

Sprint 1 Retrospective

Team 18 : Brendan Raftery, Kate Lorenzen, Vinson Luo, Jonathan Grider

# What went well?

### **General**

* Communication channels were effective
  + Throughout the Sprint our group utilized multiple forms of communication, including version control systems (GitHub), online text and voice chat services (Facebook, Discord) and in-person verbal communication. When our group members used these tools, we were able to coordinate our progress on tasks and user stories, set up meetings, finalize weekly reports, and ensure that all members were on the same page.
* Tasks that could be completed individually were completed with few to no issues
  + Several of our user stories during this sprint involved tasks that could be implemented without the others, with the different tasks only coming together to complete a user story. This allowed our group to work with little initial dependence on other group members’ progress.

# Tasks

The following are User Stories that we have successfully implemented in Sprint 1:

* User Story # 1: As a user, I would like to be able to see all of the elements and compounds that I have created within my workspace represented visually as an atom or collection of atoms.
  + Task 1: Create Art Assets needed for Buddies
    - Visual sprites were created on a separate illustrator program and imported into the project folder, where they were used in prefabs.
  + Task 2: Create Cosmic Ranch Class
    - A C# script was written to interact with game objects inside the workspace of the application and the workspace itself.
  + Task 3: Create Basic Buddy Class
    - A C# script was written that contains all the fields that each Buddy that appears in the workspace, and stored in the backpack, will inherit, and its information is required for regeneration in runtime when a user returns to the workspace.
  + Task 4: Create Algorithm to generate Buddy Structures
    - A C# script was written that reads input from other scripts and generate the newly required Buddy structure, display it on the workspace of the application, and store relevant information within our dynamic data structure.

The following are tasks that we have successfully implemented in Sprint 1 whose encompassing User Story was not completed (I.e., Tasks 1 through 4 and 6 through 9 of User Story 2 were completed, but Task 5 was not completed):

* User Story # 2: As a user, I would like to be able to perform Fusion on two atoms to create a new element.
  + Task 1: Create Art Assets needed for the Fusion menu tabs
    - