Final Report of Group BLOCKS7

Block Model Compression Algorithm

A1806297 Po-Yi Lee

Table of Content

Project Vision Reflection

According to the initial report, the customer's requirements in this project is to develop an algorithm that can compress the block model as part of client's (Maptek) pipeline and reduce the amount of storage and bandwidth. In the end of the project, we have designed four algorithms and completed two versions of the compression algorithm including sameDomain and GreedyExpander which reduces the number of times the data has to be revisited, whilst also allowing for reasonably good compression to satisfy the requirements.

Since we implemented the compression algorithm using Python rather than C language which could significantly affect the running time of the compression, we developed the methods with a heavier weighting towards compression. Even though our two completed algorithms are not an optimal algorithm which is basically the customer's expectation, our team has passed three of five datasets and have a good compression ratio performance in the competition including intro_one (93.7378%), fast_one (79.6388%) and stratal_one (98.9630%).

Customer Q&A Reflection

In the beginning of the project, we are required to develop a program which can take uncompressed input data on standard input and produce compressed output data on standard output with no loss but we had no experience with this kind of problem. Therefore, we asked the customer if we should just find the researched methods such as Pruning, Quantization, Low-rank and approximation and sparsity etc or better to implement the algorithm from scratch.

The customer answered that in terms of our "literature review" on the research into various methods that might be useful for this problem, this is a broad topic that has no right answer. He said if there is a great paper out there that presents optimal approaches to this problem, then he does not know about it. But even a paper that presents a good algorithm is only a start with this problem. To Implement the algorithm efficiently to run fast, choice of data structures, choice of language and platform, use of hardware resources such as multi-threading or GPU or maybe trade-offs between RAM and disk storage might all come into play and are unlikely to be well discussed in a paper that just presents an algorithm. This is an equally important part of this challenge than just coming up with a smart algorithm. A sub-optimal algorithm written well and fast will outperform an optimal algorithm in this challenge. Therefore, chasing the perfect algorithm is not advised. He suggested that we would start with writing our own implementation from scratch just to get a feel for the problem before going out and looking at the research. Then our experiences could guide what research papers might be useful based on what we have learnt writing our first few implementations.

According to the customer's suggestion, we had a good start to research our algorithm methods by writing our own implementation from scratch. We got more familiar with what the actual problem would be and successfully implemented multiprocessing methods to improve the speed problems. For me, this question helped the team a lot to understand how the 3D model compression could work, plan the subteam and made it easier to assign the tasks to the team.

There are five client's meeting during this project:

- the kickoff and sprint 1 planning meeting on 5th August, 2021 with the tutors, Aryaman Dhawan and Will Reid:
- the sprint 1 review and sprint 2 planning meeting on 26th August, 2021 with the tutors, Aryaman Dhawan and Will Reid.
- the sprint 2 review and sprint 3 planning meeting on 10th September, 2021 with the tutors, Aryaman Dhawan and Will Reid.
- the sprint 3 review and sprint 4 planning meeting on 8th October, 2021 with the tutors, Aryaman Dhawan and Will Reid.
- the sprint 4 review and sprint 5 planning meeting on 22th October, 2021 with the tutors, Aryaman Dhawan and Will Reid.

In the beginning of the project, we received the customer's requirements including developing a 3D model compression algorithm from the datasets in the meeting. We asked if the requirements specification with the submitted solution is to take standard input and produce the result on standard output and get a clearer feedback for our solution. However, in the following sprints, I felt more confused with how the software went because I had never written this kind of algorithm problem and met some technical problem when I managed to write the code. In the sprint 2 and 3 planning meeting, I was very silent during the meeting since the new requirements or datasets came out but my technical knowledge was not good enough to analyze the problems and asked about related questions immediately.

Then, after the discussion with the team and especially Karl, I kept asking how the block could be cut and divided for the algorithm and finally understood it more clearly. During the last two meetings, we asked more questions about what problem we met such as the clarification of what order blocks needed to be output, how many threads are available in the software and so on and I felt more confident with this project. Having these meeting experiences with the client, if there are meetings for the future project, I will definitely do more research, discuss with my team in advance, predict what potential requirements could be released and be ready to ask more related questions to my clients.

Users and User Stories

This program is designed in response to Maptek's requirements for the Block Model Compression Algorithms. Users of this program can be divided into the following roles:

- Senior software engineers (developer): Firstly, they are responsible for developing and maintaining a large-scale codebase of software. Secondly, they should integrate the whole software. In addition, they should complete tasks including 3D modelling, spatial analysis and design technology for the global mining industry.
- Junior software Engineers (researcher): They need to have some understanding of existing code. In addition, they should understand the logic and algorithm behind it. Their tasks could help optimise and refactor current codes to improve the compression ratio and efficiency.
- Data scientists: They are responsible for data collecting and providing the datasets to their senior software engineers and junior software engineers. These datasets could make it easier for software engineers to design model compression algorithms. They could also manage the backend of the testing platform for their software engineer.

For senior software engineers (developer):

- Proving of the optimal compression algorithm: find and prove the optimal compression rate for the 2*2*2 dataset so that we can show that the optimality should be possible with a "brute force" proof.
- Acceptable compression speed but lossless data : The compressor script is unnecessary to be compressed with high speed. But it is important to keep the compressed data lossless.
- Running code on the GPU: implement the compression algorithms running on a GPU that can complete more work in the same amount of time as compared to a CPU. So, it will increase the throughput of data and the number of concurrent calculations within an application.

For junior software Engineers (researcher):

- Basic function of block algorithm: Achieve the basic compression algorithm to help our client save time and bandwidth. The code of the compressor should be accessible and readable so that I could understand how it works.
- The principle of code running on the GPU: 3D visualization: to visualize the compressed result to see how the algorithm works.
- Researching how the CPU implements: collect the researched papers about the GPU and send them to the senior engineer.

For Data scientists:

- Raw datasets collection: collect the raw datasets to provide the other software engineers to test their algorithms.

- Data cleaning and preprocessing: preprocess the block data into the CSV file and make sure the datasets are usable.
- Compressed data collection: collect the compressed dataset and send it to the visualization part.

By reviewing the initial reports, we found that the actual user population and their stories might be somewhat different from the user stories we mentioned in the initial report. The initial user story focused on the results of compression, which is how the output data and visualizations will be used once we compress the data successfully. So we mentioned people in the mining industry, including geologists and reservoir Engineers. We believed that they are interested in predicting and understanding the physical way that the Earthworks by feeding in drilling or other sampling data and obtaining domain or grade models. Therefore, efficient algorithms can help them save time and bandwidth if the model processing time can be reduced and the compression rate remains well.

But in fact, in the later stage, we thought that the algorithm implementation we were mainly responsible for was more technical, that is, our main users should be searchers responsible for software development and the developers who implemented the code in the company. Through the code design, we can help developers to achieve the compression of these data, and have a good balance between compression rate and compression time, and put forward some possible solutions for the code running on the GPU. The documentation we wrote helped researchers better understand how compression works and how code runs on the GPU.

Software Architecture

In this section, I will illustrate the final architecture of our software.

Figure 1 shows how the input block data from the client goes through in the software. The input block data in the CSV file will be read and parsed. The module will read 1 parent block high slice at a time and encoded into an internal representation. It then creates Block objects for each line on the input. These Blocks are then collected into ParentBlocks. The position of the Blocks is relative to the start of the ParentBlock. After that, ParentBlcoks will be sent into the compression engine which is responsible for deciding which algorithm is applied to which ParentBlocks. The algorithms include sameDomain, GreedyExpander and MaxBlock. This architecture allows for different algorithms to be switched in and out, and adds the possibility of running different algorithms for different types of ParentBlocks. Then the compressed data is written to the output file (stdout) in CSV format once all the ParentBlocks are processed.

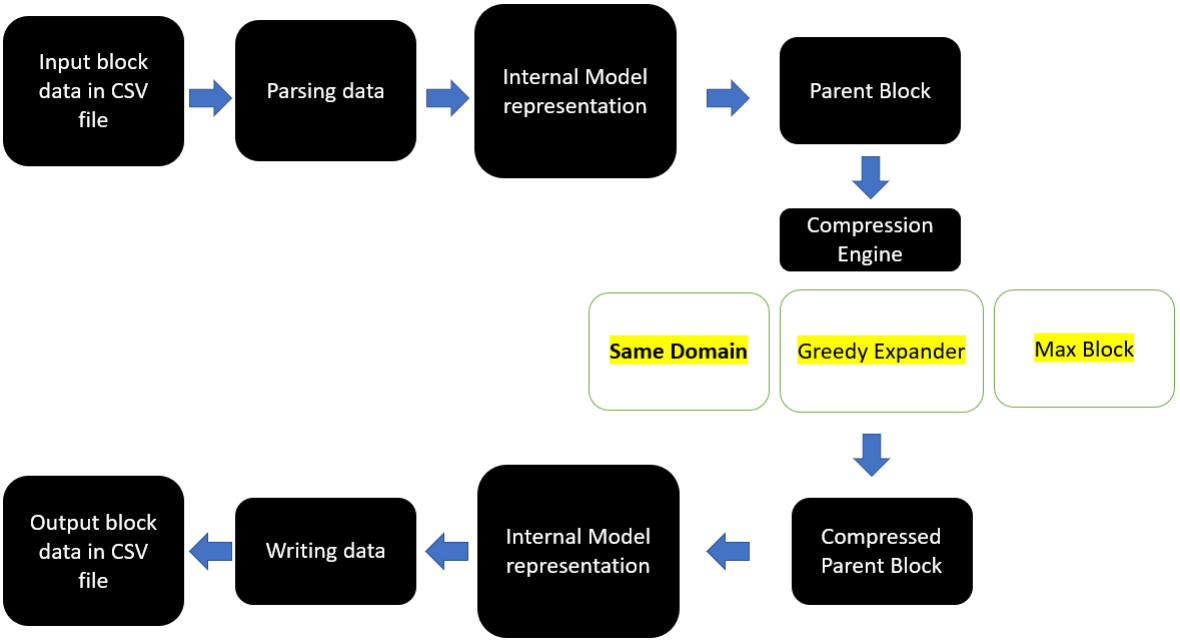


Figure 1. The diagram of input block data flow

There are three algorithms for block model compression including sameDomain, GreedyExpander and MaxBlock:

Firstly, sameDomain algorithm is simple to understand. It goes through the parentBlock data and checks if all of the blocks belong to the same domain. If so it will merge all of the blocks.

Secondly, GreedyExpander starts at the (0,0,0) location and determines how many blocks it can merge with, in each of the x, y, and z directions. It then merges with as many blocks as it could, making an arbitrary decision in case of ties. It merges this block with neighbouring blocks iteratively, until no block can be merged. It then moves on to the next available block.

Last but not least, the incomplete algorithm is MaxBlock which has two versions for the CPU and the GPU. The largest block that can form through merging in the positive x, y, and z directions is calculated for every block location in the parent block. After that, the largest of these blocks will be selected as the parent block. When there are any of the expanded blocks that overlap, the selected blocks are re-calculated and then the next largest block is selected, and the procedure continues in this method until all the blocks of the ParentBlock have been covered.

In conclusion, SameDomain algorithm can check the block quickly and also saves costly computation. GreedyExpander algorithm can reduce the number of times that the data has to be revisited, whilst also allowing for reasonably good compression. MaxBlock should be an optimal algorithm for compression but the code for the CPU and the GPU will be slightly different.

Tech Stack and Standards

We developed this software in Python3 and most of the software was developed in the “backend” since it was used to compress data without direct interaction from a user. The final software only depends on standard library classes, reducing dependencies and maximising portability. For the "frontend” we planned to develop a visualizable model to see how the algorithm operates. However, the visualization operation is still in progress and we can only implement our visualization tool on a small amount of datasets with the Udacity library.

We planned to implement the system in Python including Numpy library and the Numba tool and once we have consistent good compression, we may re-evaluate implementing the system in C since with Python a good compression algorithm can be re-implemented to run faster, but a fast algorithm can’t be easily made to compress better. Rather than using C for any implementation due to the time limitation, we developed a multiprocessing module in Python which can apply our algorithms to the ParentBlocks in parallel to achieve speed increases. In addition, we developed the software framework including development environment and static analysis which can help the algorithm design and the testing of the speed and compression rate.

Development Tools: No IDE was mandated because everyone was comfortable and familiar with their own IDE. The editor that the development team used includes VS Code, Atom editors. Both of these have plugins to assist in Python Development. Python Virtual Environments (venv module) was used to organise all development dependencies. This is built into the Python standard library and easily works across all operating systems.

The flake8 and mccabe static code analysis packages were used. These tools check (“lint”) the code for unused variables etc., code complexity, and ensure the code and comments are written to the Python community standard PEP8. These can be integrated nicely into editors and can give interactive feedback.

Code Standard: We planned that all pull requests for code must be submitted on our Github repository, tested and reviewed by 2 other team members (or 1 in the case of documentation/admin) and static code analysis must pass for the branch to be merged.

The Static Analysis ensures the code is written to PEP8 style standards (what all python code should aim for) which mandates a set of conventions for things such as class/function names, spacing, and comment style, and for other issues such as unused variables, and cyclomatic complexity. Throughout the project, we successfully operated this code standard to allow all the developers to be able to read every developer’s code in the same way.

Communication Tools: Slack was used by team's general discussions, technical questions, and keeping in contact with the PO. Slack was set up by the PO and keeping all team communication in the same place simplifies communicating, and makes it easier to find previous discussion points. Slack also allowed us to communicate asynchronously, especially since each team member is working across multiple time-zones.

Zoom was used every week for the interactive group calls. It was convenient for us to report our own progress and discuss the details of 5 sprint retrospectives and reviews and the next sprint plannings. This software was operated well by all team members. It also allowed screen sharing to allow the team to all see the same thing which was important since most of the team were working remotely.

Group Meetings and Team Member Roles

We set up the meeting schedule based on the meeting schedule below. Our meetings were planned to be held every Wednesday and Sunday from 8:00 pm to 8:30 pm. The upcoming meeting will be held virtually on Zoom.

The upcoming meeting schedule:

Week	Sprint	Meeting Date/ Due Date	Starting Time/ Due time	Note
2	0	5 Aug (Thursday)	20:30	Team meeting
		8 Aug (Sunday)	20:20	Team meeting
3	1	11 Aug (Wednesday)	20:00	Team meeting
		15 Aug (Sunday)	20:00/ 23:59	Team meeting/ Initial Report, Snapshot 1.1
4		18 Aug (Wednesday)	20:00	Team meeting
		22 Aug (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 1.2
5	2	25 Aug (Wednesday)	20:00	Sprint Retrospective 1 meeting
		29 Aug (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 2.1, Sprint Retrospective 1
6		1 Sep (Wednesday)	20:00	Team meeting

		5 Sep (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 2.2
7	3	8 Sep (Wednesday)	20:00	Sprint Retrospective 2 meeting
		12 Sep(Sunday)	20:00/ 23:59	Team meeting/ Snapshot 3.1, Sprint Retrospective 2
8		15 Sep (Wednesday)	20:00	Team meeting
		19 Sep (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 3.2
Break				To be discussed.
Break				To be discussed.
9	4	6 Oct (Wednesday)	20:00	Sprint Retrospective 3 meeting
		10 Oct (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 4.1, Sprint Retrospective 3
10		13 Oct (Wednesday)	20:00	Team meeting
		17 Oct (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 4.2
11	5	20 Oct (Wednesday)	20:00	Sprint Retrospective 4 meeting
		24 Oct (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 5.1, Sprint Retrospective 4
12		27 Oct (Wednesday)	20:00	Team meeting
		31 Oct (Sunday)	20:00/ 23:59	Team meeting/ Snapshot 5.2
13	-	-	-	Final Report Final Presentation

Our team met once or twice per week via Zoom to report our own progress, update the customer's requirements and discuss the task of every sprint. More detail below:

- the sprint 1 retrospective meeting on 25th August, 2021 with the other team members.

- the sprint 2 retrospective meeting on 16th September, 2021 with the other team members.
- the sprint 3 retrospective meeting on 9th October, 2021 with the other team members.
- the sprint 4 retrospective meeting on 24th October, 2021 with the other team members.

In addition to the Zoom meeting, the customers set up an additional feedback channel with every team in Slack, where we can always ask questions about the user story and technical problems we met.

In each sprint, our team had a scrum master scheduled in the table below:

Sprint	Name	Student ID
1	Po-Yi Lee	a1806297
2	Yuanpeng Liu	a1784375
3	Karl Asenstorfer	a1162576
4	Xiaoman Li	a1804817
5	Kaiyang Xue	a1784184

Throughout the project, I considered that we improved the efficiency of our meeting progress.

In the beginning, there was no host to lead the meeting so there was a lot of time that no one spoke. In order to deal with this time-wasting problem, I became an active speaker to talk actively and ask the team what we had to discuss and how their progress worked. Over this project, I found out that It was crucial to be active and being shy was not a good way to improve the efficiency of the meetings. After I led the meetings, our meeting time lasted for approximately 30 minutes rather than 1 hour as before.

In addition, an agenda is a good material to go through the meetings in a more efficient way. As I mentioned above, I started to be more active to speak in the meetings. In order to lead the meeting, I drafted an agenda and shared my screen in advance before the meetings. This really made the team dive into the discussed context more quickly in the meetings.

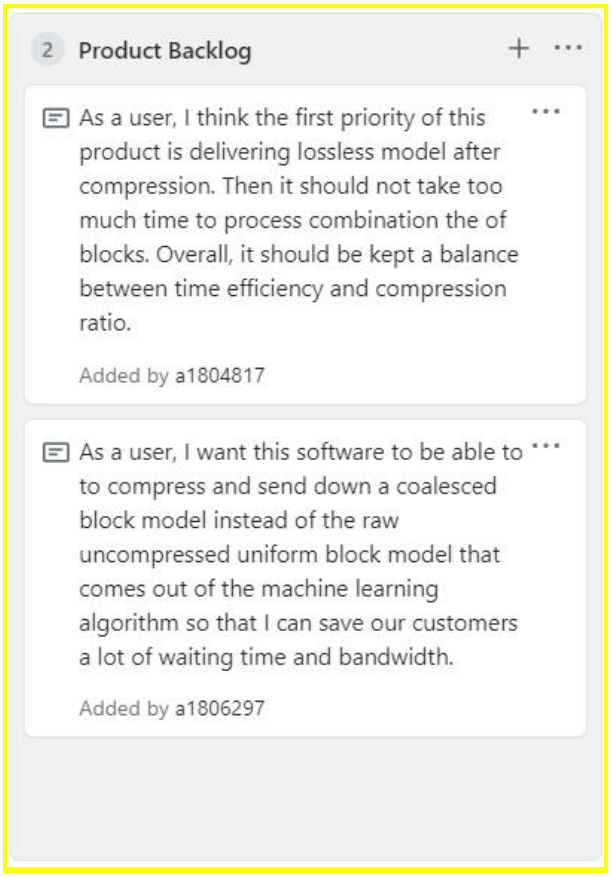
Snapshot

Snapshot Week 2 of Group BLOCKS7PG

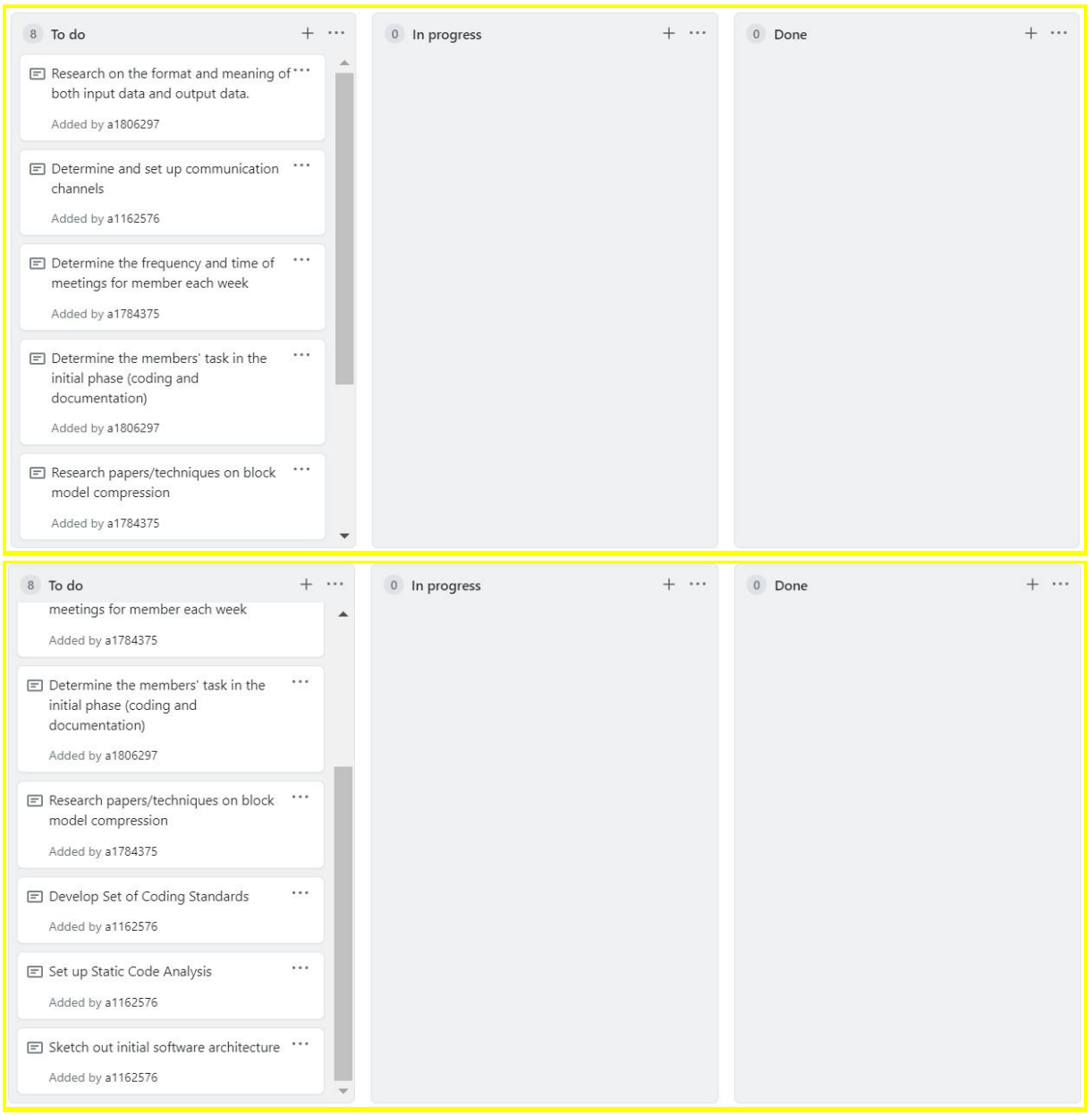
(Snapshot 0)

Product Backlog and Task Board:

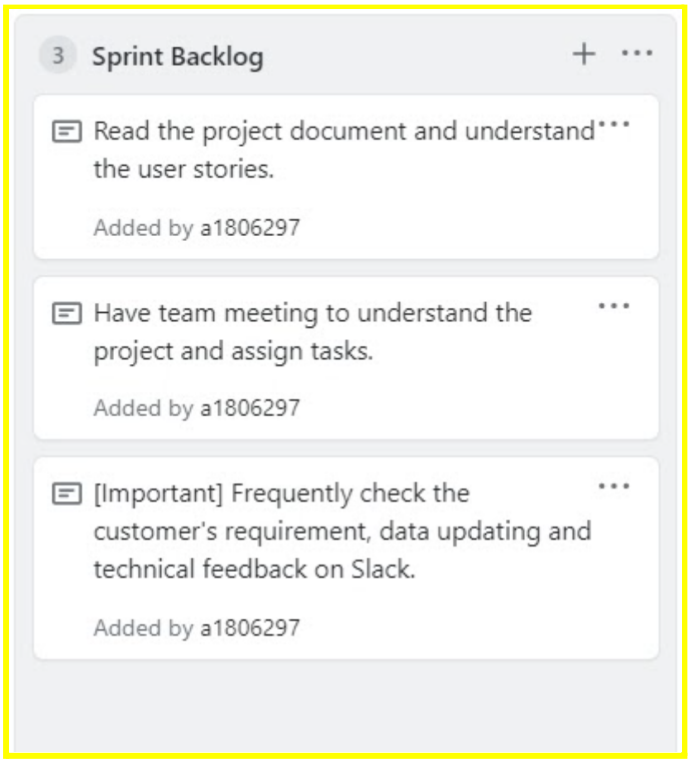
Product Backlog:



Task Board:


8 To do
Research on the format and meaning of both input data and output data.
Added by a1806297
Determine and set up communication channels
Added by a1162576
Determine the frequency and time of meetings for member each week
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Research papers/techniques on block model compression
Added by a1784375
0 In progress
0 Done
8 To do
meetings for member each week
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Research papers/techniques on block model compression
Added by a1784375
Develop Set of Coding Standards
Added by a1162576
Set up Static Code Analysis
Added by a1162576
Sketch out initial software architecture
Added by a1162576
0 In progress
0 Done

Sprint Backlog and User Stories:



In the kickoff meeting, the customer shared a project document that described their requirements, including finding the best algorithms for the block model compression and competing with the other teams. To achieve these requirements, we needed to carefully **read the input and output data requirements, the type of uploaded files, the constraints of the project, and so on**.

Before the first sprint, we plan to **have our first meeting to let our team understand the details of the project, the Scrum process and the User Stories**. We also plan to assign tasks to group members to implement the project more efficiently and smoothly.

Moreover, the Scrum Master asks the team to **frequently check the updated information on Slack**, where the customer will update the user stories and any technical feedback. It can help the team design and implement the programs.

Definition of Done:

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our Github repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards and other issues such as unused variables and cyclomatic complexity.

Summary of Changes:

After receiving the project's requirements and user stories from the customer, we listed the tasks in the "to do" part on Github Projects. The tasks include setting up the communication tools, scheduling our regular weekly meetings, collecting the team member's ideas, deciding the Scrum Master for every sprint, assigning the tasks, etc.

We will develop our team’s Definition of Done (DOD) to set up the standard of code we write, ordering and emerging a list of user stories on the Product Backlog to complete the product vision. In the Sprint Backlog, we will set up the goal including reading the project documentation and writing the code from scratch according to the customer’s requirement.

Snapshot Week 3 of Group BLOCKS7PG

Product Backlog and Task Board:

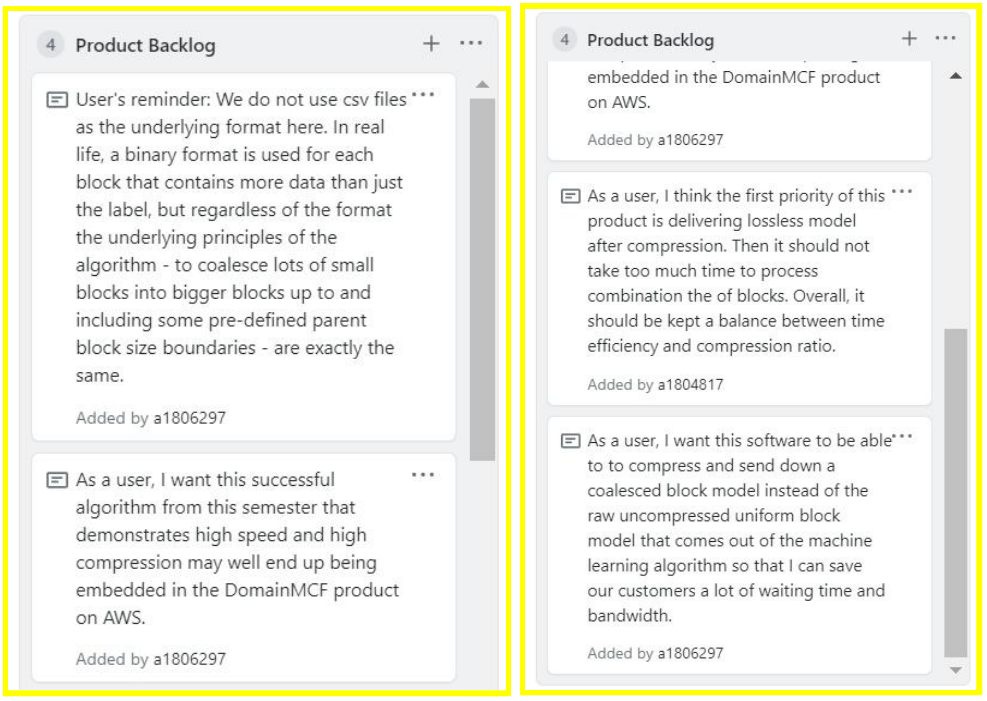


Figure1: Product Backlog1.1

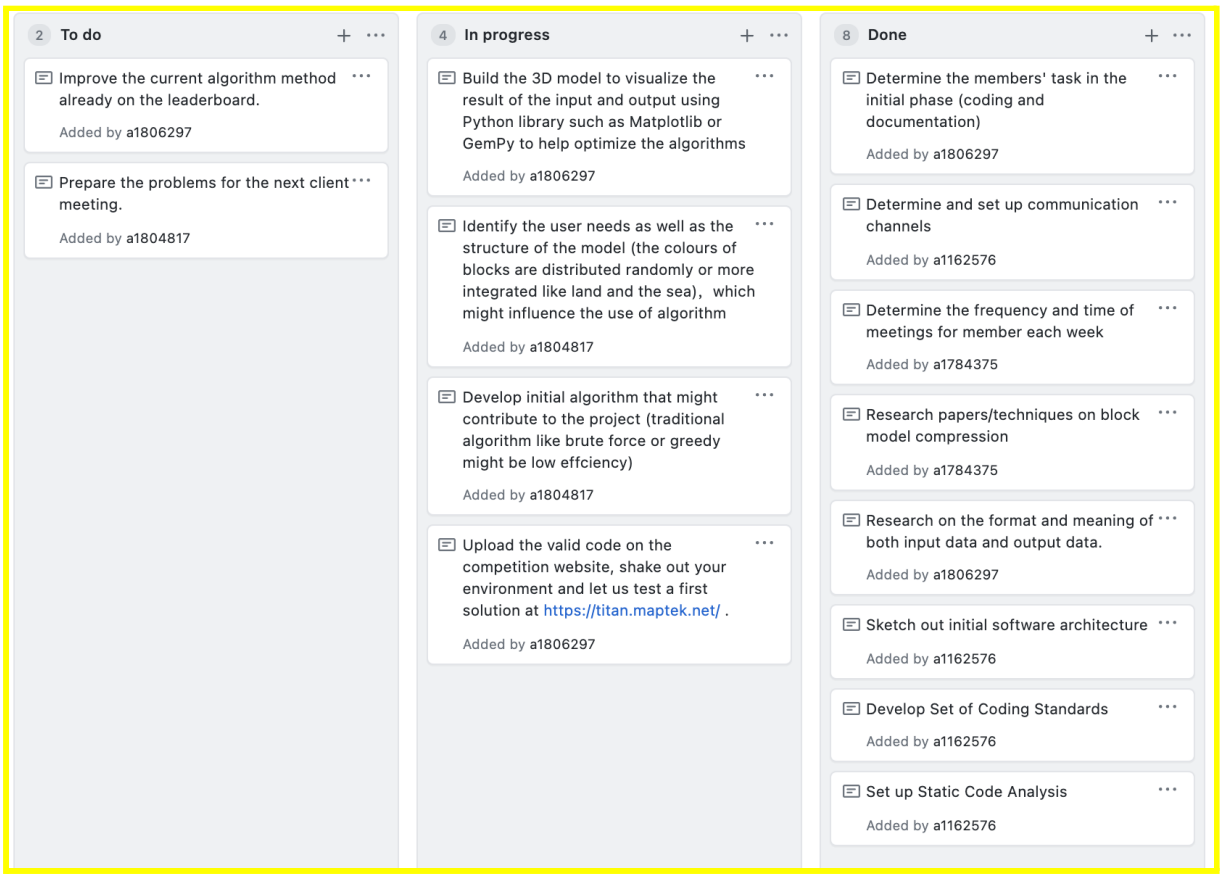


Figure2: Task Board1.1

Sprint Backlog and User Stories

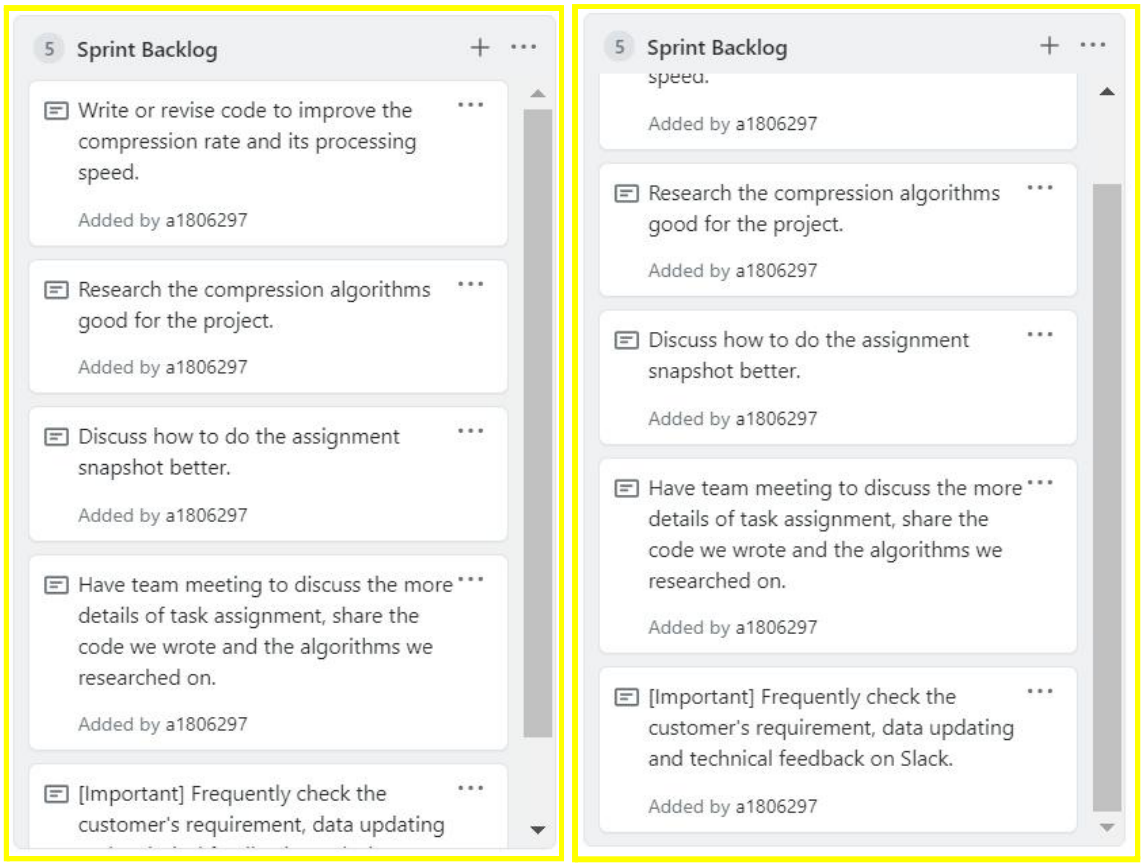


Figure3: Sprint Backlog and User Stories 1.1

In the middle of the week, Yuanpeng has written and submitted the first version of code to the competition system and have a compression rate of 89.23% and processing speed of 37.32% for the input data "the_intro_one_32768_4x4x4" and compression rate 74.59 % and processing speed 36.18% for the input data "the_fast_one_376000_2x2x2 ". In the sprint, we would revise the current code to **improve its compression rate and processing speed**. In addition to the code written from scratch, we also **research the algorithm methods** such as Run Length Encoding (RLE) to achieve better compression and speed.

In the team meeting, we **discuss how to assign tasks and assignments better** to implement the project more efficiently. The Scrum Master asks the team to frequently check the updated information on Slack, where the customer will update the user stories and any technical feedback. It can assist the team in designing and implementing the programs.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our Github repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.

Summary of Changes:

In the first week of the first sprint, we set up the communication tools, scheduled regular weekly meetings, collected the team member's ideas, assigned the tasks, etc. We also added user stories in the Product Backlog on our Github Projects and revised the Definition of Done (DOD).

The sprint goals have been set in the Sprint Backlog, including figuring out the project and user stories. We executed the tasks such as designing the algorithms and figuring out how to use the competition system and successfully wrote the first version of the code. We managed to revise and develop better algorithms.

Snapshot Week 4 of Group BLOCKS7PG

Product Backlog and Task Board:

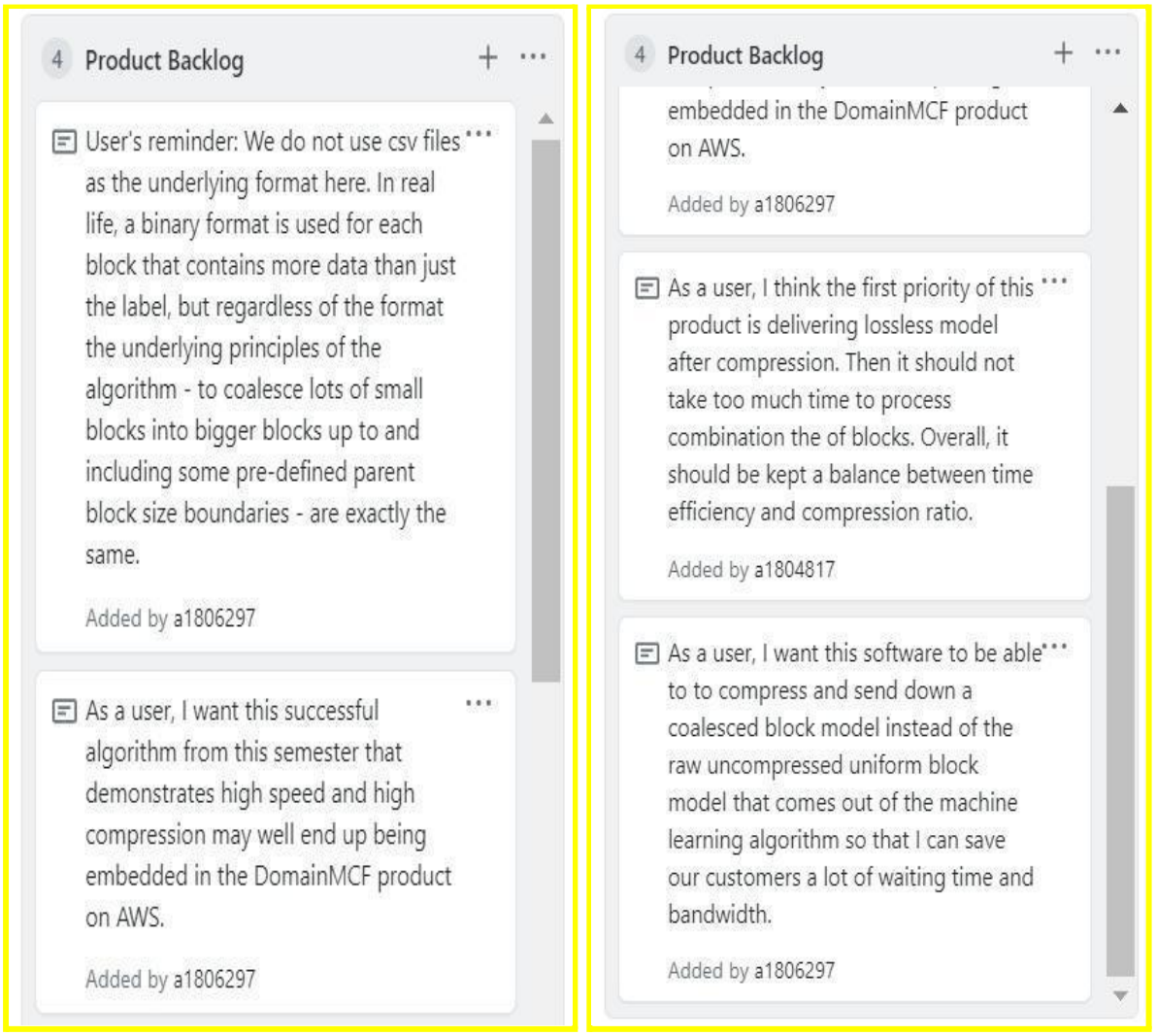
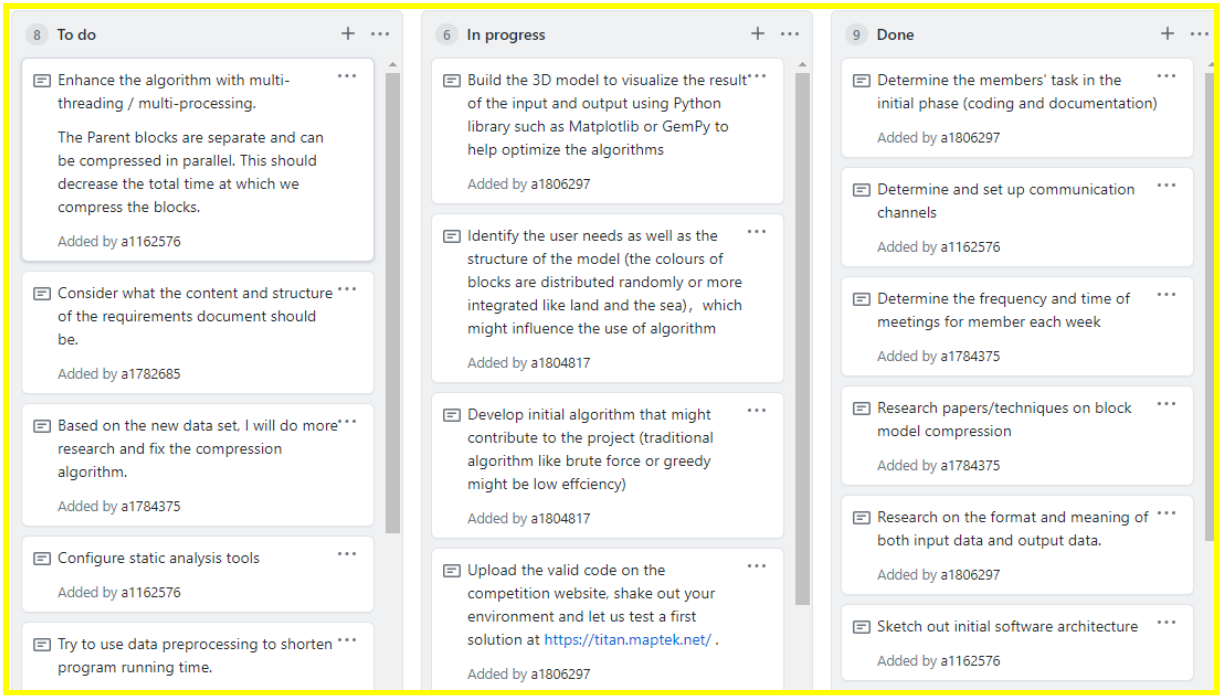


Figure 4: Product Backlog 1.2



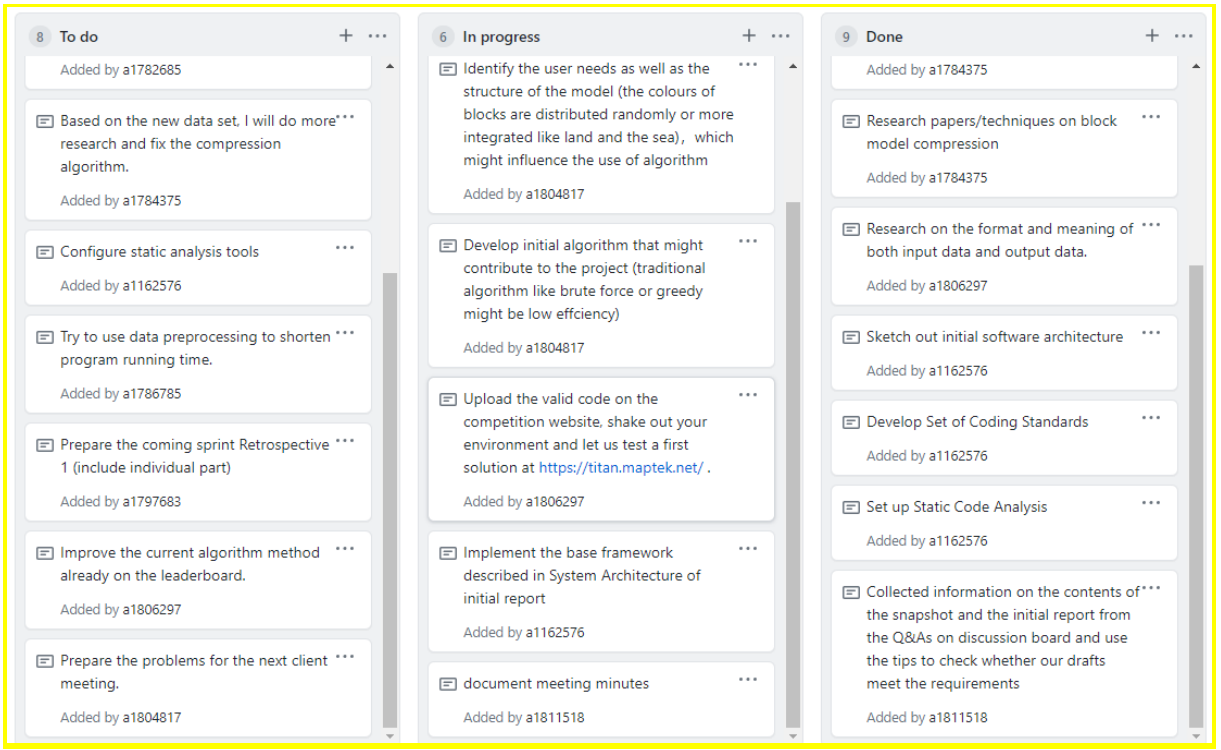


Figure 5: Task Board 1.2

Sprint Backlog and User Stories

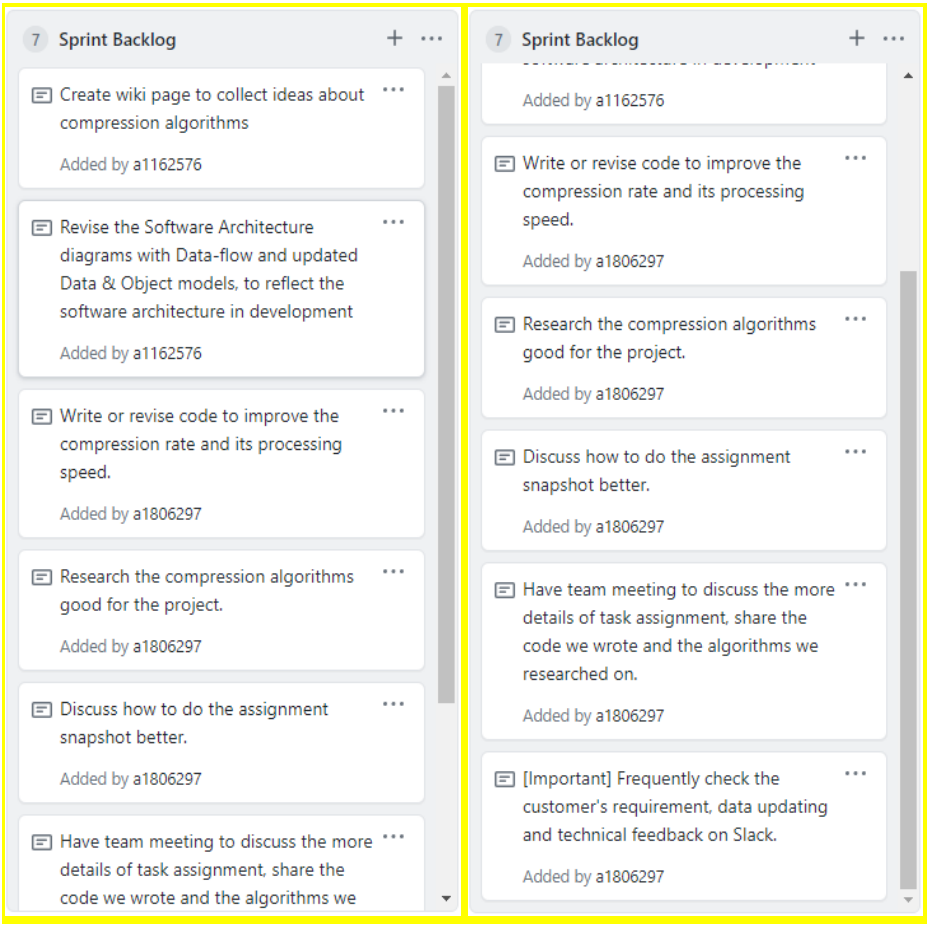


Figure 6: Sprint Backlog and User Stories 1.2

Since we have completed the first stage of the client's requirements ahead of schedule, and the new test file has not been uploaded by the client, the main work this week is to **optimize the coding algorithm and data structure.** We try to **create a wiki page to collect ideas** about compression algorithms so that we have more options to try to compress images. In addition, we **revised our software architecture** and tried to **use Object models to reflect the software architecture** in development.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.

Summary of Changes:

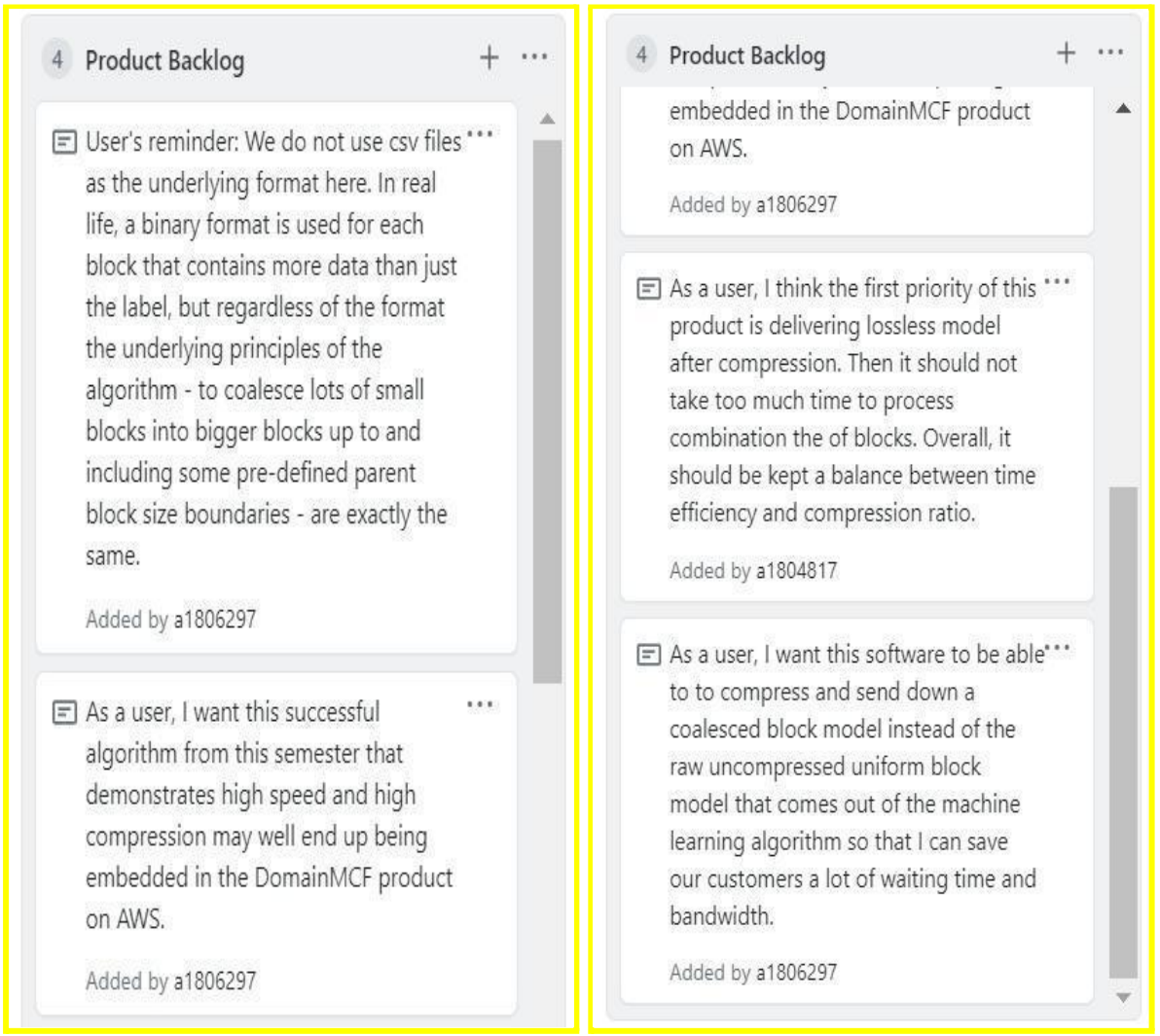
The main work of this week is improving the existing code. After the sprint review, we focused on the following three areas: algorithm improvement, data structure improvement, and multithreading/multiprocessing in Python. In addition, our software architecture has also been further improved, and we have tried to use 3D object models to facilitate our model analysis.

In terms of client communication and project progress, we plan to use the wiki page to collect ideas about the compression algorithm, summarize and analyze problems in the current project, and prepare questions for the client meeting next week. In addition, we selected suitable candidates to be Scrum masters according to the current situation of team members, so as to ensure that these scrum masters can effectively organize all team members, assign tasks, and complete the goals of each sprint.

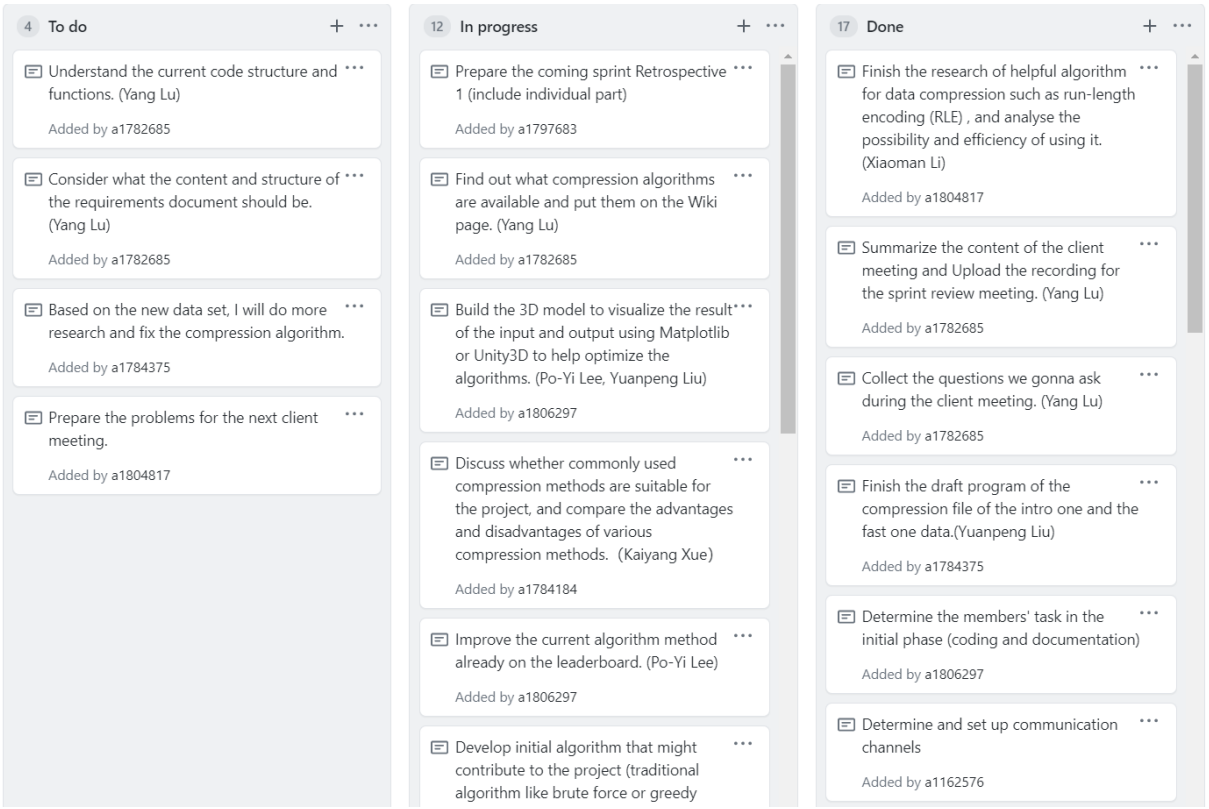
Snapshot Week 5 of Group BLOCKS7PG

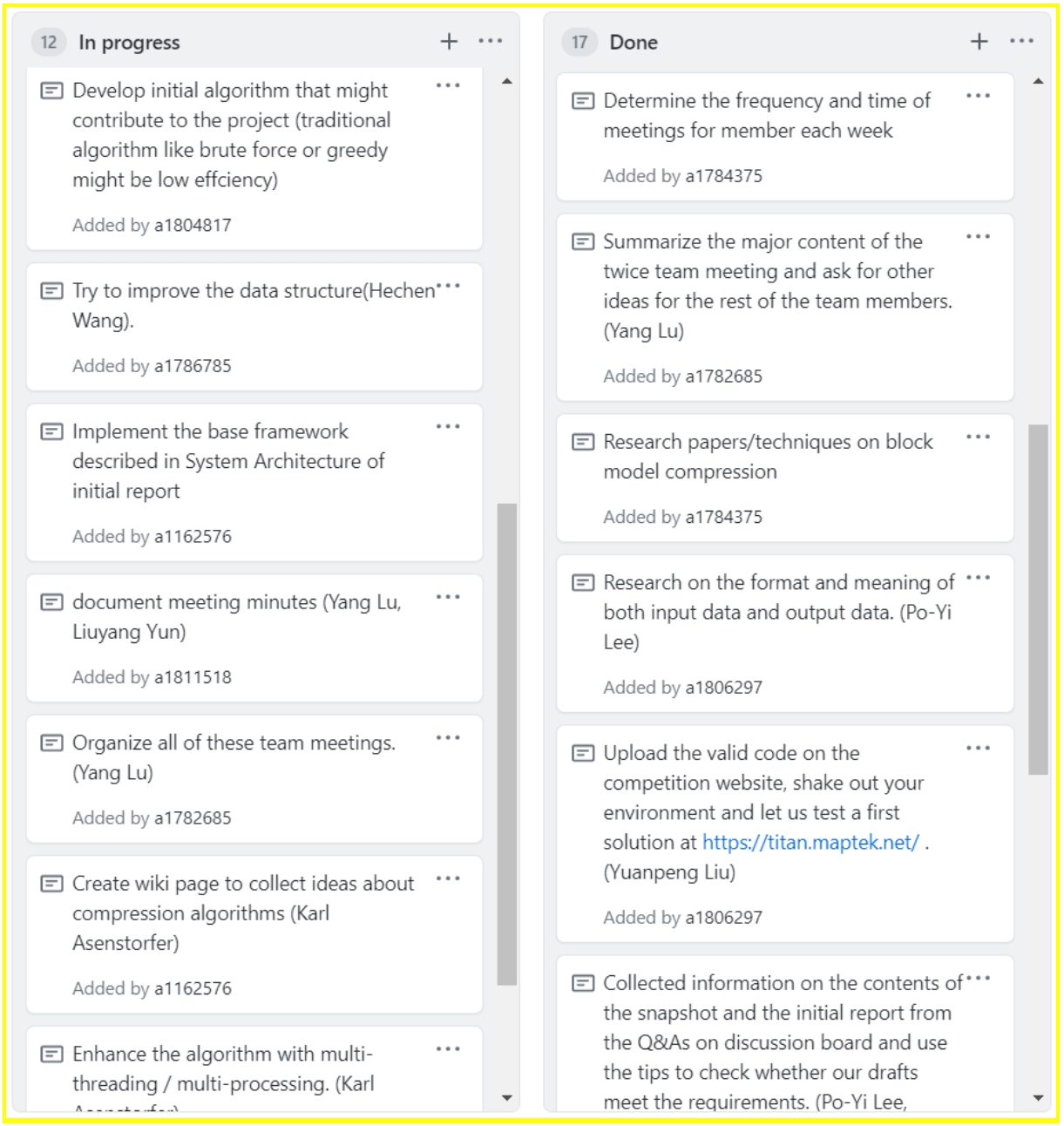
Product Backlog and Task Board

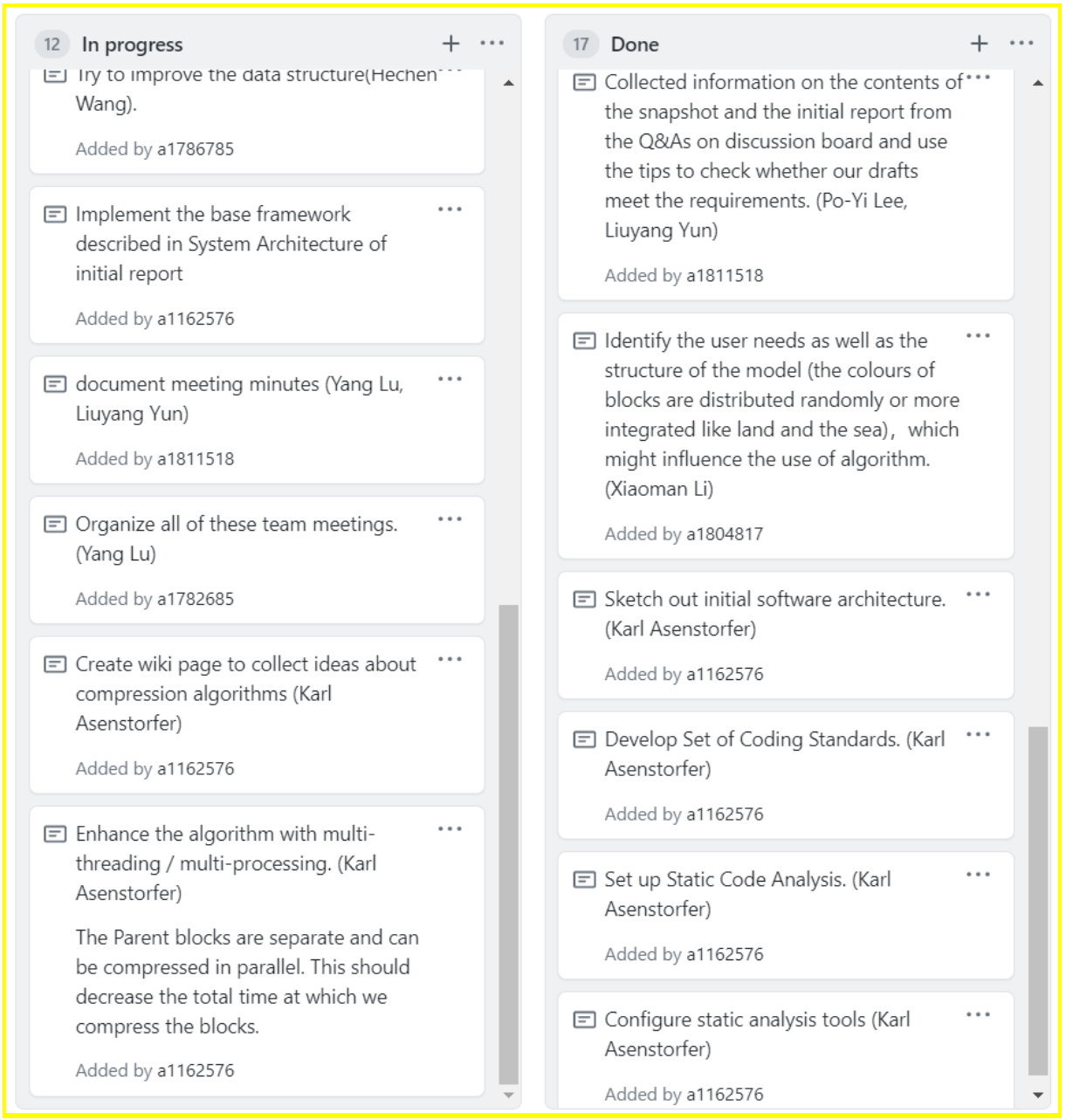
Product Backlog:



Task Board:


4 To do
Understand the current code structure and functions. (Yang Lu)
Added by a1782685
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
12 In progress
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Find out what compression algorithms are available and put them on the Wiki page. (Yang Lu)
Added by a1782685
Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
Added by a1806297
Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy
17 Done
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817
Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
Added by a1782685
Collect the questions we gonna ask during the client meeting. (Yang Lu)
Added by a1782685
Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Determine and set up communication channels
Added by a1162576


12 In progress
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Try to improve the data structure(Hechen Wang).
Added by a1786785
Implement the base framework described in System Architecture of initial report
Added by a1162576
document meeting minutes (Yang Lu, Liuyang Yun)
Added by a1811518
Organize all of these team meetings. (Yang Lu)
Added by a1782685
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576
Enhance the algorithm with multi-threading / multi-processing. (Karl
17 Done
Determine the frequency and time of meetings for member each week
Added by a1784375
Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
Added by a1782685
Research papers/techniques on block model compression
Added by a1784375
Research on the format and meaning of both input data and output data. (Po-Yi Lee)
Added by a1806297
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
Added by a1806297
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee,


12 In progress
Try to improve the data structure(Hechen Wang).
Added by a1786785
Implement the base framework described in System Architecture of initial report
Added by a1162576
document meeting minutes (Yang Lu, Liuyang Yun)
Added by a1811518
Organize all of these team meetings. (Yang Lu)
Added by a1782685
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576
Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
17 Done
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun)
Added by a1811518
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576

Sprint Backlog and User Stories

There are no major modifications to the Sprint Backlog and User Stories, and we are mainly trying to implement the software architecture mentioned in the initial report. The abstraction of the model was to be initialized implementation.

Also, after the client meeting, we realized that we still needed to collect information on various compression algorithms and put them into a wiki page. And to do that, we need a better division of labor.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned

relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.

- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a classmethod to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compressionEngine is responsible for delegating the compression to the various algorithms. The parentBlocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.

Summary of Changes:

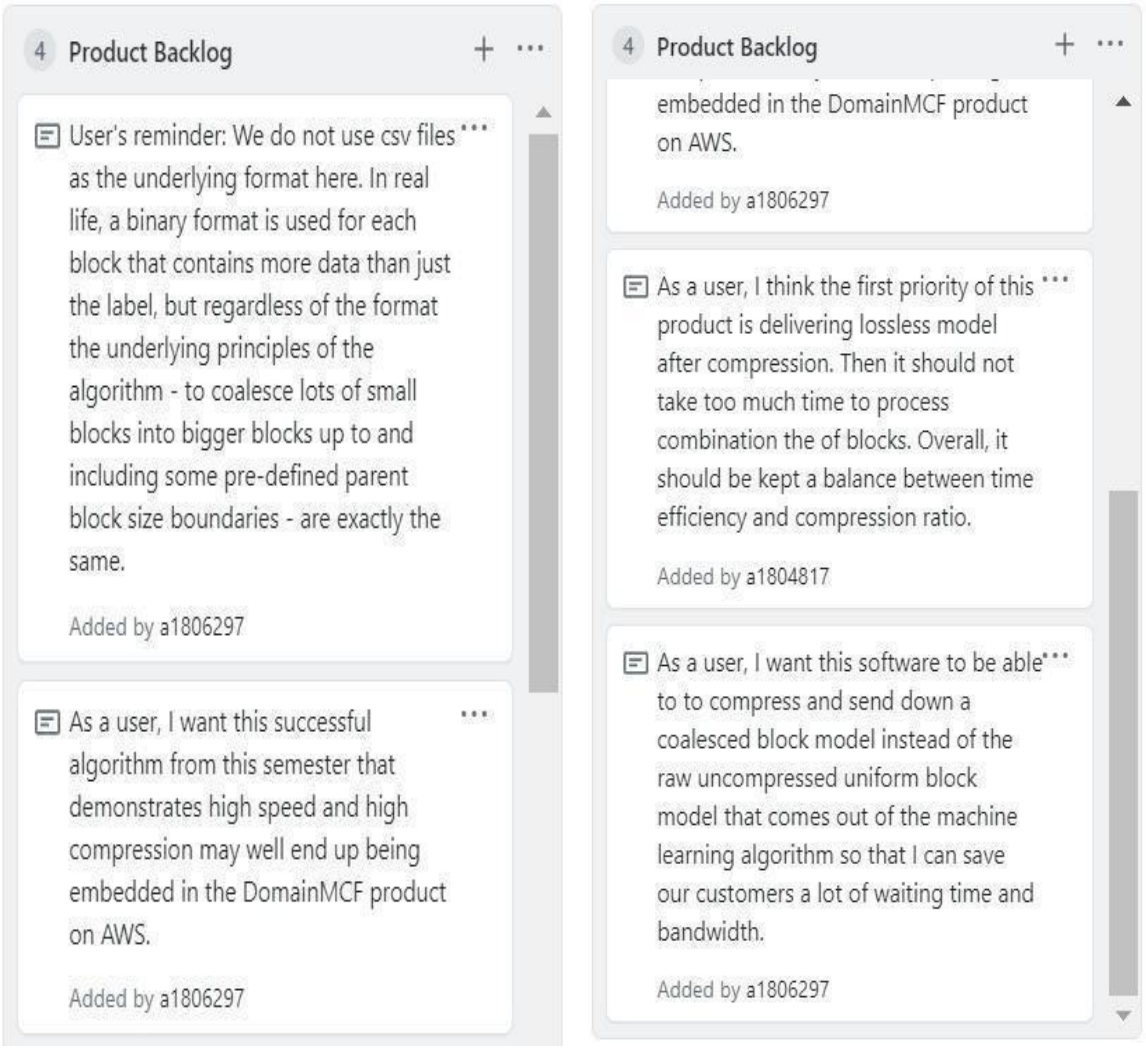
This week we had the client meeting and the sprint review meeting. During the formal sprint review meeting, we discussed what we had already done: test our algorithm for the dataset one and two. In addition, we talked about what we plan to do in the next step: implementing the algorithm which can accept some larger datasets, and trying to visualize our output more clearly. We collected both the technical problems and some of the non-technical problems, which we will probably face when implementing the new algorithm in the next week. At the end of the review meeting, we decided what technical and non-technical questions we planned to ask during the client meeting.

During the client meeting, most of the questions were answered smoothly. For issues that need to be addressed further in the implementation process, we'll ask questions on the Slack channel. At the same time, we realized that we needed a more rational division of labor. The task is now divided into several modules: compressing the dataset, visualizing the output, searching for suitable compression algorithms, and more. So we're trying to reclassify these tasks in a much more acceptable way to make sure that everyone is contributing.

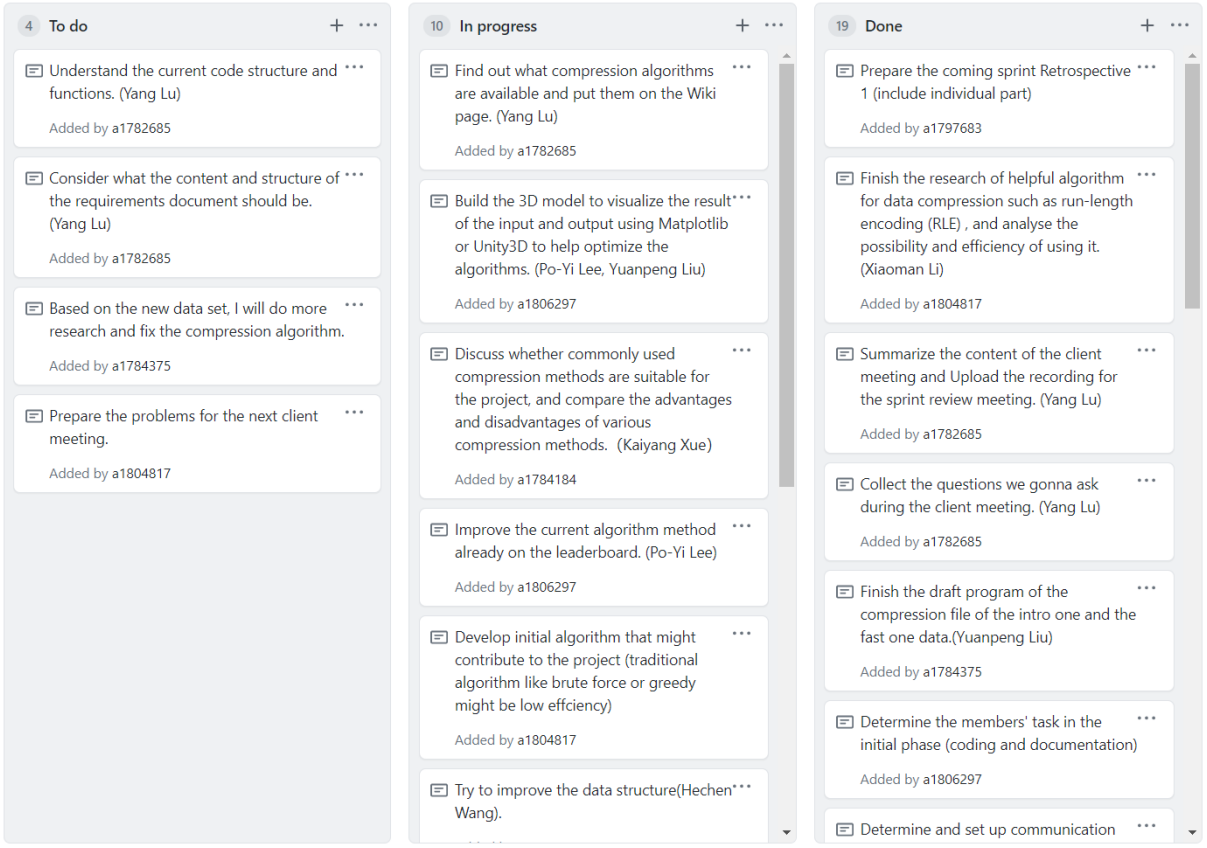
Snapshot Week 6 of Group BLOCKS7PG

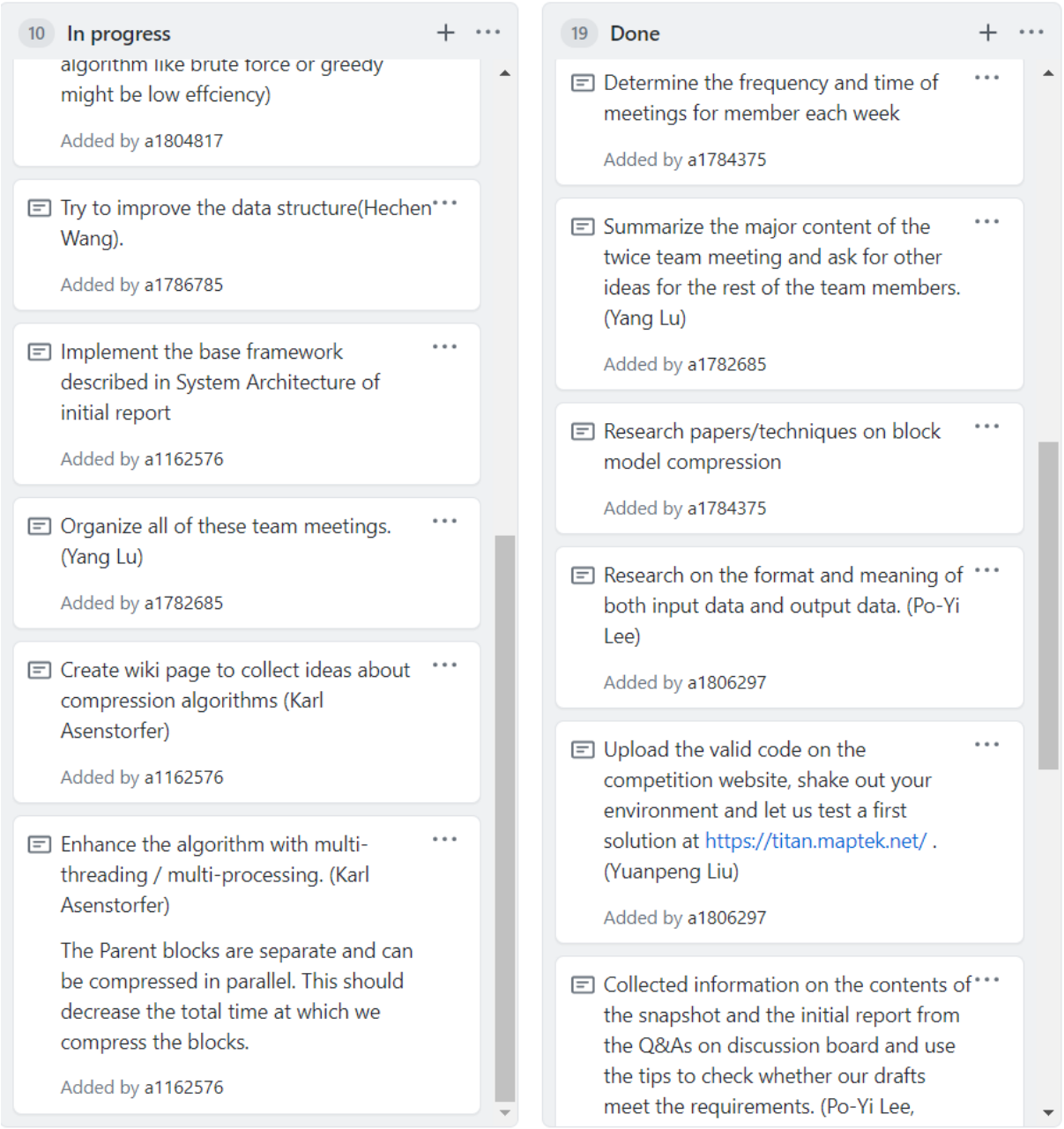
Product Backlog and Task Board

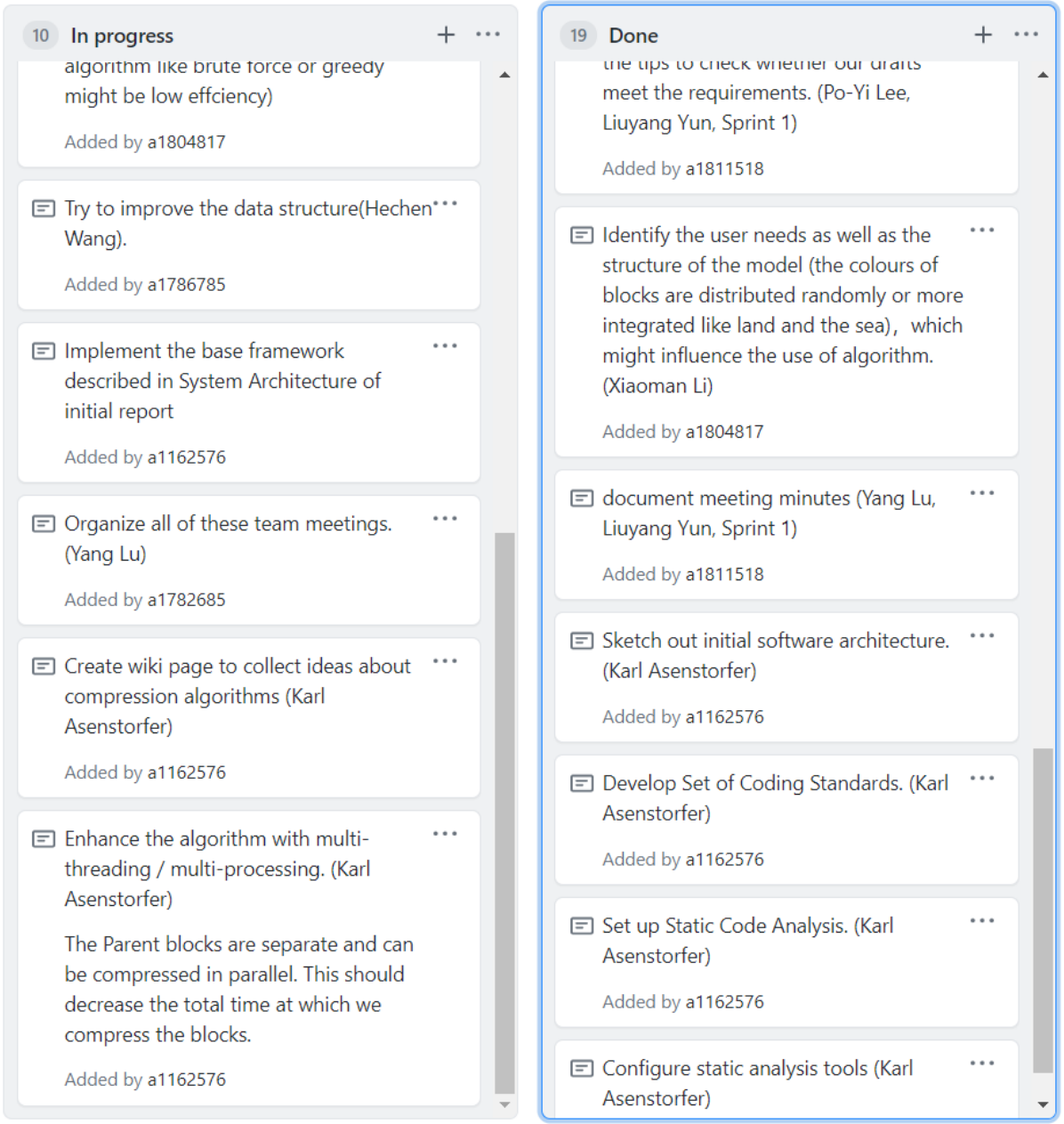
Product Backlog:



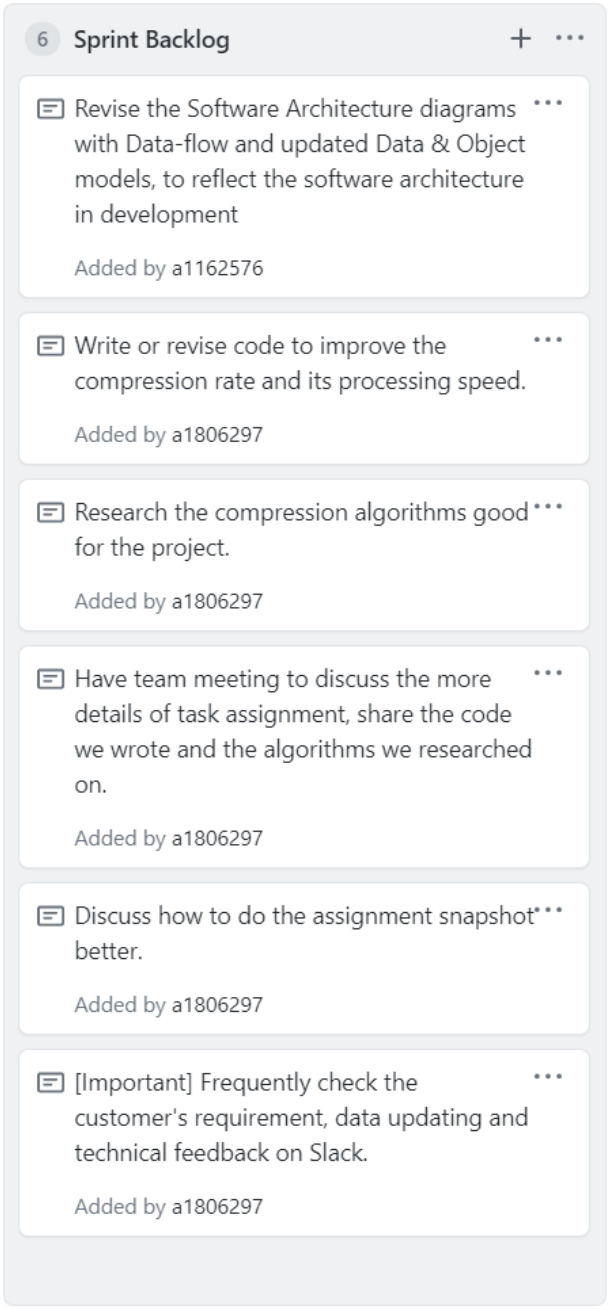
Task Board:


4 To do
Understand the current code structure and functions. (Yang Lu)
Added by a1782685
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
10 In progress
Find out what compression algorithms are available and put them on the Wiki page. (Yang Lu)
Added by a1782685
Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
Added by a1806297
Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Try to improve the data structure(Hechen Wang).
19 Done
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817
Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
Added by a1782685
Collect the questions we gonna ask during the client meeting. (Yang Lu)
Added by a1782685
Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Determine and set up communication


10 In progress
algorithm like brute force or greedy might be low effciency)
Added by a1804817
Try to improve the data structure(Hechen Wang).
Added by a1786785
Implement the base framework described in System Architecture of initial report
Added by a1162576
Organize all of these team meetings. (Yang Lu)
Added by a1782685
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576
Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
19 Done
Determine the frequency and time of meetings for member each week
Added by a1784375
Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
Added by a1782685
Research papers/techniques on block model compression
Added by a1784375
Research on the format and meaning of both input data and output data. (Po-Yi Lee)
Added by a1806297
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
Added by a1806297
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee,


10 In progress
algorithm like brute force or greedy might be low effciency)
Added by a1804817
Try to improve the data structure(Hechen Wang).
Added by a1786785
Implement the base framework described in System Architecture of initial report
Added by a1162576
Organize all of these team meetings. (Yang Lu)
Added by a1782685
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576
Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
19 Done
the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)
Added by a1811518
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)

Sprint Backlog and User Stories



There are no major modifications to the Sprint Backlog and User Stories, and we are mainly trying to implement the software architecture mentioned in the initial report. The abstraction of the model was to be initialized implementation.

Also, after the weekly meeting, we clarify some confusion about the existing framework. Besides that, we decided what we were going to do in the next phase: Test after we got a bigger data set.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a classmethod to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compressionEngine is responsible for delegating the compression to the various algorithms. The parentBlocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.

Summary of Changes:

This week, we mainly completed the familiarization with the software framework written by Karl and laid a good foundation for the subsequent development. In addition, the

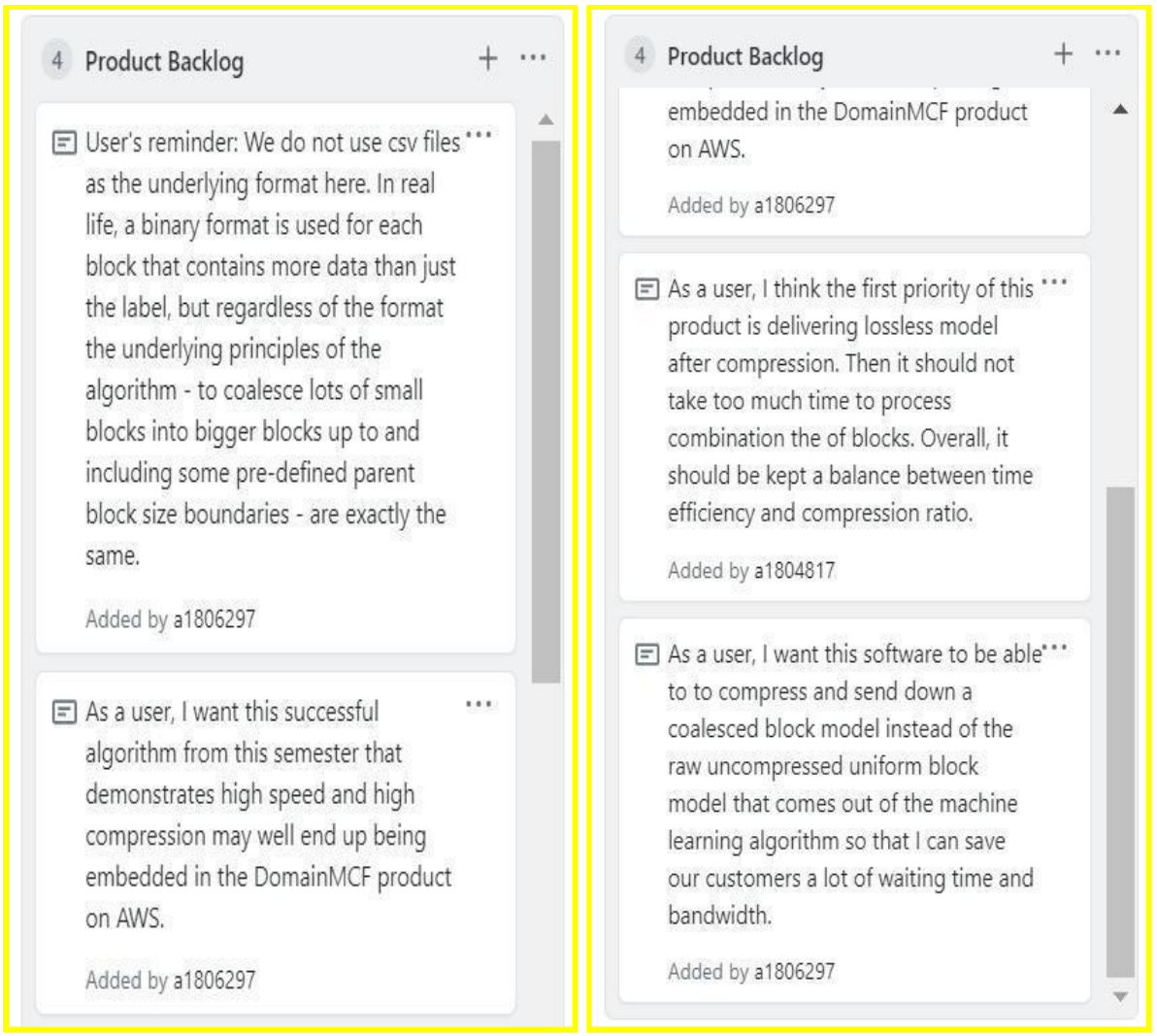
visualization of the compression results is also underway, and it has been decided to implement it by unity3D. In addition, in the weekly group meeting, we discussed the following detailed division of labour.

We collected some compression algorithms during the implementation process and evaluated their operating efficiency, compression ratio, and compression time. In this way, relatively excellent compression algorithms are selected as the compression algorithms for subsequent large data sets.

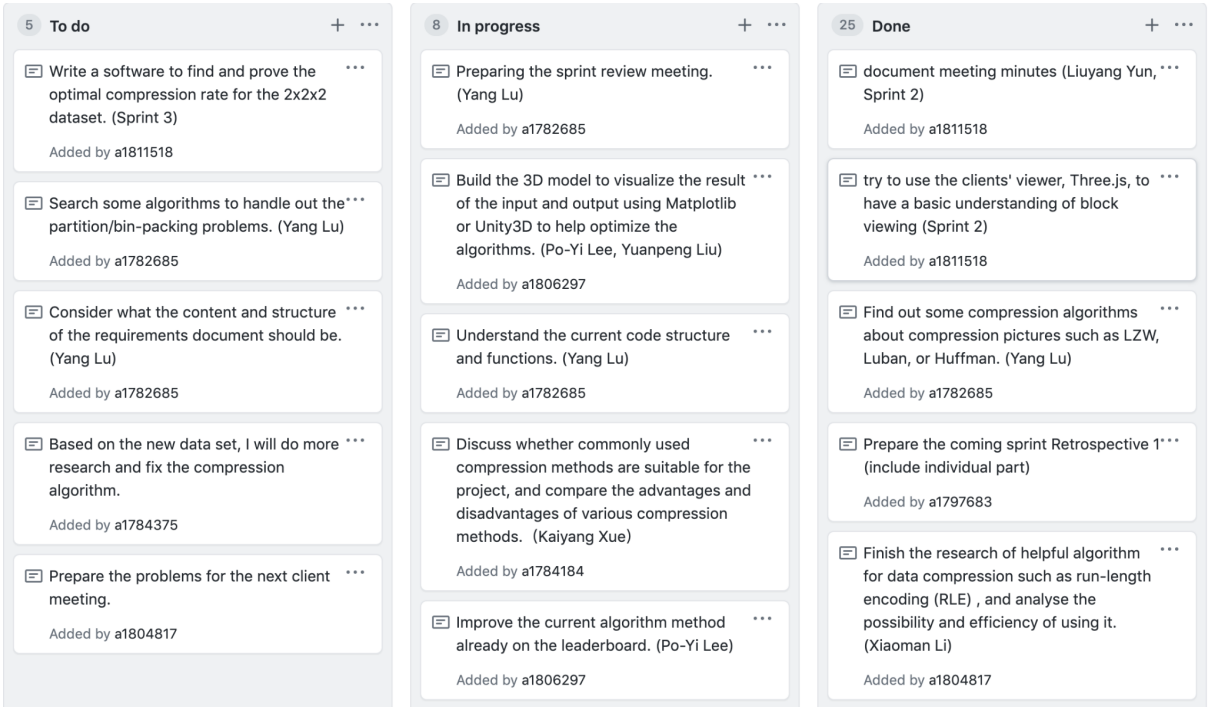
Snapshot Week 7 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:



Task Board:


5 To do
Write a software to find and prove the optimal compression rate for the 2x2x2 dataset. (Sprint 3)
Added by a1811518
Search some algorithms to handle out the partition/bin-packing problems. (Yang Lu)
Added by a1782685
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
8 In progress
Preparing the sprint review meeting. (Yang Lu)
Added by a1782685
Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
Added by a1806297
Understand the current code structure and functions. (Yang Lu)
Added by a1782685
Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
25 Done
document meeting minutes (Liuyang Yun, Sprint 2)
Added by a1811518
try to use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Yang Lu)
Added by a1782685
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817

8 In progress

- Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)

  Added by a1784184

- Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

  Added by a1806297

- Try to improve the data structure(Hechen Wang).

  Added by a1786785

- Organize all of these team meetings. (Yang Lu)

  Added by a1782685

- Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)

  The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.

  Added by a1162576

25 Done

- Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)

  Added by a1782685

- Collect the questions we gonna ask during the client meeting. (Yang Lu)

  Added by a1782685

- Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)

  Added by a1784375

- Determine the members' task in the initial phase (coding and documentation)

  Added by a1806297

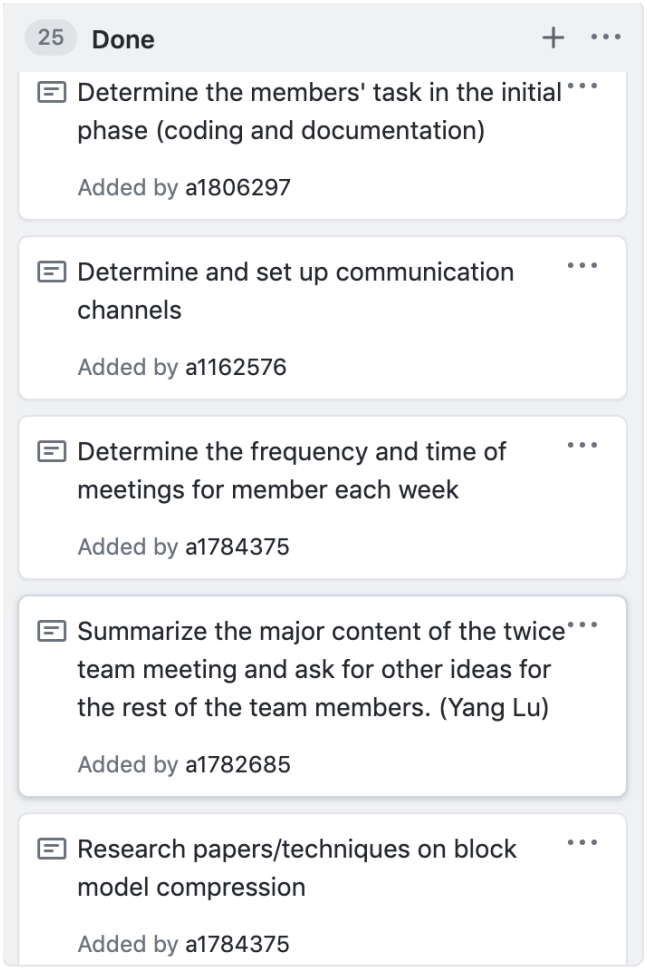
- Determine and set up communication channels

  Added by a1162576

- Determine the frequency and time of meetings for member each week

  Added by a1784375

- Summarize the major content of the twice


25 Done
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Determine and set up communication channels
Added by a1162576
Determine the frequency and time of meetings for member each week
Added by a1784375
Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
Added by a1782685
Research papers/techniques on block model compression
Added by a1784375

25 **Done**

Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)

Added by a1782685

Research papers/techniques on block model compression

Added by a1784375

Research on the format and meaning of both input data and output data. (Po-Yi Lee)

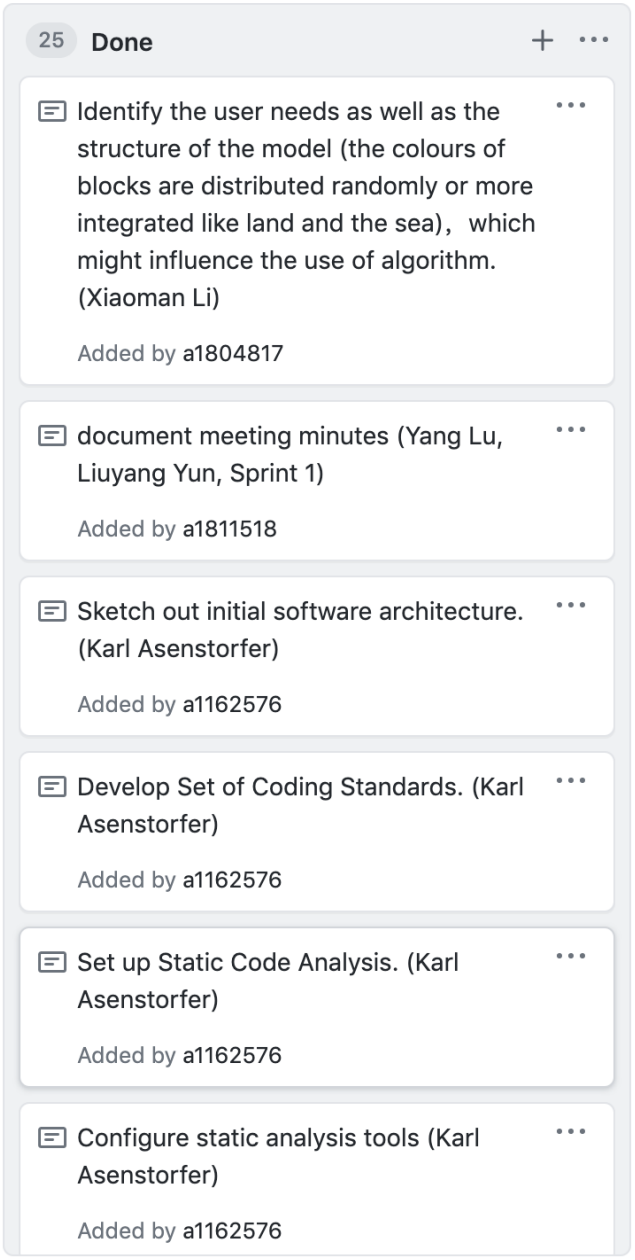
Added by a1806297

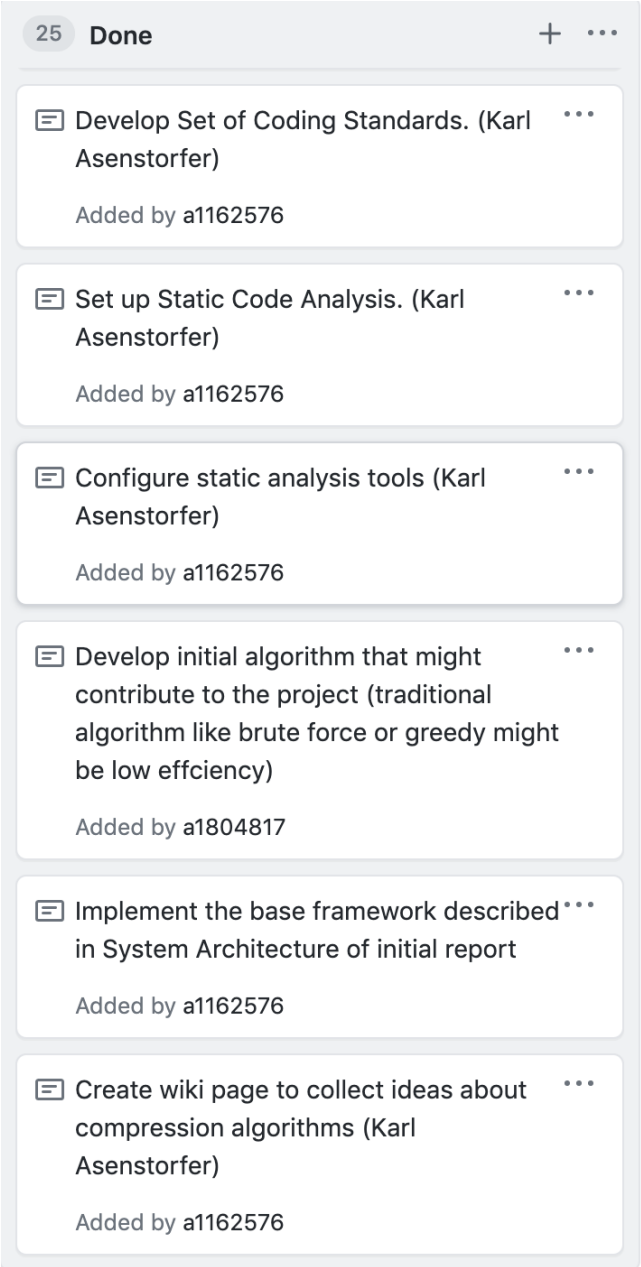
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ .(Yuanpeng Liu)

Added by a1806297

Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)

Added by a1811518


25 Done
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea)，which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576


25 Done
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Implement the base framework described in System Architecture of initial report
Added by a1162576
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576

Sprint Backlog and User Stories

There are no major modifications to the Sprint Backlog and User Stories, and we are mainly trying to implement the software architecture mentioned in the initial report. The abstraction of the model was to be initialized implementation.

Also, after the client meeting, we realized that we still needed to collect information on various compression algorithms and put them into a wiki page. And to do that, we need a better division of labour

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We try to use the clients' viewer, Three.js, to have a basic understanding of block viewing.
- We create a wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.

Summary of Changes:

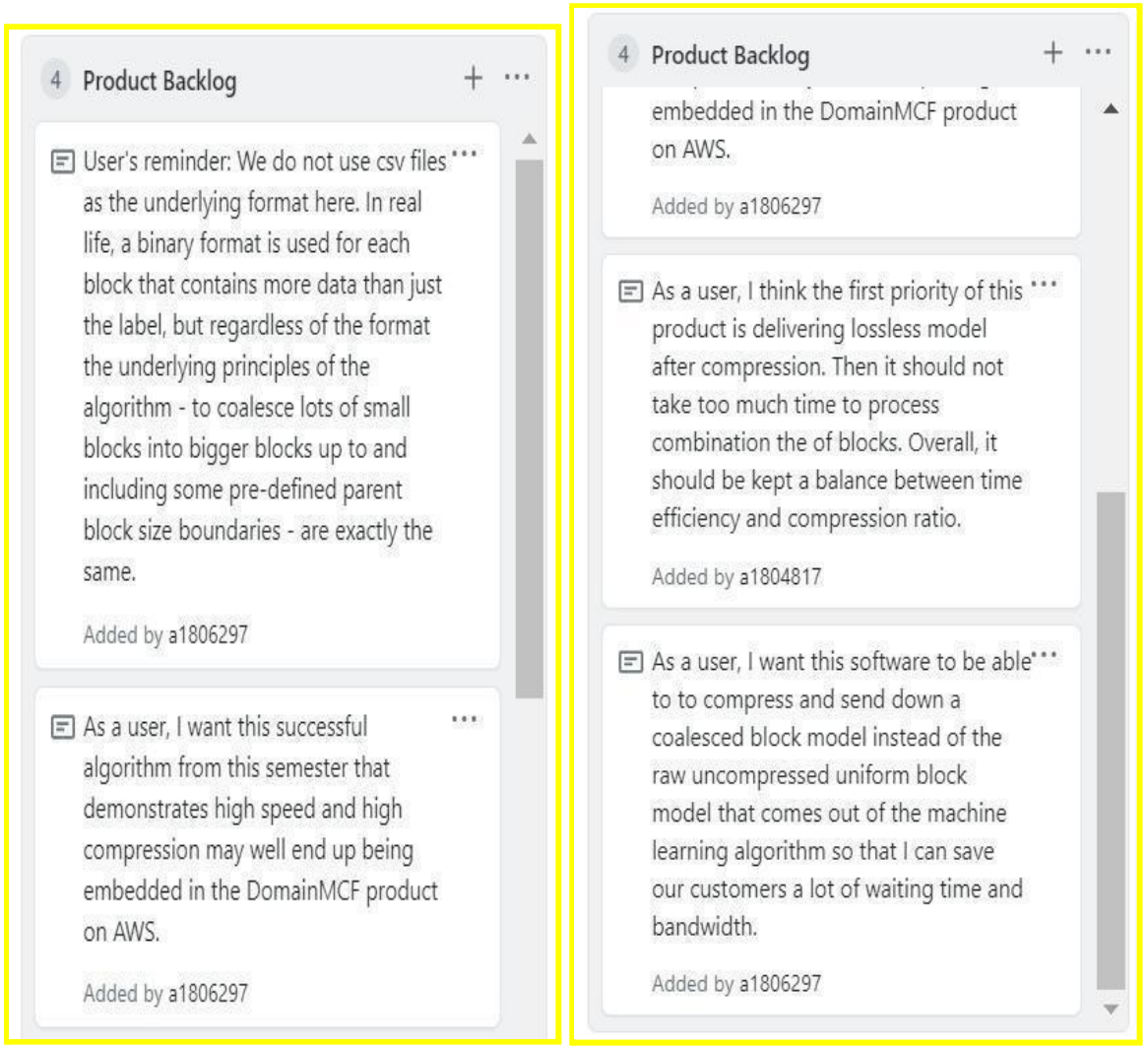
In this week, we extended the modular architecture written by Karl last week, using multiprocessing, and Karl also added some better methods to better access the blocks in the parent block. And we add the information we collected to the wiki page to improve the algorithm, including Recursive Cubes which merges 2x2x2 blocks in the same domain until no cube can be generated; Combine Along 1 Dimension, which starting at the (0,0,0) block and continuing along with one of the dimensions x, y, or z and combining all blocks in the same domain into 1 block; Expanding Blocks, which Starting from a block at (x, y, z), if this block can be merged, it will continue to judge from its adjacent position such as (x+1, y, z) until no more blocks can be added.

In this client meeting, we reported our project progress to the client and clarified the goals for the next sprint, including proving what the optimal compression is for the 2*2*2 data set and submitting an algorithm to compress the new datasets by the end of the next week (submitting and iterating to get improvements to the system).

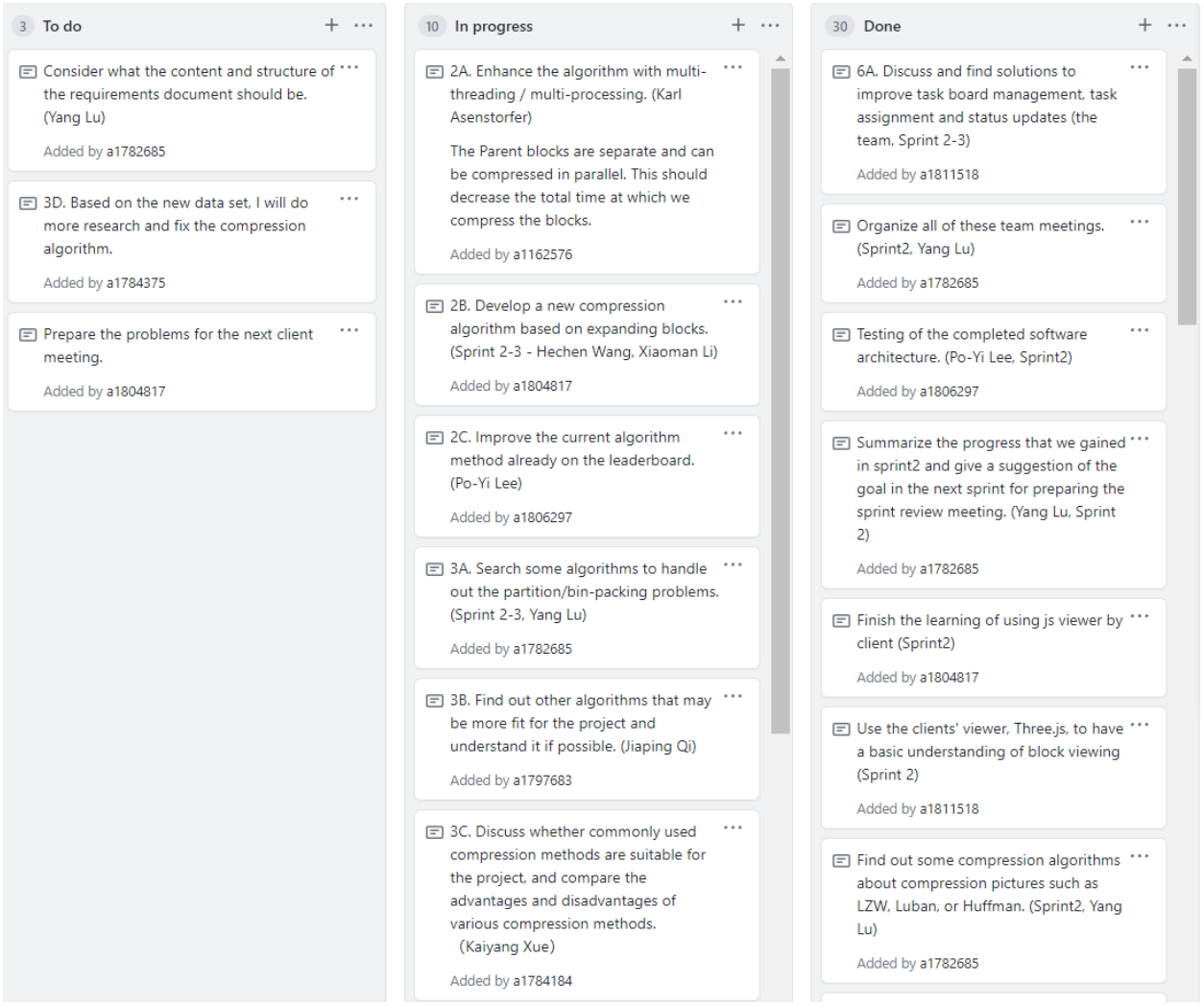
Snapshot Week 8 of Group BLOCKS7PG

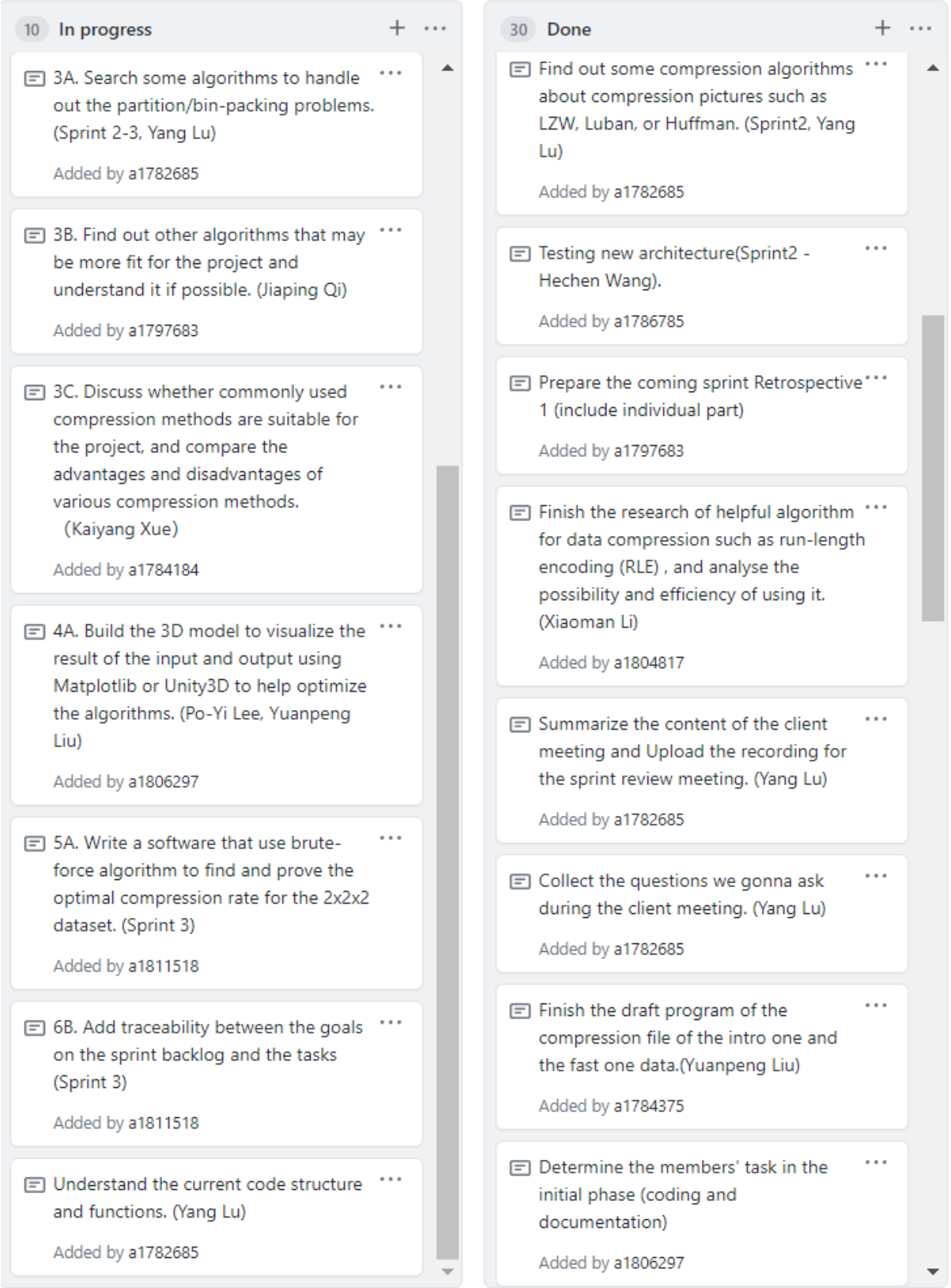
Product Backlog and Task Board

Product Backlog:



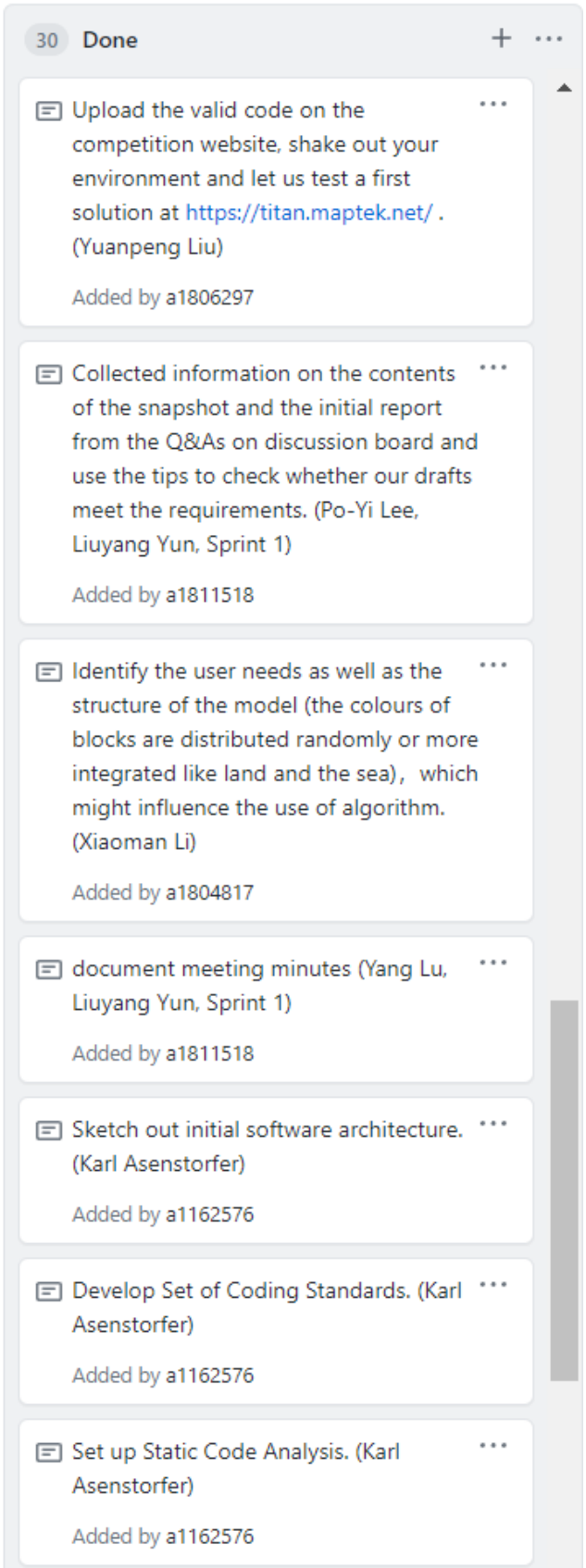
Task Board:

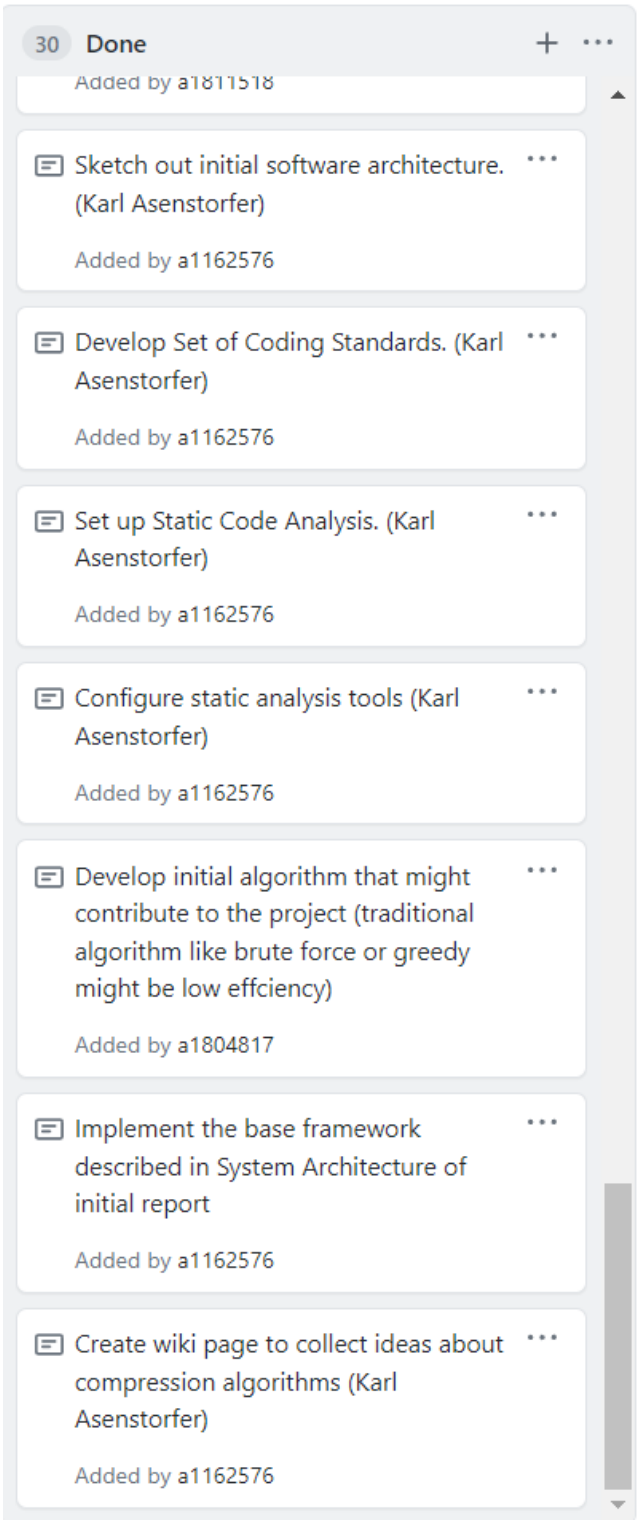

3 To do
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
3D. Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
10 In progress
2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
2B. Develop a new compression algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)
Added by a1804817
2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
3A. Search some algorithms to handle out the partition/bin-packing problems. (Sprint 2-3, Yang Lu)
Added by a1782685
3B. Find out other algorithms that may be more fit for the project and understand it if possible. (Jiaping Qi)
Added by a1797683
3C. Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
30 Done
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)
Added by a1811518
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
Added by a1782685


10 In progress
3A. Search some algorithms to handle out the partition/bin-packing problems. (Sprint 2-3, Yang Lu)
Added by a1782685
3B. Find out other algorithms that may be more fit for the project and understand it if possible. (Jiaping Qi)
Added by a1797683
3C. Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
Added by a1806297
5A. Write a software that use brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset. (Sprint 3)
Added by a1811518
6B. Add traceability between the goals on the sprint backlog and the tasks (Sprint 3)
Added by a1811518
Understand the current code structure and functions. (Yang Lu)
Added by a1782685
30 Done
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
Added by a1782685
Testing new architecture(Sprint2 - Hechen Wang).
Added by a1786785
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817
Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
Added by a1782685
Collect the questions we gonna ask during the client meeting. (Yang Lu)
Added by a1782685
Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297

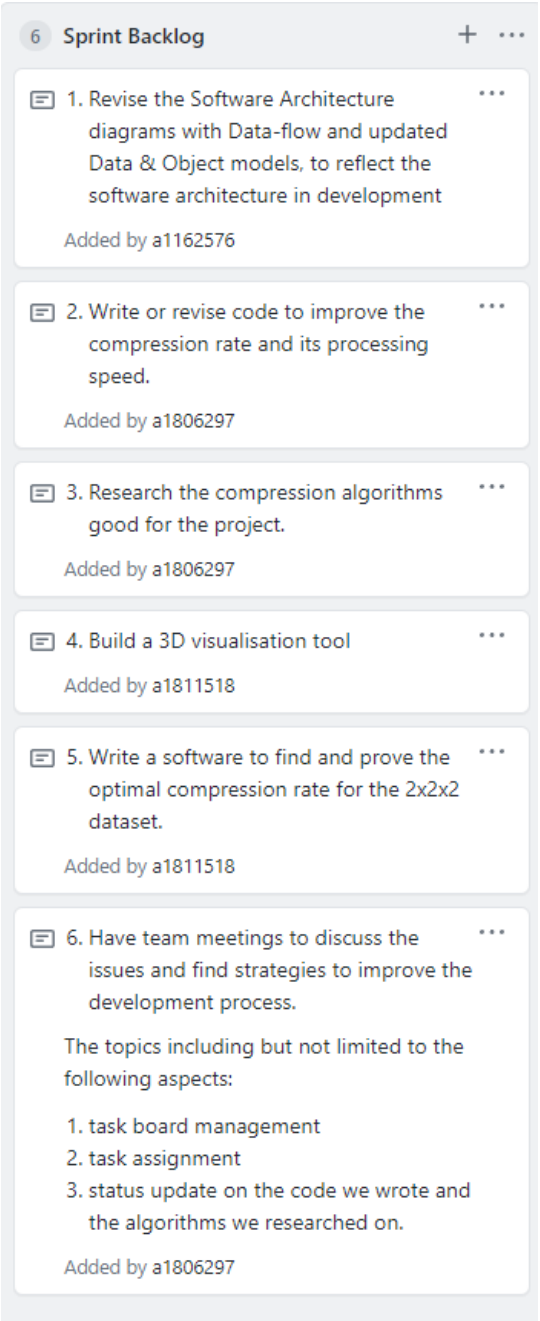
30 Done

- Determine the members' task in the initial phase (coding and documentation)
  Added by a1806297
- Determine and set up communication channels
  Added by a1162576
- Determine the frequency and time of meetings for member each week
  Added by a1784375
- Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
  Added by a1782685
- Research papers/techniques on block model compression
  Added by a1784375
- Research on the format and meaning of both input data and output data. (Po-Yi Lee)
  Added by a1806297
- Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
  Added by a1806297


30 Done
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
Added by a1806297
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)
Added by a1811518
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576


30 Done
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Implement the base framework described in System Architecture of initial report
Added by a1162576
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576

Sprint Backlog and User Stories



For this week, there are no major changes to the sprint backlog. The main task currently is to optimise the current algorithm based on the existing software architecture. According to the discussion in this week's sprint review meeting, we came up with several new points for the standard of the task board and the plan of deliverables for the coming sprint. Moreover, each member of the group should be familiar with the use of the latest Modular Compressor Architecture.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We try to use the clients' viewer, Three.js, to have a basic understanding of block viewing.
- We create wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.
- We implement the Multiprocessing as well as the Greedy-Expander Algorithm.
- We add traceability between the sprint backlog and product backlog.
- We add the estimate of time and record the actual time on task board for comparison.
- We develop the algorithm to find and prove the best compression rate.
- We add the deadline to the tasks

Summary of Changes:

In this week, we mainly focused on the refinement of the current solution based on existing software architecture developed by Karl, which aims to process the new dataset with larger size. The use of developed Greedy-Expander Algorithm and Multiprocessing have been implemented so that the efficiency of the current algorithm could be improved a lot. Besides, writing the code based on brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset was another task set for this sprint.

In the sprint review meeting this week, we have proposed some points that might contribute to team cooperation and communication, as well as the changes about agile management in the coming sprint. Firstly, for the communication, there would be a short update on the individual tasks every 3-4 days on slack. For the algorithm, we improved the performance of the algorithm on the first two datasets and tried to apply the improved algorithms on the new datasets. And there are more detailed changes considered like adding the deadline of the tasks in the task board to compare the estimated and actual spent hours. As the mention of feedback, we also add the traceability between your sprint backlog and product backlog so that to follow up the progress.

Snapshot Week 9 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:

5 **Product Backlog**

As a user, I want the compression algorithms running on a GPU which can complete more work in the same amount of time as compared to a CPU. So, it will increase the throughput of data and the number of concurrent calculations within an application.

Added by a1811518

User's reminder: We do not use csv files as the underlying format here. In real life, a binary format is used for each block that contains more data than just the label, but regardless of the format the underlying principles of the algorithm - to coalesce lots of small blocks into bigger blocks up to and including some pre-defined parent block size boundaries - are exactly the same.

Added by a1806297

As a user, I want this successful algorithm from this semester that demonstrates high speed and high compression may well end up being embedded in the DomainMCF product on AWS.

Added by a1806297

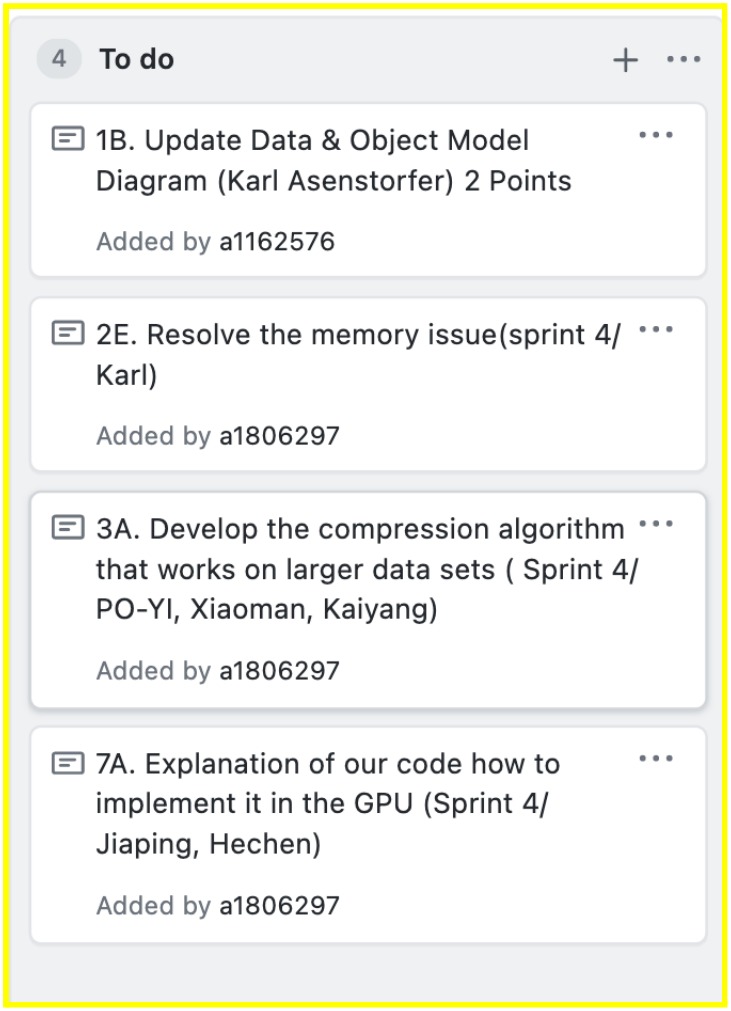
As a user, I think the first priority of this product is delivering lossless model after compression. Then it should not take too much time to process combination the of blocks. Overall, it should be kept a balance between time efficiency and compression ratio.

Added by a1804817

As a user, I want this software to be able to to compress and send down a coalesced block model instead of the raw uncompressed uniform block model that comes out of the machine learning algorithm so that I can save our customers a lot of waiting time and bandwidth.

Added by a1806297

Task Board:


4 To do
1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2E. Resolve the memory issue(sprint 4/ Karl)
Added by a1806297
3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)
Added by a1806297
7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)
Added by a1806297

7 **In progress**

1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points

Added by a1162576

2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points

The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.

Added by a1162576

2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

Added by a1806297

2D. Understand the current code structure and functions. (Yang Lu)

Added by a1782685

4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)

Added by a1806297

5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.

(Liuyang Yun, Yang Lu, Sprint 4)

Added by a1811518

7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)

Added by a1786785


36 Done
1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)
Added by a1804817
5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)
Added by a1782685
5A. Research on brute-force algorithm and dynamic programming. (6 points) Outline of how to get to the optimal compression for the 2x2x2 dataset with an illustration of a simple example on wiki page. (4 points) (Liuyang Yun, Sprint 3)
Added by a1811518
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)
Added by a1811518
2B. Develop algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)
Added by a1804817
6B. Add traceability between the goals on the sprint backlog and the tasks (Karl Asenstorfer, Liuyang Yun, Sprint 3)
Added by a1811518
36 Done
6C. Prepare the problems for the next client meeting.
Added by a1804817
3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)
Added by a1797683
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518

36 Done

Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)

Added by a1782685

Prepare the coming sprint Retrospective 1 (include individual part)

Added by a1797683

Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)

Added by a1804817

Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)

Added by a1782685

Collect the questions we gonna ask during the client meeting. (Yang Lu)

Added by a1782685

Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)

Added by a1784375

Determine the members' task in the initial phase (coding and documentation)

Added by a1806297

36 Done

Determine and set up communication channels

Added by a1162576

Determine the frequency and time of meetings for member each week

Added by a1784375

Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)

Added by a1782685

Research papers/techniques on block model compression

Added by a1784375

Research on the format and meaning of both input data and output data. (Po-Yi Lee)

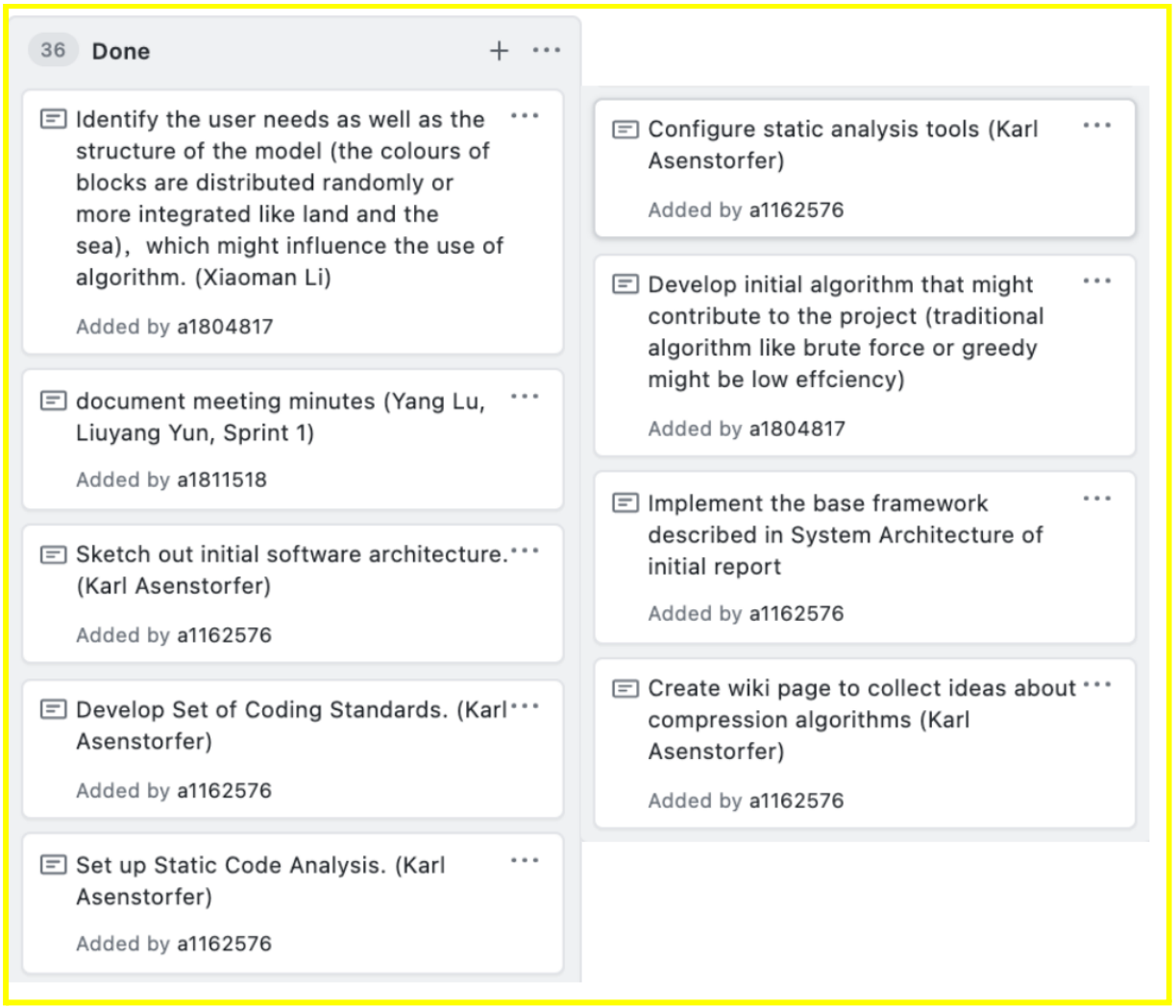
Added by a1806297

Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)

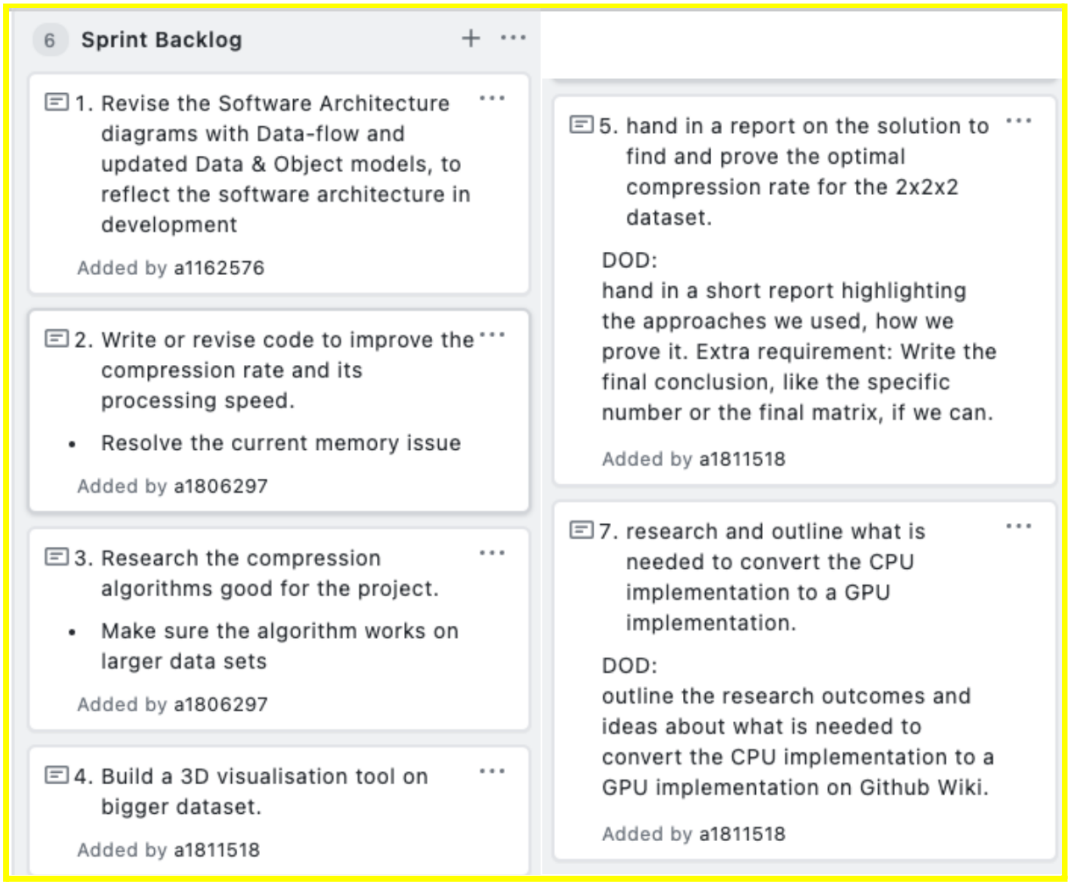
Added by a1806297

Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)

Added by a1811518


36 Done
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Implement the base framework described in System Architecture of initial report
Added by a1162576
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576

Sprint Backlog and User Stories



This week, there are both user stories rolled from the previous sprint and new user stories on the sprint backlog. We have four major goals in the next sprint: 1. Write or revise code to improve the compression rate and its processing speed, and **resolve the memory issue on the current algorithm**; 2. Develop a compression algorithm that can handle the big_one dataset; 3. Hand in a report on the solution to find and prove the optimal compression rate for the 2x2x2 dataset; 4. **research and outline what is needed to convert the CPU implementation to a GPU implementation**.

Since there are four main tasks, we should have a clear division of personnel and improve our communication in the next sprint.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We create wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.
- We implement the Multiprocessing as well as the Greedy-Expander Algorithm.
- For the task to improve the task board management: 1) add traceability between the sprint backlog and product backlog; 2) add the estimate of time and record the actual time on task board for comparison. 3) We add the deadline to the tasks.
- For the task to find the optimal compression algorithm: hand in a short report highlighting the approaches we used and how we prove it.
- For the GPU task: outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.

Summary of Changes:

Following changes happened since the last snapshot:

What we have done in this week

- We attended the sprint review and planning meeting and had a group meeting to discuss the division of personnel to fulfill the four goals for sprint 4;
- We submitted the new algorithm to Titan, and it passed the first three datasets;
- We updated the outline of how to get to the optimal compression algorithm for the 2x2x2 dataset on the GitHub wiki page.
- Yuanpeng has built a 3D visualisation tool that will work on small datasets;

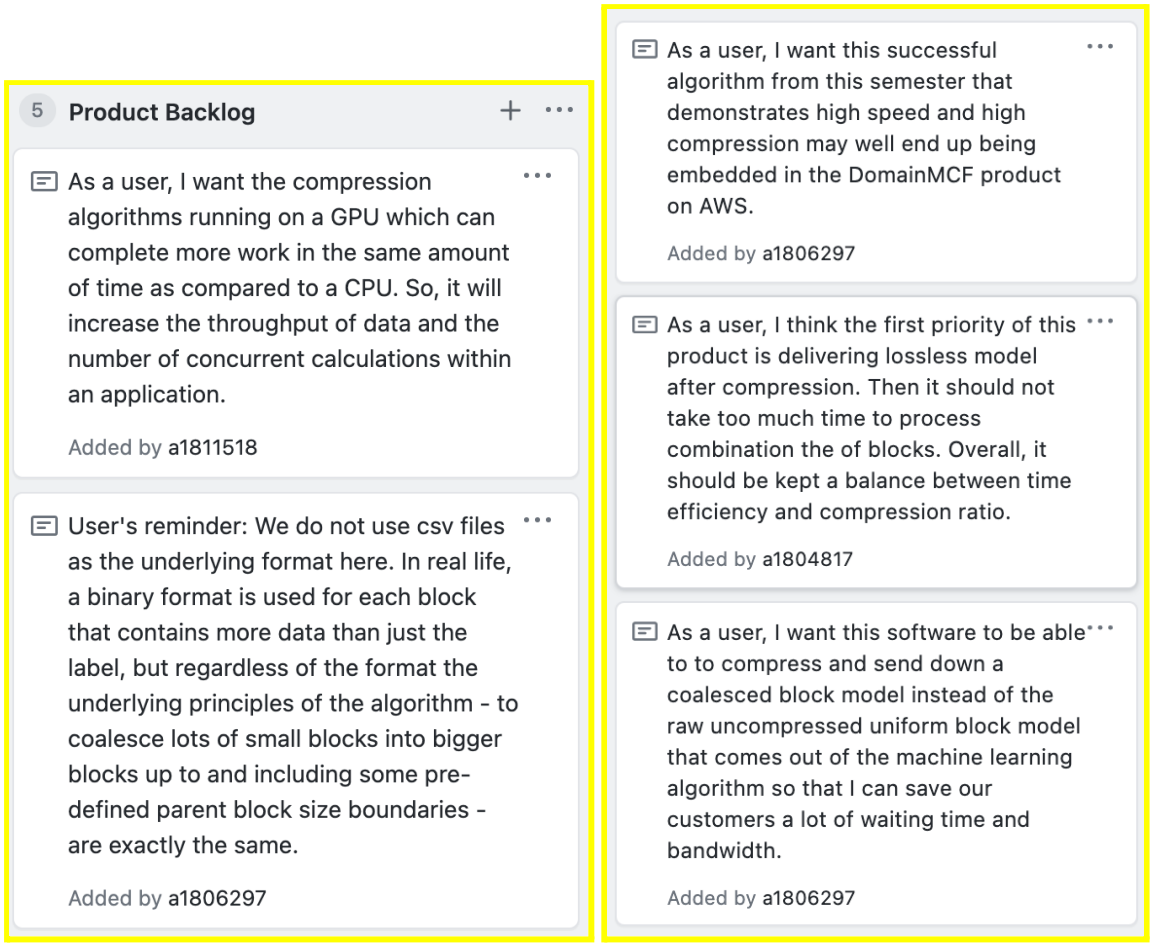
changes on client requirements:

- The ultimate solution for the project is running on a GPU. We need to research and develop some ideas on how to convert our current algorithm from a CPU implementation to a GPU implementation.

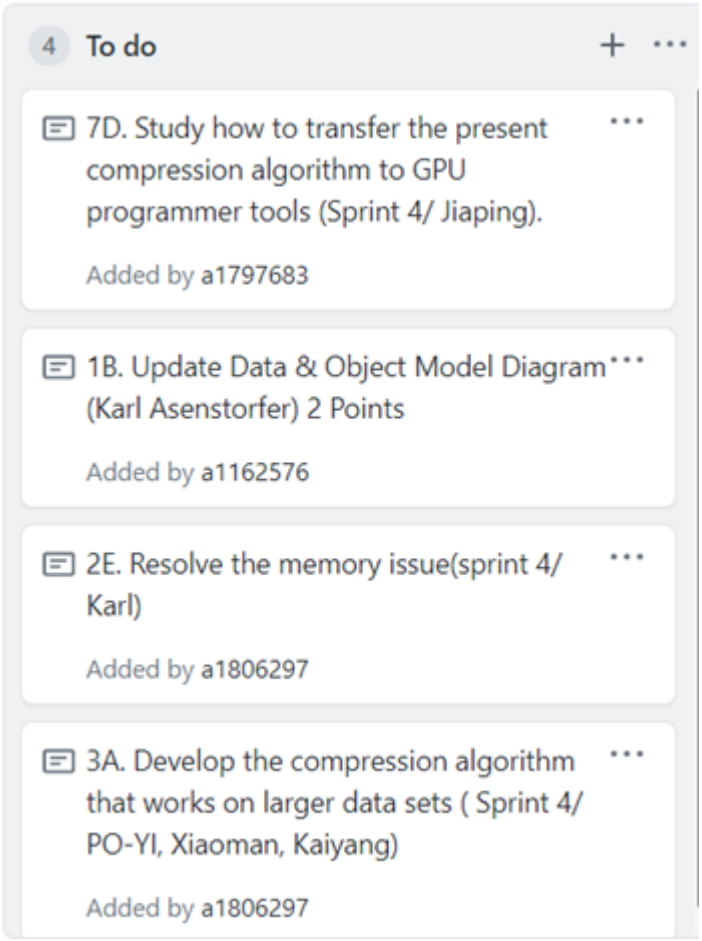
Snapshot Week 10 of Group BLOCKS7PG

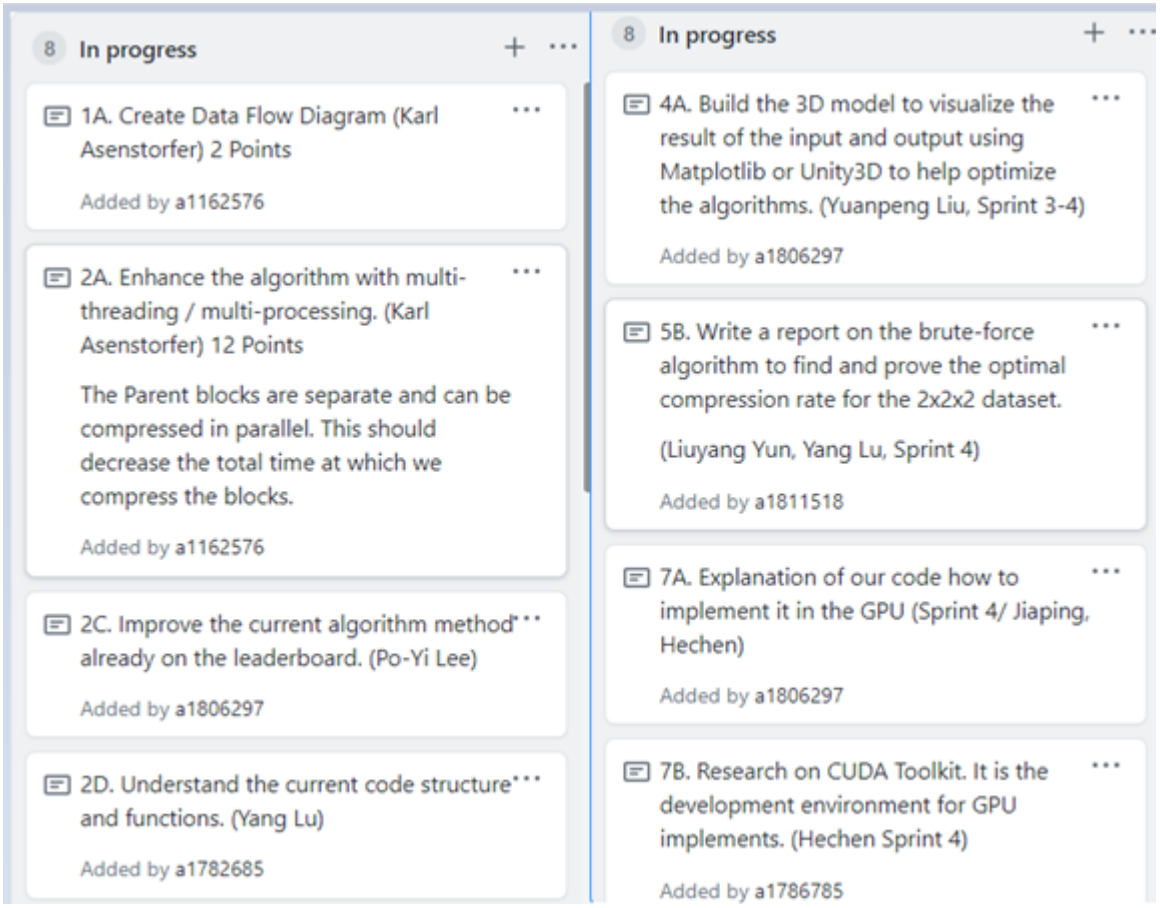
Product Backlog and Task Board

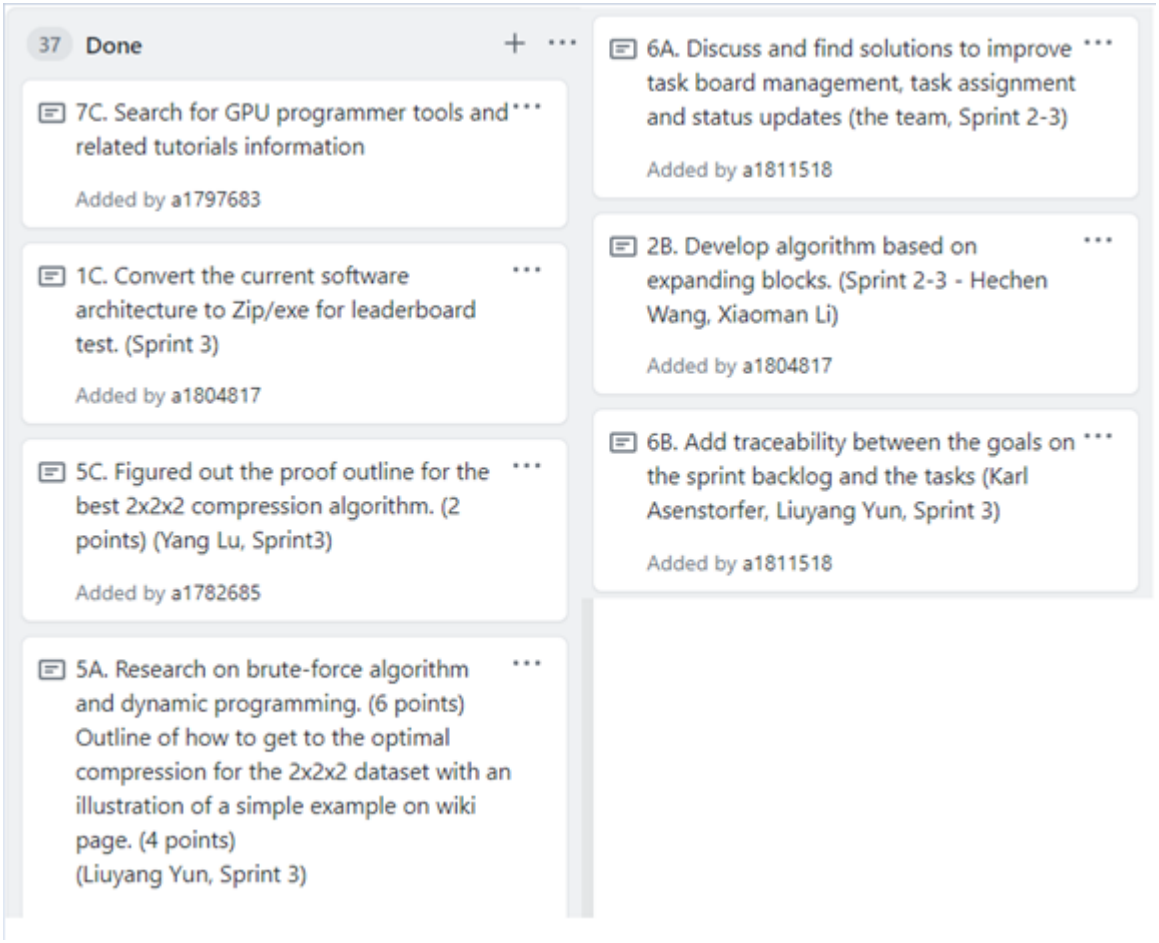
Product Backlog:

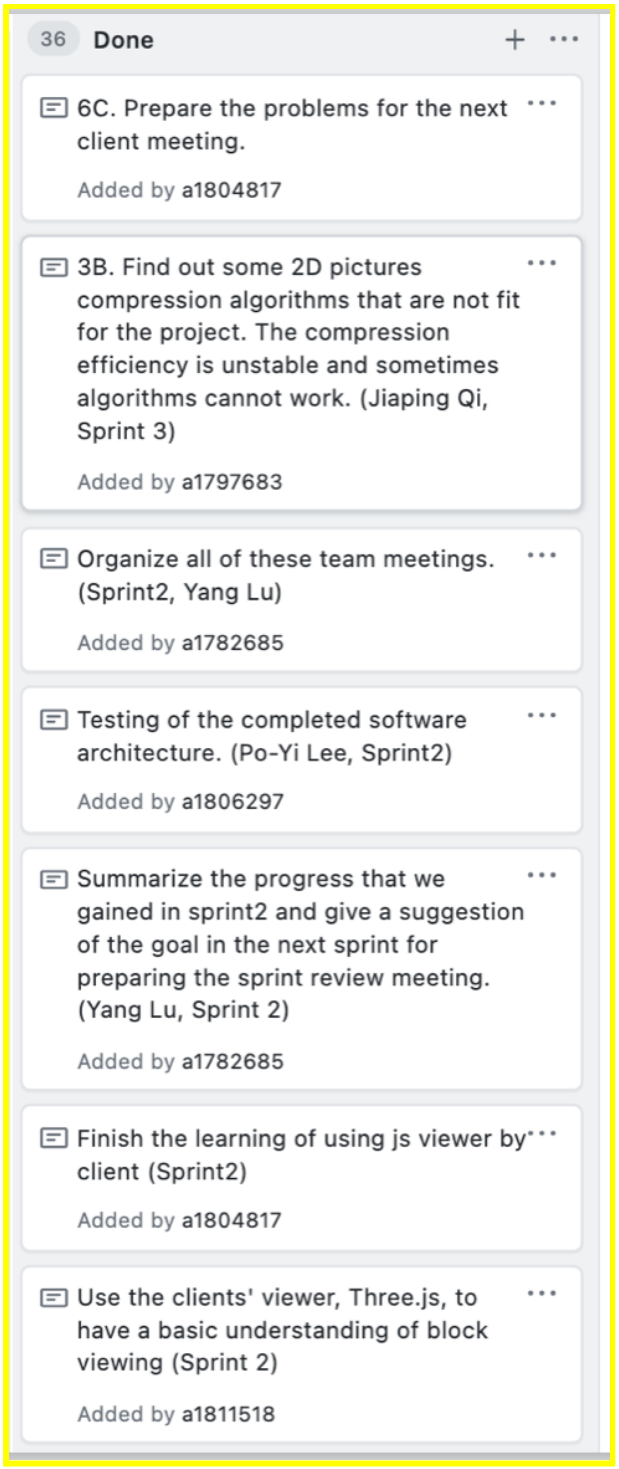


Task Board:

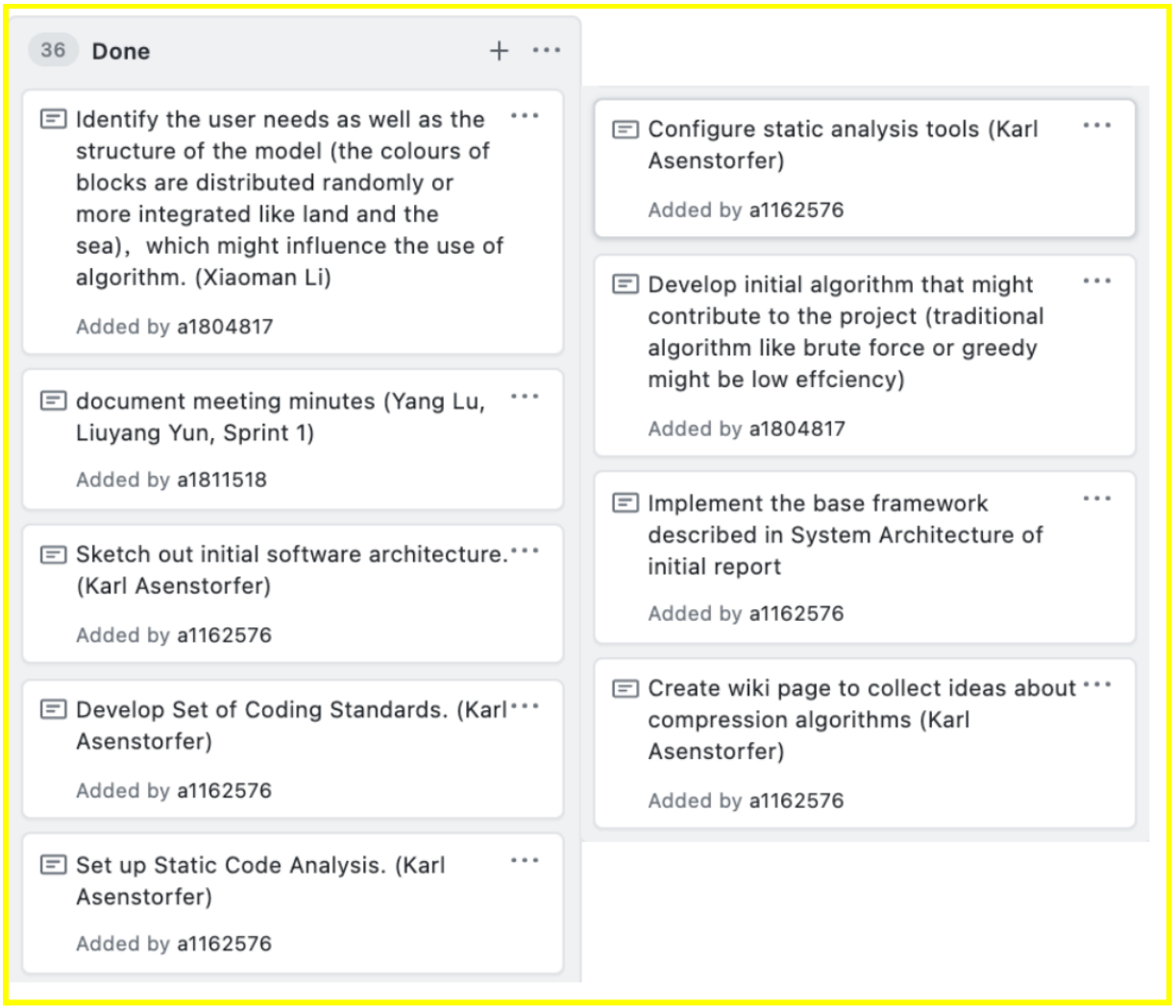

4 To do
7D. Study how to transfer the present compression algorithm to GPU programmer tools (Sprint 4/ Jiaping).
Added by a1797683
1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2E. Resolve the memory issue(sprint 4/ Karl)
Added by a1806297
3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)
Added by a1806297


8 In progress
1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
2D. Understand the current code structure and functions. (Yang Lu)
Added by a1782685
8 In progress
4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)
Added by a1806297
5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.
(Liuyang Yun, Yang Lu, Sprint 4)
Added by a1811518
7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)
Added by a1806297
7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)
Added by a1786785

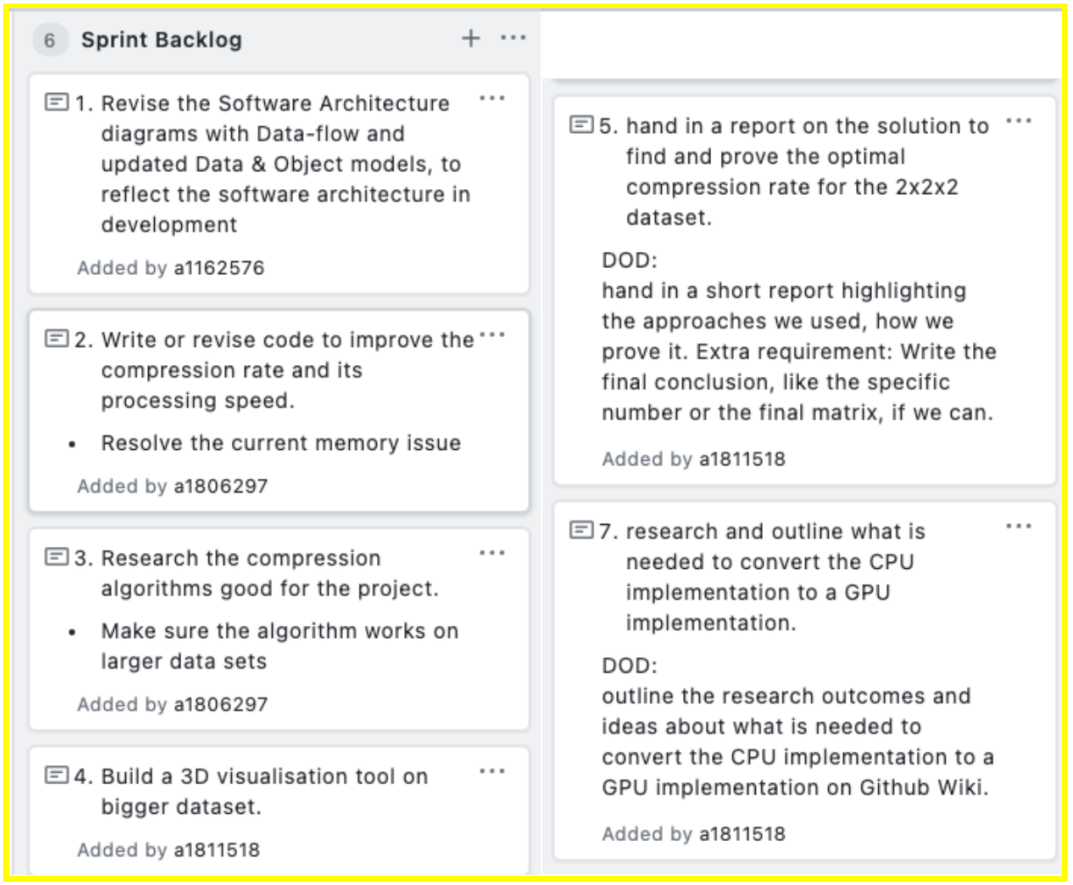

37 Done
7C. Search for GPU programmer tools and related tutorials information
Added by a1797683
1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)
Added by a1804817
5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)
Added by a1782685
5A. Research on brute-force algorithm and dynamic programming. (6 points) Outline of how to get to the optimal compression for the 2x2x2 dataset with an illustration of a simple example on wiki page. (4 points) (Liuyang Yun, Sprint 3)
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)
Added by a1811518
2B. Develop algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)
Added by a1804817
6B. Add traceability between the goals on the sprint backlog and the tasks (Karl Asenstorfer, Liuyang Yun, Sprint 3)
Added by a1811518


36 Done
6C. Prepare the problems for the next client meeting.
Added by a1804817
3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)
Added by a1797683
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518


36 Done
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
Added by a1782685
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817
Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
Added by a1782685
Collect the questions we gonna ask during the client meeting. (Yang Lu)
Added by a1782685
Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
36 Done
Determine and set up communication channels
Added by a1162576
Determine the frequency and time of meetings for member each week
Added by a1784375
Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
Added by a1782685
Research papers/techniques on block model compression
Added by a1784375
Research on the format and meaning of both input data and output data. (Po-Yi Lee)
Added by a1806297
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
Added by a1806297
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)
Added by a1811518


36 Done
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
Implement the base framework described in System Architecture of initial report
Added by a1162576
Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)
Added by a1162576

Sprint Backlog and User Stories



For this week, there are no major changes to the Sprint Backlog and User Stories. The main tasks of our team at this stage include continuously improving our compression algorithm to enhance its compression rate and speed, handle the memory issue when compressing the huge dataset, and studying for the new issues our client raised about changing the compression implementation from CPU to GPU.

