
   Predetermined Hours 4
   Work organization 4
   Tech Stack 4
5. Work Schedule 5
   Original Schedule 5
   Final Schedule 6
6. Risk Analysis 7
7. Design Draft 9
   UI Paper prototype 9
   UI redesign 10
8. Changes due to the pandemic (COVID-19) 11
   The situation 11
   Work agreement 11
   Extra Availability 11
9. Progress Overview 12
   Sprint 0: Animal Crossing 12
   Sprint 1: Bayonetta 14
   Sprint 2: Cultist simulator 16
   Sprint 3: Doom Eternal 18
   Sprint 4: Elite Dangerous 20
   Sprint 5: Ketchup 22
   Sprint 6: Frostpunk 24
   Sprint 7: Grim Fandango 26
   Sprint 8: Half-life 28
10. Performance Improvements 30
11. Conclusion 31
12. Future work 32
   Postponed Tasks 32
   Research Grant 32
13. References 33
1. The Team

The project team consists of all 3rd year students at Reykjavík University, studying computer science. The members had never worked on anything together before this project, but all shared a common interest in video games and video game development.

2. The Company: SolidClouds

Solid Clouds is an Icelandic start-up founded in 2013. They have been working hard on their only project so far, Starborne, since they started the company. As of 2020, they have around 20 employees at their office in Eiðistorg.

Starborne is an MMORTS (Massively multiplayer online real-time strategy) game set in space. In the game, you build stations and gather resources, with the eventual goal of teaming up with other players in order to control a dyson sphere, which ends the game. A single match can take up to two months to finish. The game has been in alpha since 2017 and went into open beta in April 2020. Currently the game has about 7000 active users.
3. The Project: Starborne Map Editor

For this project, we created a map editor in Unity3D for the game Starborne. The map editor allows users to easily place, edit, and rearrange map objects onto a hex grid. Objects can be grouped together into an area called templates that can also be manipulated. The end goal of the project is to have a sleek map creation tool, capable of easily prototyping different map types and constructing a finished map.

The goal of the project was to create

- A large-scale hex grid visualizer
- Sophisticated tooling for placement of map objects
- Easily digestible information about map statistics

SolidClouds provided textures and descriptions on how the map is represented in JSON. The user interface (UI), user experience (UX) design, and implementation were up to us. Þorgeir Auðunn Karlsson was our Solid Clouds company contact and product owner. Kyle Smith was the potential user for the map editor so a lot of the design was based around his preference. The staff at Solid Clouds helped us immensely throughout the project and we are very grateful for their valuable assistance.
4. Work Agreement

Predetermined Hours

12 week time period (pre COVID-19)
- Mondays: ~15:00 - 18:00 at the Solid Clouds offices
- Wednesdays: 11:00 - 18:00 at the Solid Clouds offices
- Fridays: 12:30 - 13:30, instructor meeting at HR

12 week time period (during COVID-19)
- Work from home: No set time, aimed for 40 hours of work each week

3 week time period
- Weekdays: 11:00 - 19:00 at the Solid Clouds office

Team contribution
- Hallgrímur Snær Andrésson: 304 (hrs)
- Hjörður Jóhann Vignisson: 345 (hrs)
- Mikael Sigmundsson: 305 (hrs)

Work organization

We used “ScrumBut”, which consists of using Scrum, but changing some things. We had an online task board (Trello) which kept track of Solid Clouds product requirements, as well as our product and sprint backlog, which was made from said requirements, using “planning poker” for story points, as was recommended by Solid Clouds staff.

For the duration of the initial 12 week period, we used 2 week long sprints. We then switched over to 1 week sprints for the remaining 3 week period. Each sprint contained designated tasks and goals from our product backlog.

At the end of each sprint we conducted retrospective meetings to improve our workflow and to organize our next sprints. These meetings were occasionally attended by Solid Clouds staff, to give us feedback on what we had been working on. We used a burn-up chart to track project progression using the points from planning poker as a metric.

Tech Stack
- Unity, for the application building
- Visual Studio or VS Code, for script editing
- Git and GitHub, for version control
5. Work Schedule

Original Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Sprint</th>
<th>Tasks</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>Build “product backlog” with functionality and UI design in mind.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Break down product backlog into individual tasks, paper prototype and create reports for the first project status meeting.</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Project setup, Hexagon visualizer, hexagon object class and object placement. Object rotation, JSON export and template creation.</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>Camera movement and zoom, basic statistics. Start implementing main map creation, template picking UI.</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>Template moving, export to file. JSON Serialization, Save project functionality.</td>
<td>25*</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>Load map from file, Rotate templates Template copy-paste, Multi-template select.</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>Template tagging system, random hex objects with weights. Template sector tagging, zoomed out view shows sectors. Export and load files for saved templates.</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>Intricate UI with searching and thumbnails. Overview menu, background image selection, autosave.</td>
<td>~25*</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>Prepare for final hand in (User guide book, final report etc).</td>
<td>∞</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>Final Presentation.</td>
<td>∞</td>
</tr>
</tbody>
</table>

Table 1: Original work schedule estimate

* hours highly expected to change as we make progress.

In order to be able to meet all requirements we had to determine how many points to aim for per sprint to ensure completion. We decided that each story point would correspond to roughly 2 hours of work. This meant we aimed for ~24 points per sprint for the first 4 sprints. We expected this estimate to increase when we had more time at the end of the semester that could be dedicated to the project.
Final Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Sprint</th>
<th>Tasks</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>Break down product backlog into individual tasks, paper prototype.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create reports for the first project status meeting.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>Project setup. Hexagon visualizer. Hexagon object class.</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>More hexagon visualization work. Camera movement and zoom.</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Individual hexagon selection.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>First UI implementation, including map and template views. Basic statistics.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial JSON serialization implementation.</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More JSON serialization.</td>
<td>22</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>Multi object selection. Object rotation. Color selection.</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More JSON serialization.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>Copy/Paste functionality. Random object functionality. Template tagging system.</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>Final texture implementation. Initial heatmap statistics implementation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>More JSON serialization.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>Finalized heatmap. UI redesign and implementation.</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>Final sector functionality implementation.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>Sector functionality finalized. More new UI implementations. Bug fixes.</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>Final UI touches. Bug fixes. User manual and final report finalization.</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: Final work schedule breakdown

When the two tables are compared, we can see that quite a few things changed. The biggest change is that JSON serialization was not something we could finalize in a single sprint, but rather, it was something that had to be implemented gradually as the project evolved.

Another significant change that was necessary regarded the statistics implementation. Initially we started implementing counters for each type of item on the map, but then, after getting some feedback, we realized that a resource heatmap was going to be far more useful for the applications user.

We always knew, and were prepared for, the initial plan to change significantly. We are pretty happy with our initial estimates and we feel we are much better prepared to organize another project of this scale in the future.
6. Risk Analysis

<table>
<thead>
<tr>
<th>Incident</th>
<th>Prevention</th>
<th>Solution</th>
<th>Impact</th>
<th>Probability</th>
<th>X factor</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect time estimation for project tasks</td>
<td>Estimate time realistically but over-estimate the time a task will take.</td>
<td>Redo all time estimation</td>
<td>4</td>
<td>3</td>
<td>12</td>
<td>Hjörtur</td>
</tr>
<tr>
<td>An A level task takes longer than anticipated and as a result we run out of time</td>
<td>Plan well ahead and make sure to split big tasks as much as possible</td>
<td>Keep track of the project progress and make sure we’re as close to the schedule as possible</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>Mikael</td>
</tr>
<tr>
<td>Personal matters that result in less time for work.</td>
<td>Schedule around the assumption that lost days will occur.</td>
<td>Maintain an environment that will make it easy for the sick person to catch up when they return.</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>Mikael</td>
</tr>
<tr>
<td>Our application can’t support a large enough map</td>
<td>Focus on optimization from the get go while keeping time constraints in mind.</td>
<td>Ask Þorgeir (Tech Lead at SolidClouds) for help with optimization.</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>Hjörtur</td>
</tr>
<tr>
<td>JSON file is corrupted or incorrect resulting in crash and/or potential data loss.</td>
<td>Use established JSON formats and be careful with serializations.</td>
<td>Try and recover what was lost through Unity’s auto-save functionality and don’t use this JSON file again.</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>Hallgrímur</td>
</tr>
<tr>
<td>Unity crashes while editing a scene.</td>
<td>Save regularly (at least every 5 minutes).</td>
<td>Try to recover scenes from Unity’s auto-save functionality.</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>Hjörtur</td>
</tr>
</tbody>
</table>
A personal computer breaks down. | Be careful with our laptops. | Borrow a laptop or use a home computer to work from home. | 4 | 2 | 8 | Hjörtur

Overestimation of Unity know-how | Practice working with Unity and gather study materials online | Get help from instructor and SolidClouds contact | 3 | 2 | 6 | Mikael

Loss of a team member | Maintain good communication between members | Reduce project scope. | 5 | 1 | 5 | Hallgrímur

Github goes down. | Keep local git repo up to date. | USB drives or Google drive to share local repos. | 1 | 1 | 1 | Mikael

Transport failure. | Hallgrímur takes care of his car. | Take the bus. | 1 | 1 | 1 | Hallgrímur

The project gets corrupted | Push to git frequently | Pull old version from github | 1 | 1 | 1 | Hallgrímur

Table 3: Risk analysis chart

Some of the risks we predicted came to fruition. As expected, our initial time estimations were too optimistic, and we needed additional time to work on some functionality. We noticed this before it became too difficult to solve, and updated our backlog to better match our new reality.

One of the team members' laptop screen broke. We were able to solve that by asking Solid Clouds if they had any extra screens laying around the office, which luckily, they did.

Finally, during our commute on one of our last days at the Solid Clouds office, we blew a tire and had to go to a mechanic. We weren’t very far from our houses, so we decided to work from home that day instead of risking more damage driving on a temporary replacement tire.
7. Design Draft

UI Paper prototype

Image 1: Paper draft of our UI design

This was our first draft of the UI design we had in mind. Initially it was one window with all the functionality, but after discussing this with the user of the application we found out that we needed at least one or two other windows that we planned on putting into tabs. At this point we determined that we were on a solid trajectory in terms of our UX development. The tool that was being used at this time was fairly limited and hard to use.

The two main windows are “Template View” and “Main Map View”. The template view is used to create templates that the user can use on the main map view to place down many hexes with ease. Once placed, these templates can be moved, rotated, copied, and re-randomized. Each hex in these templates has a set of rules, dictating what can spawn there, implemented as weights.
UI redesign

Later on in the project, we got to the point where we could start rethinking our UI to be more user friendly. We stayed close to the original design with a few modifications. The most significant changes were to the two top menu bars, one for actions and one for tools. This UI design was presented to a few staff at Solid Clouds, including the person who would likely use the map editor the most. They approved the layout and liked the functionality it showed, so we moved on to implementing this design.
8. Changes due to the pandemic (COVID-19)

The situation
COVID-19 broke out during our third sprint, Doom Eternal, and a lot of our work methods changed drastically. The virus did not directly affect us, but on the 17th of March, one of us potentially had the virus and from there on, we worked solely from home until the 4th of May. Not due to us contracting the virus, but rather we had no interest in catching it and potentially infecting staff at Solid Clouds.

Work agreement
The most significant change was that we now worked from home on Mondays and Wednesdays instead of being able to meet in person at the University. Later on, as other courses finished up sooner than we had anticipated due to the virus, we were able to work on the project daily, with a goal of 8 hours of work per weekday.

Extra Availability
After the fifth sprint we originally planned to take a break for the Easter holiday and then focus on exams, however, due to the pandemic the team members no longer had exams, or any extra obligations. We decided to use the time to conduct an extra long sprint to make up for lost work prior in order to better meet our deadlines.
9. Progress Overview

Sprint 0: Animal Crossing

Sprint Burn-Up
There was no burn up for sprint 0 since there were no official stories or tasks worked on.

Sprint Overview

Project definition, setup, presentation and initial documentation.

- Attended some meetings over the course of the sprint with various Solid Clouds employees to further define and understand the project requirements
- Created a draft of the application’s UI design
- Set up our task tracking manager (Trello) which included:
  - Reviewing the requirements list we received from SolidClouds
  - Creating a product backlog
  - Breaking the backlog into A/B/C tasks
  - Using planning poker to assign points to each task

![Image 3: Original trello layout after creating backlog](image)

- Worked on the first draft report, which we handed in to our instructor
- Put together tables and set up our work environment at the SolidClouds office
- Worked on a presentation for students and course teachers
- We had an extremely productive meeting with Kyle, who is the level designer at SolidClouds and our end user, where we discussed and he approved of our UI design and further explained some technical aspects of the products requirements
- Worked on a project status meeting report
Status Image

Here is a little sneak peak at what we had been able to do in Unity at this point. We got the Hexagon grid system implemented almost perfectly.