According to the discussion in the last meeting, we have managed to divide team members into four groups. All of the team members are working for their goals.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We create wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.
- We implement the Multiprocessing as well as the Greedy-Expander Algorithm.
- For the task to improve the task board management: 1) add traceability between the sprint backlog and product backlog; 2) add the estimate of time and record the actual time on task board for comparison. 3) We add the deadline to the tasks.
- For the task to find the optimal compression algorithm: hand in a short report highlighting the approaches we used and how we prove it.
- For the GPU task: outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.
- We clarify the team members' division in the meeting.
- We add the visualization block to clarify the compression result.
- We add GPU programmer tools in the Github wiki.

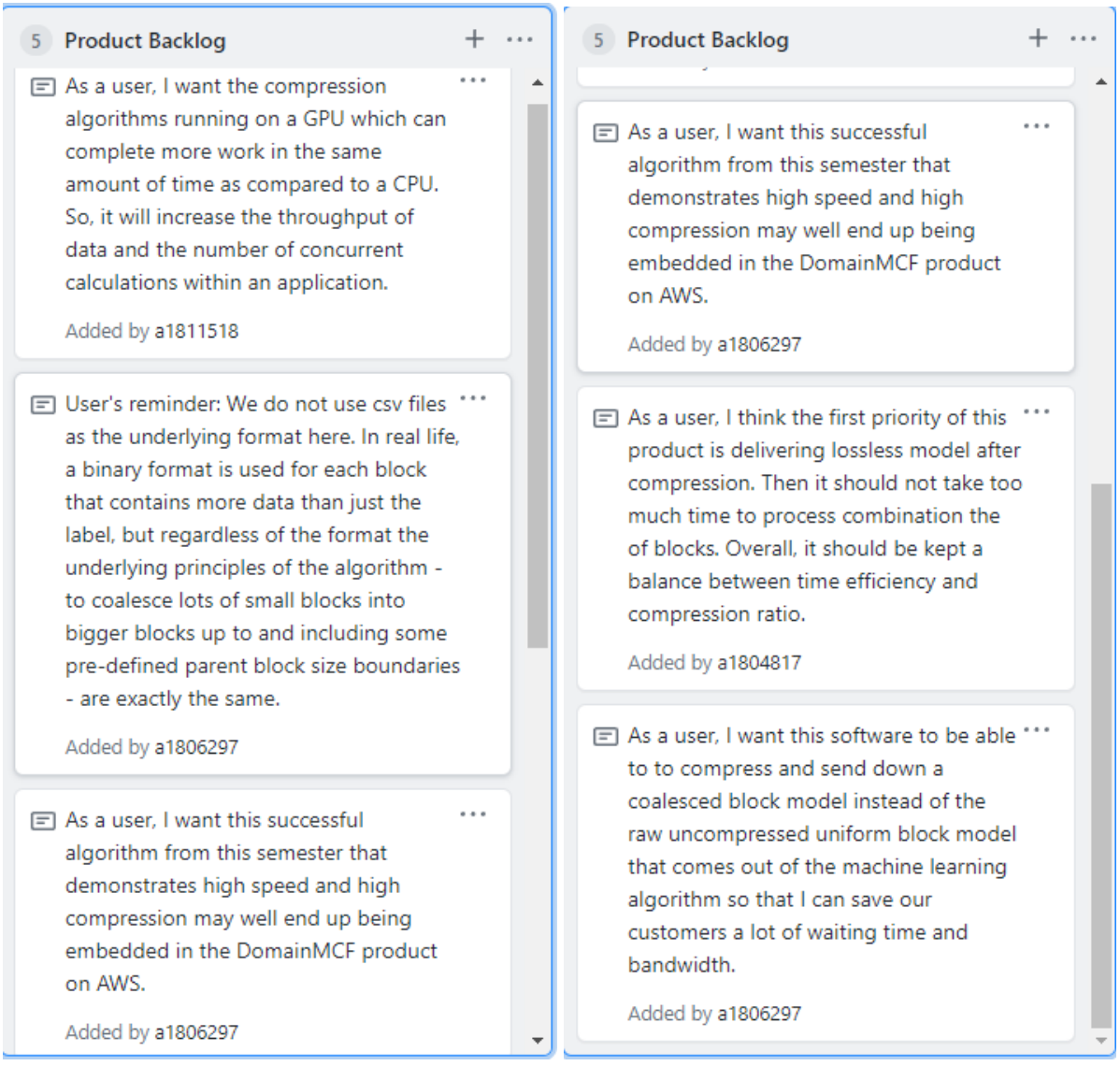
Summary of Changes:

In this week, we mainly focused on the four goals we determined in the last meeting. The improving work continues. The compression rate for the 2*2*2 of the new architecture we upload can be 79.63% now and the 4*4*4 one can be 93.74%. The visualization function could help us with observing the compression result. According to that, the algorithm team could analyze the compression figure and do improving work. The proof of the optimal compression is still working. We have uploaded the brief outline of the proof to the Github wiki. The GPU team has finished the data preparation stage and is ready to work on implementing the algorithm on GPU programmer platforms.

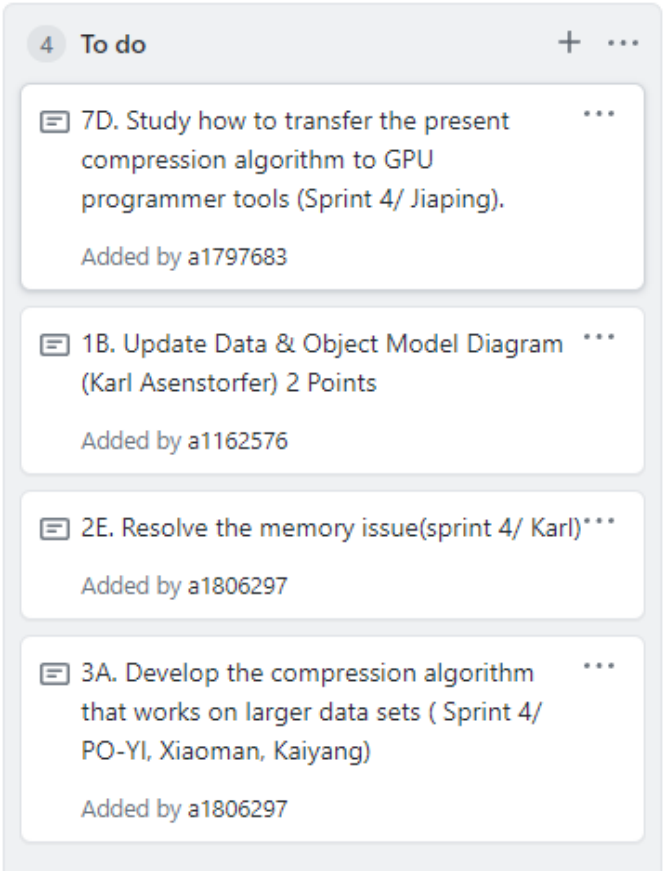
Snapshot Week 11 of Group BLOCKS7PG

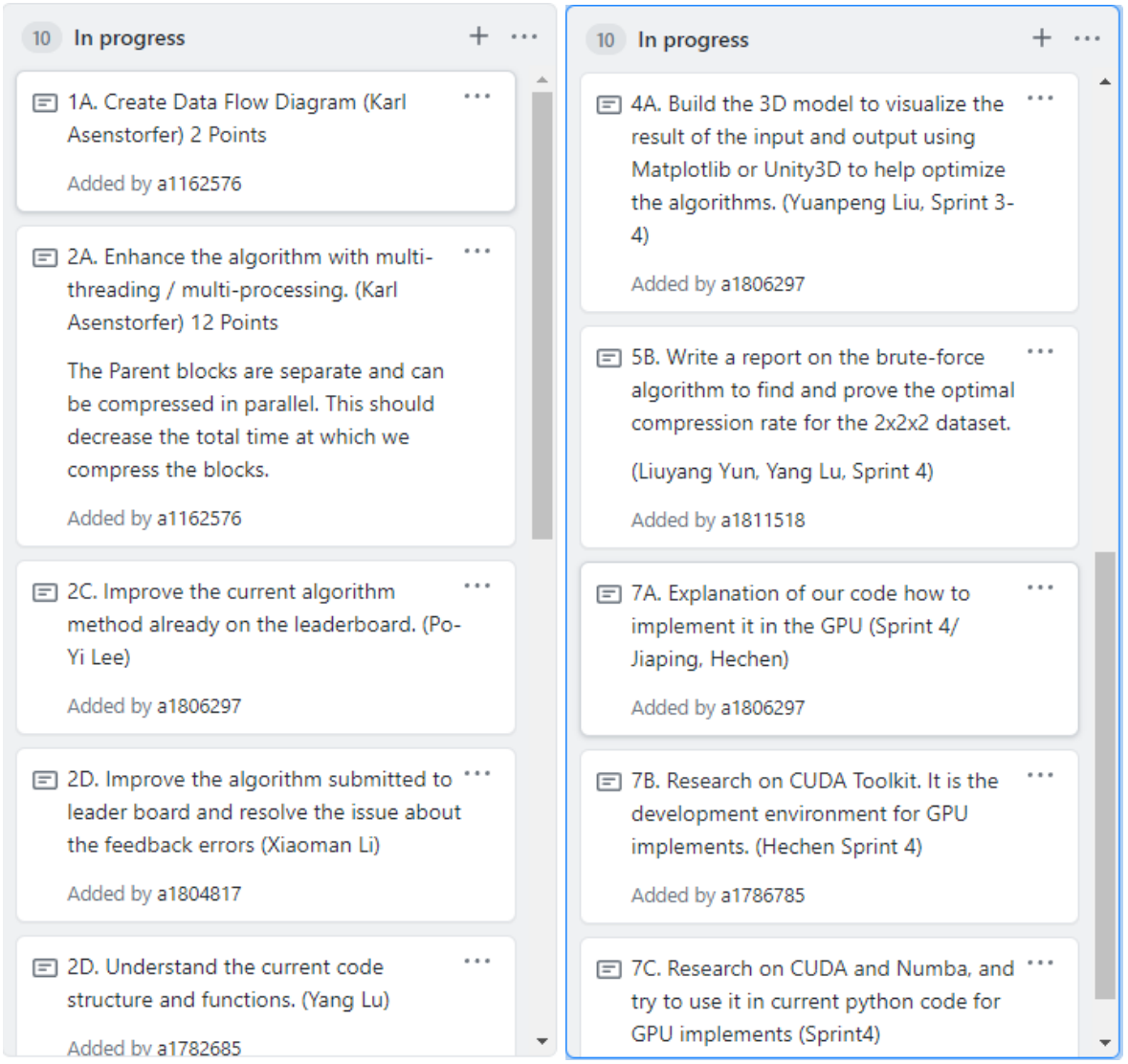
Product Backlog and Task Board

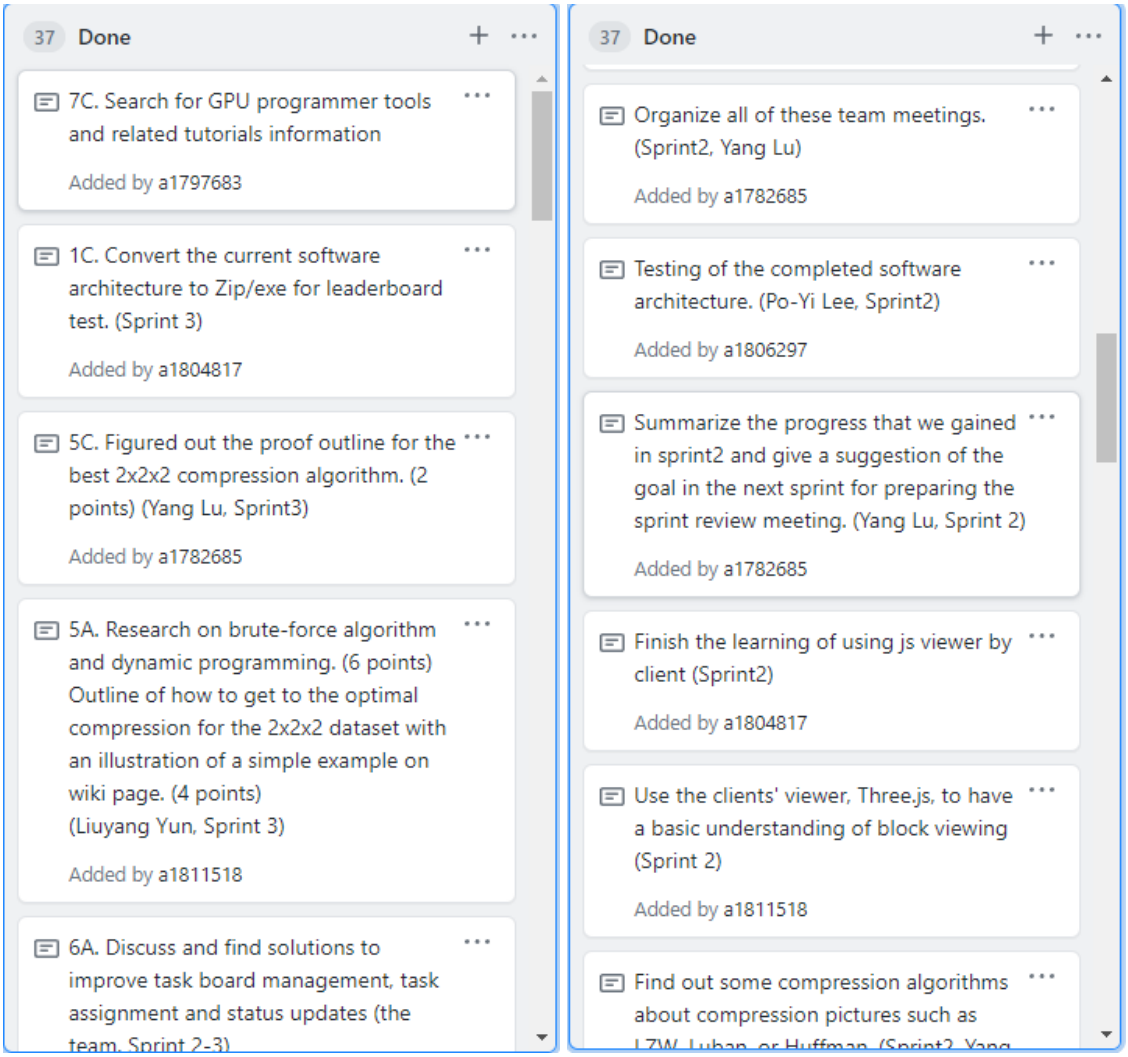
Product Backlog:

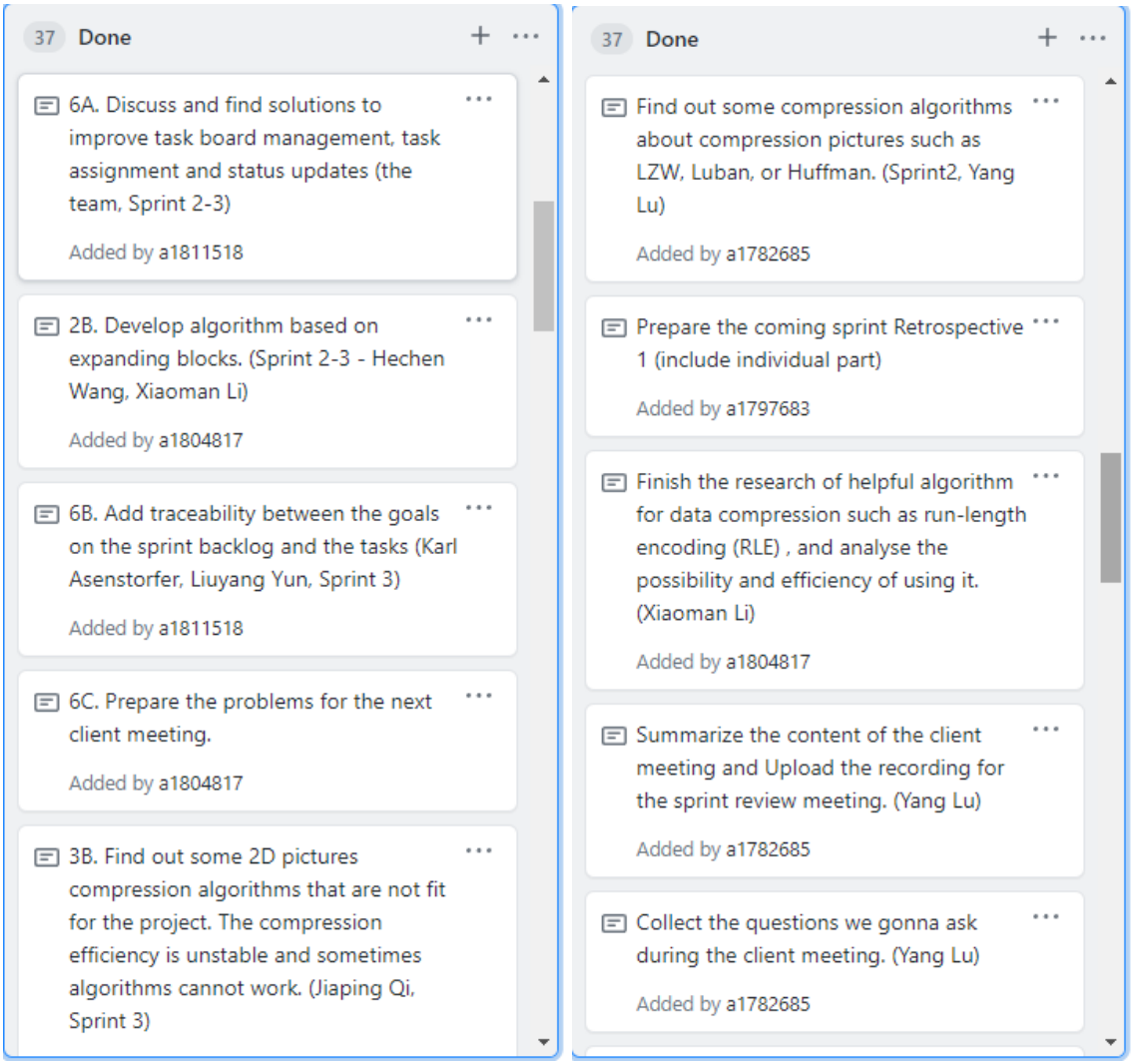


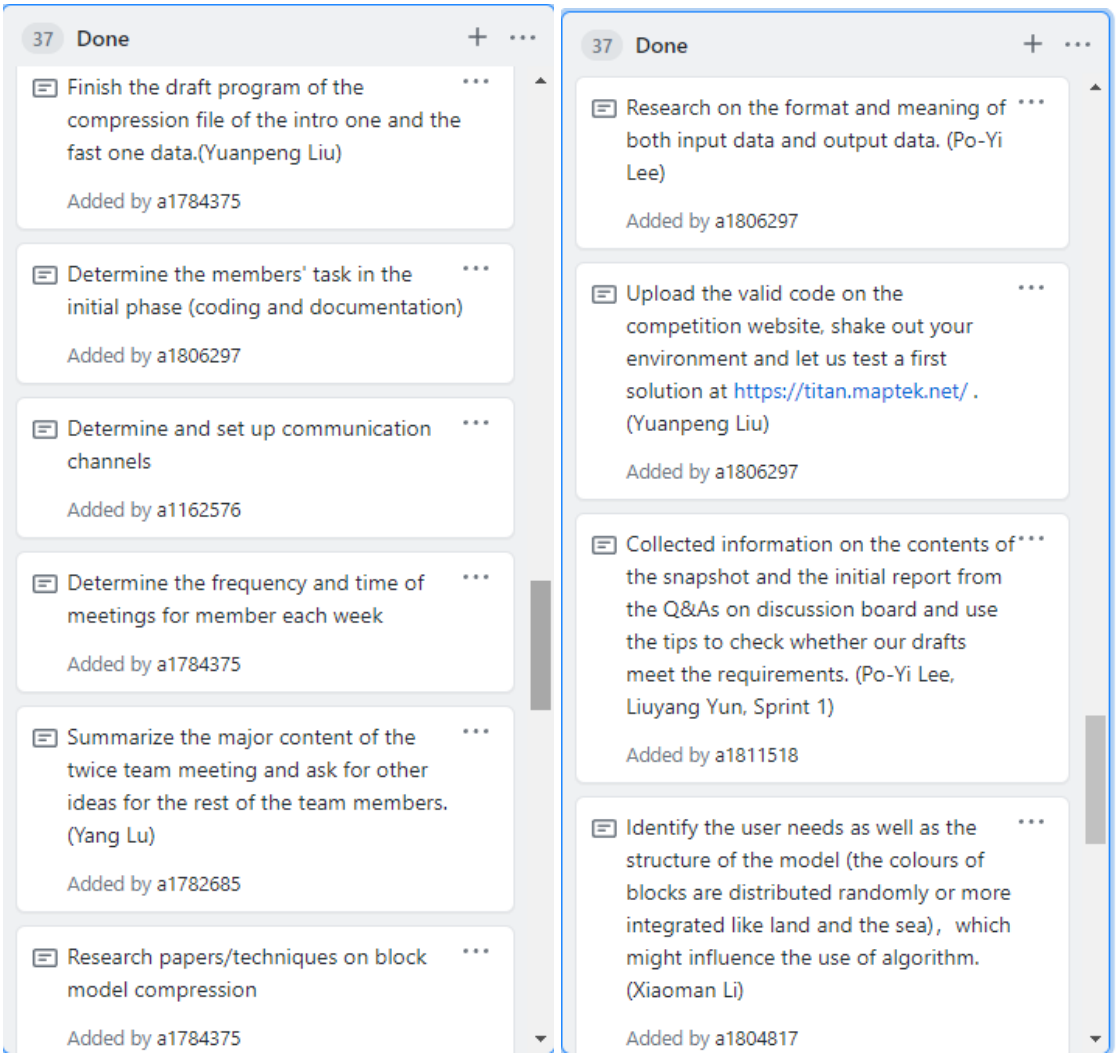
Task Board:

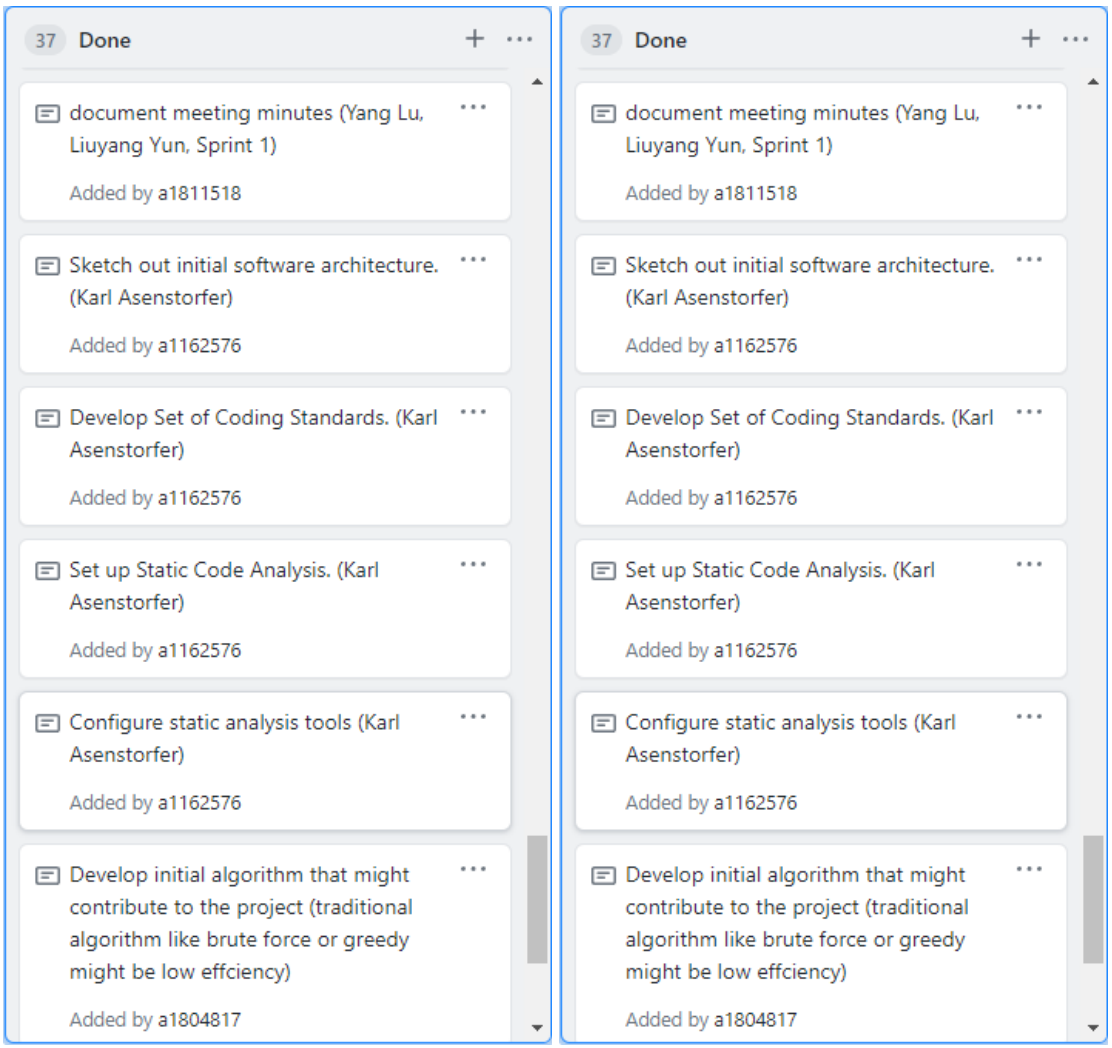

4 To do
7D. Study how to transfer the present compression algorithm to GPU programmer tools (Sprint 4/ Jiaping).
Added by a1797683
1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2E. Resolve the memory issue(sprint 4/ Karl)
Added by a1806297
3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)
Added by a1806297


10 In progress
1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
2D. Improve the algorithm submitted to leader board and resolve the issue about the feedback errors (Xiaoman Li)
Added by a1804817
2D. Understand the current code structure and functions. (Yang Lu)
Added by a1782685
10 In progress
4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)
Added by a1806297
5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.
(Liuyang Yun, Yang Lu, Sprint 4)
Added by a1811518
7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)
Added by a1806297
7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)
Added by a1786785
7C. Research on CUDA and Numba, and try to use it in current python code for GPU implements (Sprint4)

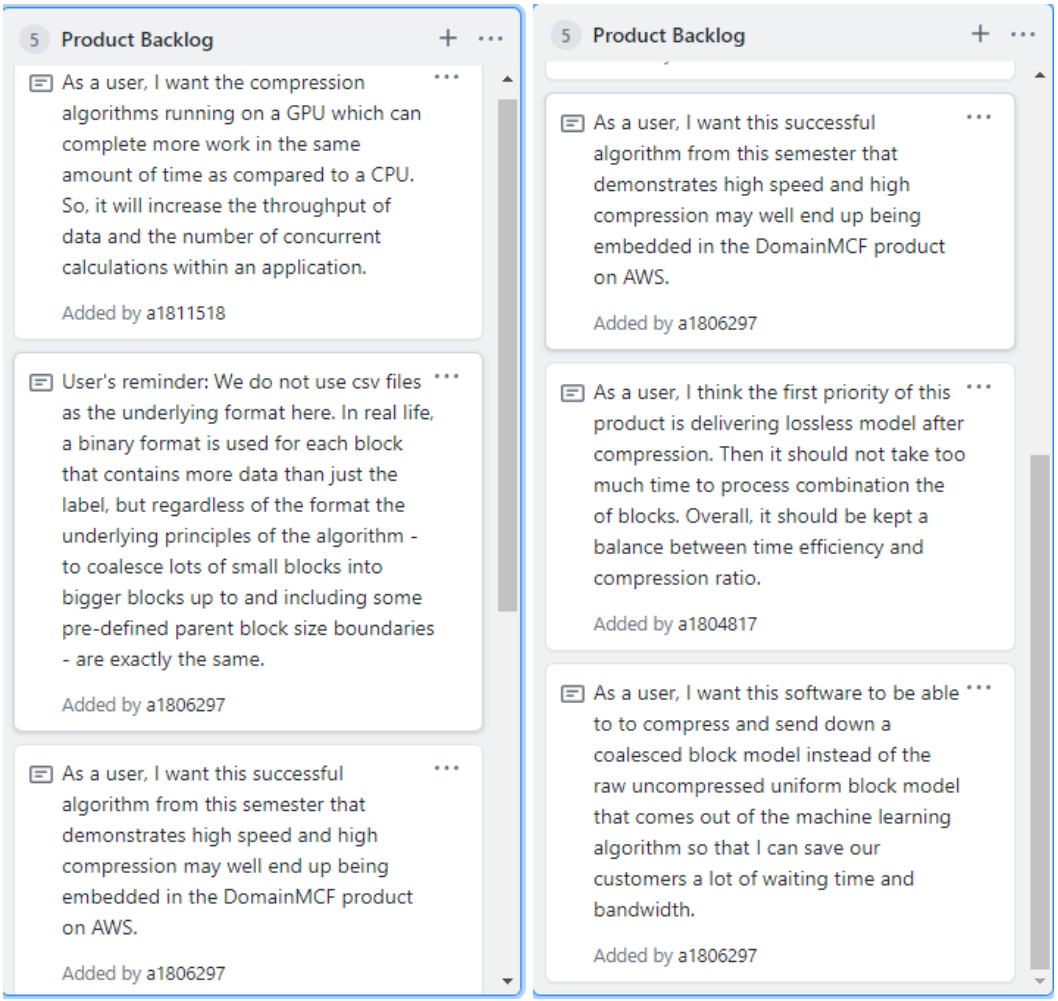

37 Done
7C. Search for GPU programmer tools and related tutorials information
Added by a1797683
1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)
Added by a1804817
5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)
Added by a1782685
5A. Research on brute-force algorithm and dynamic programming. (6 points) Outline of how to get to the optimal compression for the 2x2x2 dataset with an illustration of a simple example on wiki page. (4 points)
(Liuyang Yun, Sprint 3)
Added by a1811518
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the
37 Done
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518
Find out some compression algorithms about compression pictures such as


37 Done
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)
Added by a1811518
2B. Develop algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)
Added by a1804817
6B. Add traceability between the goals on the sprint backlog and the tasks (Karl Asenstorfer, Liuyang Yun, Sprint 3)
Added by a1811518
6C. Prepare the problems for the next client meeting.
Added by a1804817
3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)
37 Done
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
Added by a1782685
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817
Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
Added by a1782685
Collect the questions we gonna ask during the client meeting. (Yang Lu)
Added by a1782685


37 Done
Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
Added by a1784375
Determine the members' task in the initial phase (coding and documentation)
Added by a1806297
Determine and set up communication channels
Added by a1162576
Determine the frequency and time of meetings for member each week
Added by a1784375
Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
Added by a1782685
Research papers/techniques on block model compression
Added by a1784375
37 Done
Research on the format and meaning of both input data and output data. (Po-Yi Lee)
Added by a1806297
Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
Added by a1806297
Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)
Added by a1811518
Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
Added by a1804817


37 Done
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817
37 Done
document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
Added by a1811518
Sketch out initial software architecture. (Karl Asenstorfer)
Added by a1162576
Develop Set of Coding Standards. (Karl Asenstorfer)
Added by a1162576
Set up Static Code Analysis. (Karl Asenstorfer)
Added by a1162576
Configure static analysis tools (Karl Asenstorfer)
Added by a1162576
Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)
Added by a1804817

Sprint Backlog and User Stories



For this week, client released a new worldly dataset. As the last dataset, it has less data than the largest data set, so next we only need to modify the existing architecture to achieve the successful operation and compression of the dataset.

In addition, we also need to improve the existing architecture to solve the memory problem of the architecture. Existing architectures will cause excessive memory usage when running large datasets. For the 2x2x2 dataset algorithm, we may need to analyse the results in combination with visualization and find an uncompressed error result due to algorithm problems to achieve the highest compression rate required by client.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.
- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We create wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.

- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.
- We implement the Multiprocessing as well as the Greedy-Expander Algorithm.
- For the task to improve the task board management: 1) add traceability between the sprint backlog and product backlog; 2) add the estimate of time and record the actual time on task board for comparison. 3) We add the deadline to the tasks.
- For the task to find the optimal compression algorithm: hand in a short report highlighting the approaches we used and how we prove it.
- For the GPU task: outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.
- We clarify the team members' division in the meeting.
- We add the visualization block to clarify the compression result.
- We add GPU programmer tools in the Github wiki.
- We complete the draft of proof report and sent to client, waiting for feedback.

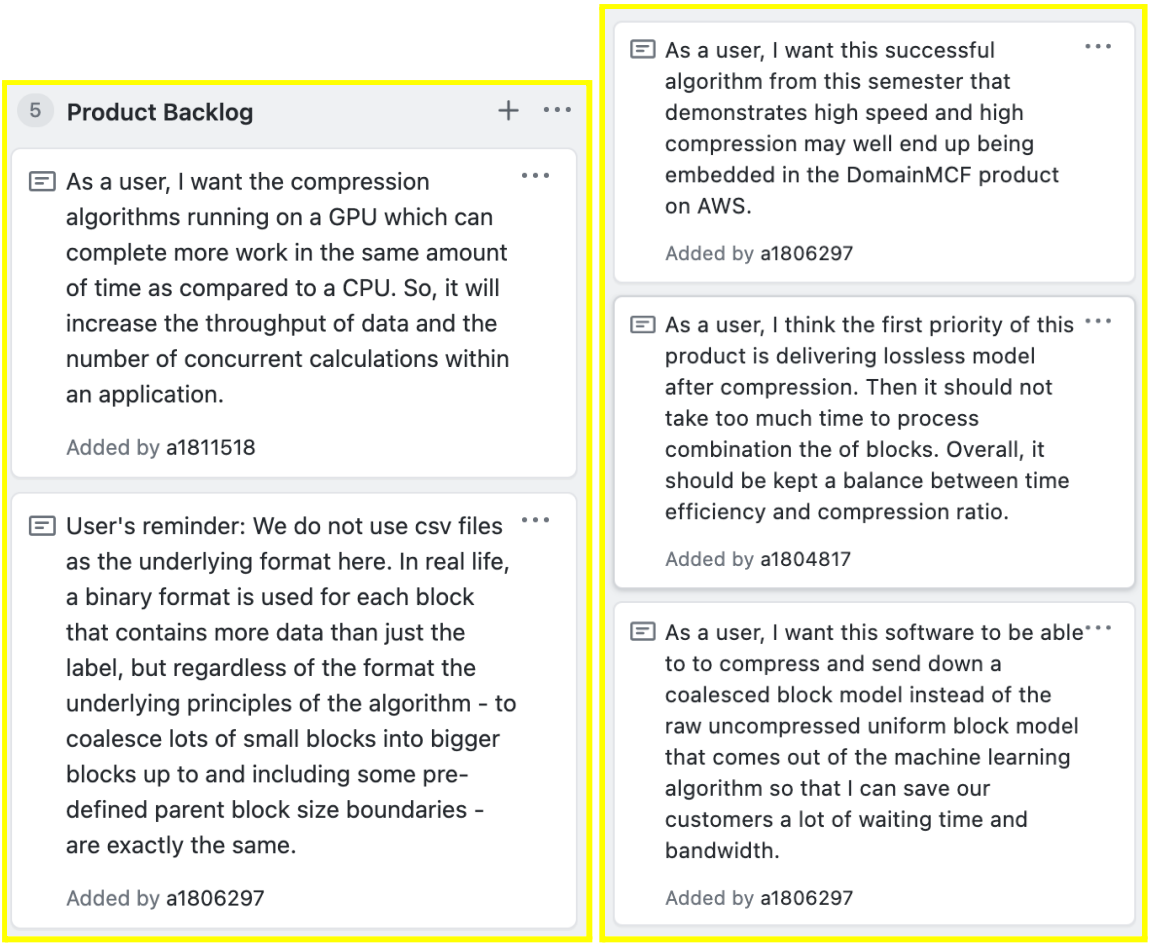
Summary of Changes:

In this week, we mainly focused on improving our current code to make sure that it could compress the larger data sets. The compression rate for the 2*2*2 of the new architecture we upload can be 79.63% now and the 4*4*4 one can be 93.74%. So far our code has been able to run successfully on the first few data sets, and we are working on memory issues that may arise when it is applied to larger data sets. On the visualization side, we are now able to visualize the compression junctions of small data sets and try to use the visualization results to find the parts that have not been successfully compressed to improve the compression rate of the code. We submitted the outline of the proof of the best compression for the 2*2*2 dataset to the client and wait for feedback then we will improve the draft according to the feedback. In terms of how code runs on a GPU environment, we found a tool and decided to try it out on a GPU environment.

Snapshot Week 12 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:


5
Product Backlog
As a user, I want the compression algorithms running on a GPU which can complete more work in the same amount of time as compared to a CPU. So, it will increase the throughput of data and the number of concurrent calculations within an application.
Added by a1811518
User's reminder: We do not use csv files as the underlying format here. In real life, a binary format is used for each block that contains more data than just the label, but regardless of the format the underlying principles of the algorithm - to coalesce lots of small blocks into bigger blocks up to and including some pre-defined parent block size boundaries - are exactly the same.
Added by a1806297
As a user, I want this successful algorithm from this semester that demonstrates high speed and high compression may well end up being embedded in the DomainMCF product on AWS.
Added by a1806297
As a user, I think the first priority of this product is delivering lossless model after compression. Then it should not take too much time to process combination the of blocks. Overall, it should be kept a balance between time efficiency and compression ratio.
Added by a1804817
As a user, I want this software to be able to to compress and send down a coalesced block model instead of the raw uncompressed uniform block model that comes out of the machine learning algorithm so that I can save our customers a lot of waiting time and bandwidth.
Added by a1806297

Task Board:


3 To do
7E. Optimise the Numba code to run on a GPU, for the best possible performance (Sprint 5) 12 points
Added by a1162576
7D. Study how to transfer the present compression algorithm to GPU programmer tools (Sprint 4/ Jiaping).
Added by a1797683
3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)
Added by a1806297

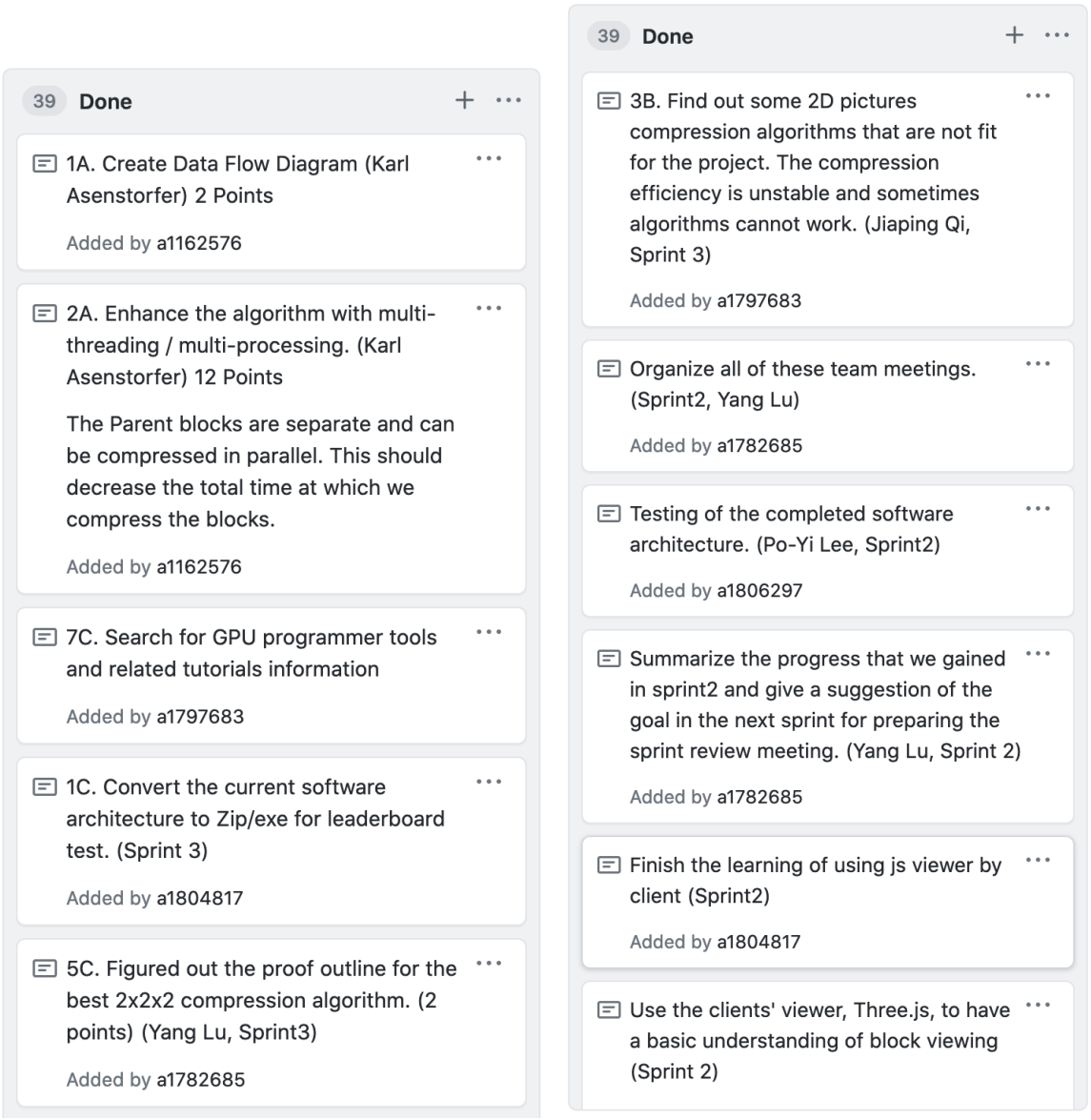
12 **In progress**

- 1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points
  Added by a1162576
- 2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
  Added by a1806297
- 2D. Improve the algorithm submitted to leader board and resolve the issue about the feedback errors (Xiaoman Li)
  Added by a1804817
- 2D. Understand the current code structure and functions. (Yang Lu)
  Added by a1782685
- 2E. Resolve the memory issue(sprint 4/ Karl)
  Added by a1806297
- 4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)
  Added by a1806297
- 7D. Port the compression engine to use the Numba JIT native CPU compiler, to increase performance. (Sprint 5/ Karl Asenstorfer) 6 points
  Added by a1162576

12 **In progress**

- 5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.
  (Liuyang Yun, Yang Lu, Sprint 4)
  Added by a1811518
- 5C. Update the algorithm to be optimal for 2x2x2 Compression. (Sprint 5 / Liuyang Yun, Yang Lu, Karl Asenstorfer)
  - Identify and implement work arounds for the edge cases holding back the algorithm from optimal compression.

  Added by a1162576
- 7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)
  Added by a1806297
- 7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)
  Added by a1786785
- 7C. Research on CUDA and Numba, and try to use it in current python code for GPU implements (Sprint4)
  Added by a1804817


39 Done
1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points
Added by a1162576
2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
7C. Search for GPU programmer tools and related tutorials information
Added by a1797683
1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)
Added by a1804817
5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)
Added by a1782685
39 Done
3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)
Added by a1797683
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)

39 **Done**

- Determine the members' task in the initial phase (coding and documentation)
  Added by a1806297
- Determine and set up communication channels
  Added by a1162576
- Determine the frequency and time of meetings for member each week
  Added by a1784375
- Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
  Added by a1782685
- Research papers/techniques on block model compression
  Added by a1784375
- Research on the format and meaning of both input data and output data. (Po-Yi Lee)
  Added by a1806297
- Upload the valid code on the competition website, shake out your environment and let us test a first solution at

39 **Done**

- Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ .(Yuanpeng Liu)
  Added by a1806297
- Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)
  Added by a1811518
- Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)
  Added by a1804817
- document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)
  Added by a1811518
- Sketch out initial software architecture. (Karl Asenstorfer)
  Added by a1162576

39 **Done**

(Karl Asenstorfer)

Added by a1162576

Develop Set of Coding Standards. (Karl Asenstorfer)

Added by a1162576

Set up Static Code Analysis. (Karl Asenstorfer)

Added by a1162576

Configure static analysis tools (Karl Asenstorfer)

Added by a1162576

Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)

Added by a1804817

Implement the base framework described in System Architecture of initial report

Added by a1162576

Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)

Added by a1162576

39 **Done**

Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)

Added by a1782685

Prepare the coming sprint Retrospective 1 (include individual part)

Added by a1797683

Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)

Added by a1804817

Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)

Added by a1782685

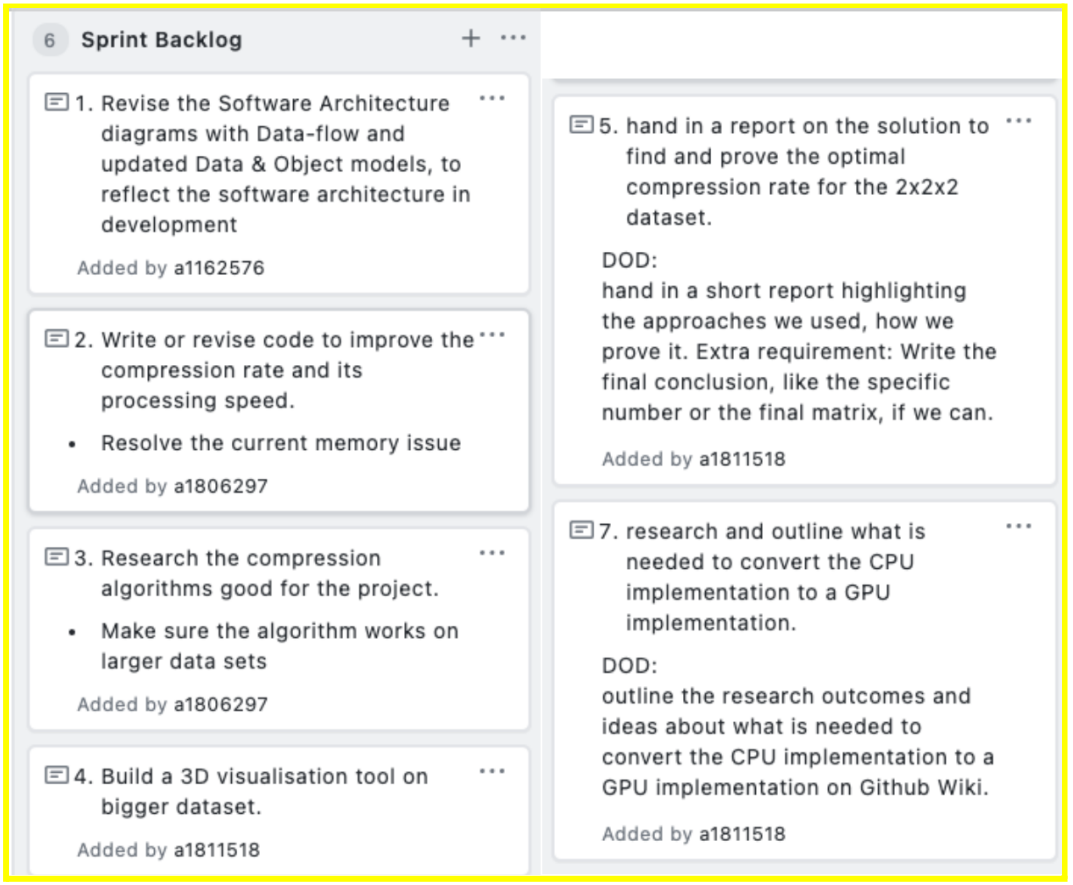
Collect the questions we gonna ask during the client meeting. (Yang Lu)

Added by a1782685

Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)

Added by a1784375

Sprint Backlog and User Stories



This week again had no major changes to the Sprint Backlog and User Stories. The tasks the team undertook this week included improving the algorithm to get around the edge-cases holding it back from optimal compression, and porting the back-end to use the Numba python module to allow compiling some functions to native code, and also solve the memory issue for large datasets. The team also researched how to use Numba to convert the algorithm to run on a GPU as requested by the client.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- All research must be shared on the GitHub wiki and the team informed via the Slack Channel.
- All draft documents shall be shared on the Slack channel with requests for comments. Once all comments have been reviewed a second draft is issued. If no team members raise any issues it is finalised.

Summary of Changes:

This week saw the proof of the optimal compression in the 2x2x2 case progress to a submitted draft. The two subgroups worked out the edge-cases and implemented work arounds/ improvements to the algorithm to reach optimal compression (79.63%). The memory issue is being resolved with using native code compiled sections of the algorithm, which increases speed and reduces the amount of memory used. This allowed the algorithm to run on larger datasets. The Numba module is also being used to port the algorithm to run on the GPU.

Personal Reflection on Software Process

Since the main project goal is to develop an optimal algorithm to compress the 3D model blocks, our team could establish the product backlog and plan the details of the sprint backlog. We followed the agile approach to develop the software in the five sprints and had the scrum meetings every week to update the change of the customer's requirements and report the progress since the customer would offer different sizes of the datasets and the requirements changed a bit in each sprint. For example, the customer proposed the implementation of model compression on the GPU at the end of the project.

In the first sprint, we managed to write the model compression algorithm on the first two datasets from scratch. This made our team get a feel for the problem before going out and looking at the research of the other optimal compression algorithms. In the sprint 2, in order to design better algorithms, we developed the software framework including a development environment which can help us to design algorithm design and the testing of the speed and compression rate. In the sprint 3, we developed the multiprocessing module to help improve the compression speed ratio. In the meantime, the research team had been looking for useful and feasible papers to help the team's developers write the code. These tasks were all achieved with the task board, sprint backlog and product backlog discussed by the team and the customer.

The customer had a high involvement with the team and was always answering our questions about the requirements on Slack. In addition, the customer had a sprint meeting every two weeks to update their requirements and explained the details. Therefore, the features of an agile approach including being highly dependent on customer involvement and the requirements being extremely responsive to change matched the situations in this project.

Having an experience of this project, I learnt how the software engineering project works in an agile scrum way which needs to focus on managing iterative development. In addition, the role of a scrum master is crucial to manage the team and increase the efficiency of tasks in terms of the scrum project. For the future project, I should be more familiar with which type of the project and how customer's involvement could affect the decision of assigning tasks, product backlog and sprint backlog.