Image 4: First status image of the project

Sprint Retrospective

We were overall happy with the progress made this sprint.

Things we felt went well:
- The meetings were really helpful, the information we gathered about the project helped considerably, especially the last meeting we had this sprint with Kyle
- We were glad to find that everyone on the team worked well together. Everyone was contributing equally to the project. Nobody was afraid to disagree with anyone else and we all respected each other’s opinions

Things we felt need to be improved:
- We all agreed that we needed to find more hours to work on the project individually, which we were confident we would be able to do now that we had individual tasks broken down and could start to work somewhat separately with a well defined common vision for the project
**Sprint 1: Bayonetta**

**Sprint Burn-Up**

![Chart 1: Burn up draft showing the first sprint progress](image)

“**Total A**” (orange) shows the total point sum of the A requirements. When this number increases, it means we’ve added a task to the A requirements. “**Accumulated**” (blue) shows the amount of completed points so far.

**Sprint Overview**

This was our first official sprint, meaning there was a lot of setup involved. Once everyone was up and running properly, we could finally begin!

Our main focus this sprint:

- All members created their own scene to work in, so we wouldn’t get merge conflicts on a scene
- Started work on the hexagon visualizer and exploring techniques to be able to render up to a few million hexes in the scene at a time
- Implemented the hexagon object that represents each hexagon on the map
- Started working on the camera movement and zoom
- Hex selection was implemented
Status Image

![Image 5: 650,000 hexes, all generated with the same texture](image)

**Sprint Retrospective**

Overall a decent sprint and we managed to exactly complete all planned story points.

**Things we felt went well:**
- Getting the project to render around 10 million hexes went way better than expected, and could even keep a high frame rate with that many hexes

**Things we felt need to be improved:**
- There was a lack of code rules, we resolved by adding a task each sprint to clean up and make sure code was readable to anyone
- The dragging was not working correctly at the end of sprint, for such a tedious and complicated class we should’ve used pair programming to keep the consistency of the code better
**Sprint 2: Cultist Simulator**

**Sprint Burn-Up**

**Accumulated and Total**

![Burn-up chart diagram](chart2.png)

Chart 2: Burn up chart after sprint two with estimates.

**Total A**: How many story points there are in our list of A requirements.

**Accumulated**: How many story points we have completed.
Sprint Overview

Making things look better and started defining more functionality for our backend.

Main focus was the following:

- Basic statistics that only include counting instances of resources placed on the map
- Started implementing the “main map creation view”
- Set up the first draft of our UI
- Fixed the camera drag
- Started setting up our JSON Serialization to be able to save things persistently
- Export templates to file

Status Image

![Image 6: First UI design and smoother map layout.]

Sprint retrospective

Overall quite happy with our progress even though we are a little behind.

Things we felt went well:

- We managed to increase the time we spend on the project overall like we wanted

Things we felt need to be improved:

- The biggest issue was that we all felt we needed to split up our sprint backlog into more detailed and separated tasks, so that's what we plan on doing at the start of the next sprint
Sprint 3: Doom Eternal

Sprint Overview

At this point we started working on creating the map by placing down templates. We discovered a lot of bugs, squashed a lot of bugs, made some really nice progress towards the final feel of the application.

Because of the COVID-19 outbreak, we moved towards working more at home instead of at the Solid Clouds office. We had a feeling this might affect our output, since working from home is a bit more difficult, both concentration and coordination wise.

Main focus was the following:
- Reorganized our product and sprint backlogs
- Refactored big classes to improve manageability
- Saving templates
- Placing templates on the main map
- Moving placed templates around
- Fixes to the shader and implementing hover states
Status Image

Image 7: Placing of template and moving

Sprint Retrospective

We were still a little bit behind, but quite happy about our overall progress.

Things we felt went well:
- We felt like working in pairs was excellent, when needed
- Going over the tasks for the sprint and reorganizing/redefining at the start of the sprint went really well, and we decided to do it at the start of every sprint from now on
- We finished most of the things we set out to finish, even with some added things that popped up

Things we felt need to be improved:
- We hadn’t started on the user manual yet
- We were slightly behind schedule when we looked at the existing tasks, but not if we look at the absolute number of points completed
Sprint 4: Elite Dangerous

Sprint Overview
For this sprint, we completed much of the core functionality of the project. Most of the vital tasks were done. We discovered a lot of important tasks that had to be included for the tool to be usable.

COVID-19 had more of an affect than anticipated. With the school being closed and workplaces encouraging people to work from home, we were pretty much stuck at home in isolation. Working together from home turned out to be tough on morale and productivity.

Main focus was the following:
- Hover States (when moving things)
- Rotating Templates
- Choose color for templates
- Move templates around
- Select more than one template at a time
Status Image

Image 8: Rotating templates

Sprint Retrospective

Optimistic about the future, happy with project status, but morale is low due to the need to work at home.

Things we felt went well:
- Core functionality almost finished
- Breaking down tasks helped a lot, but could be broken down further

Things we felt need to be improved:
- Working from home sucks
- Morale is low
- Still behind, we foresaw being able to catch up over the next few weeks, since we would end up having time we didn’t initially account on having
- A few of the older tasks needed to be revisited because of poor planning
Sprint Overview

This was never intended to be such a productive time period. We all assumed we would be busy with final exams and other courses. However, due to unforeseen circumstances, finals and other projects were effectively canceled, and gave us more time to work on this project.

Originally, our plan was to finish any outstanding tasks that leaked over from previous sprints, and maybe finish off a few other tasks. We ended up almost completing our entire backlog in this time period and were feeling quite happy about it.

Main focus was the following:
- Copy/paste for hexagons and templates
- Random object support
- Loading textures from the texture folder
- Basic Heatmap
- A large amount of bug fixes and random handy things implemented
Status Image

Image 9: New textures for hexes.

Sprint Retrospective

Fixed a ton of bugs and got well ahead of our schedule.

Things we felt went well:
- Project progress
- Bug squashing
- Improved the look of the project because we got actual assets

Things we felt need to be improved:
- Communication
- Task distribution
- Task prioritization
Sprint 6: Frostpunk

Sprint Burn-Up

Accumulated and Total

Chart 6: Most big features were done, nearing the finish

Sprint Overview

Main focus was the following:

- Resource heatmap
- UI Redesign
- Sector implementation

The resource heatmap was something Solid Clouds had wanted in their map editor for a long time. It helps them visually see where on the map resources are tightly packed, which in turn, helps them balance gameplay.

At this point we redesigned the UI, the original was only implemented for functionality while creating the backend. This new design however, is much more focused on usability and we designed it with input from our core user.
Image 10: Heatmap shows the distribution of resources on the map.

Sprint Retrospective

Project scale was starting to show its effects.

Things we felt went well:
- Coding rules made it easier to implement new stuff based on existing classes. It was now very easy to see if a class had the necessary functionality or if it doesn’t, we could implement it quickly
- We finally got pretty good at splitting tasks up in a way that everyone had something to do at all times

Things we felt need to be improved:
- The project became big enough that tracking down bugs took longer and longer
- Disciplined work from home was hard
Sprint 7: Grim Fandango

Sprint Burn-Up

Chart 7: All A tasks are completed

Sprint Overview

Main focus was the following:

- Sectors
- Refactor requests from user
- Summarized testing

With the implementation of sectors finished, all core features from Solid Clouds and all requested features from the user had been completed.

Now, after thoroughly testing all features individually, we tested them in conjunction with every other feature and every combination possible. We found some oversights that we were able to fix without any problems, mostly due to the code structure of our project.

We now have a very rich minimal viable product. We have yet to test whether Solid Clouds can use exports from our editor. It is promising though, since they mentioned early on that they would be eager to change their own map file structure to fit ours if we come up with a good one.
Status Image

Image 11: Sector view of 4 sectors and 12 templates.

Sprint Retrospective

Added the new UI and combined and tested all features together.

Things we felt went well:
  - Meeting up to work is much more productive
  - Status meeting showed us that our hard work has paid off and that we have done an amazing job

Things we felt need to be improved:
  - Stress management as we felt pressure to finalize our project
  - Team communication due to having to work remotely
**Sprint 8: Half-life**

**Sprint Burn-Up**

![Burn-Up Chart](chart.png)

*Chart 8: No real tasks were worked on, only new tasks were quality of life issues.*

**Sprint Overview**

Main focus was the following:

- Exit button
- Keys for heatmap resources
- User manual
- Final report

We wrapped up quite a few things that had been low on our priority list. We added a UI button for the backend functionality of picking which resource the heatmap was displaying and added a button to close the application. The rest of the sprint was focused on the deliverables for the project hand-in.
Status Image

Image 12: Sector view combined with the heatmap with every resource toggled on.

Sprint Retrospective

Today was a good day, all finished up, we’re happy.

Things we felt went well:
  - Working at the Solid Clouds office is still very productive
  - A lot of feedback from Solid Clouds employees this sprint

Things we felt need to be improved:
  - Nothing, everything is working as it should, we’re about to be done!
10. Performance Improvements

From the very start of the project, we knew performance on a large scale was a strict requirement. For Solid Clouds, a "large scale hex grid visualizer" meant that we needed to be able to support displaying at least 1 million hexagons, each with a different image on it.

To accomplish this we needed to write our own shader, this was due to how Unity handles batch jobs. This shader started out really simple: just showing the color of the hexagon. Once we had that going, we needed to add textures. To accomplish this, we wrote a “compute shader” on top of our regular shader. The compute shader takes data from our application and makes changes to the normal shader, as needed. With this, our shaders were done. We can easily show more than a million hexagons without breaking a sweat, even on relatively low powered computers.

This method still has plenty of room for improvement. We aren’t using the compute shader for its intended purpose. It’s intended to do calculations, as that’s what GPUs are excellent at, but we are using it for a lot of branching if-statements. Since the compute shader has to iterate through all of the hexagons we are drawing, it takes longer and longer for it to update just one hexagon. Showing more than a million hexagon is an achievement nonetheless.
11. Conclusion

With our first big, practical project finished, we have gained an understanding of the hardships and intricacies that come with being a computer scientist. We know that careful planning and team management can make a difficult project flow seamlessly. It doesn't make it a simple task, but it sets the project up to be completed in a reasonable amount of time with minimal conflict.

Our team composition and teamwork was excellent. We all quickly learned to trust each other with tasks and often asked each other for help, even if it was just to listen to the other person's thoughts. Pair programming was frequent, since it was vital not to get side tracked while writing code, this became increasingly important due to working from home. There were some typical disagreements during the project, but none were abnormal for this type of project and everyone's desire to create high quality work. Each issue was resolved relatively quickly and group morale was unaffected in this regard.

COVID-19 caused productivity issues mostly due to the need to adjust to working from home, as it is not the most conducive environment for learning.

We are proud of the functionality we were able to implement. We were able to support the displaying of millions of hexagons through our use of custom shaders and rendering methods. The application’s tooling supports a variety of sophisticated operations, such as moving, rotating, copying and pasting. Our application offers a heatmap for the Solid Clouds team to strategically construct a balanced map.

The final product was a huge success and very well received by Solid Clouds. It’s likely never going to reach a finished state, since you can always add features to editors such as this one. Regardless, in respect to this project it is finished, completed and perfect.
12. Future work

Postponed Tasks

We had some tasks that would be classified as A task, but we decided not to include them in this project, since they would either slow down the project too much or would take too long to implement. Given a chance these would be the next main tasks we would do.

1. Undo/Redo - this is a very difficult task given the size of the project, we would have needed to calculate deltas and keep them in a stack
2. Pre-placement brush - show where you are trying to place something, similar to how we show when you move things around
3. Project save file optimization - set up a database structure for the map, so we don't save excess data. Unnecessary for the current project, but offers better scalability
4. Map statistics window - originally planned since we thought it was a core requirement from Solid Clouds, but it turned out the heatmap was way more important
5. Changing project preferences directly in the editor - right now, settings are changed through editing files. This would have taken a lot of time and is not very good usage of our time, since you can change these settings in a different way

Research Grant

We plan on continuing to work with Solid Clouds through a research grant from Rannís. We want to bring our, and their, vision of the perfect map editor to life.
13. References

“Drawing thousands of meshes with DrawMeshInstanced/Indirect in Unity,”

toqozz.github.io, 01-Nov-2019. [Online]. Available:


“Hexagonal Grids,” Red Blob Games, Mar-2013. [Online]. Available:


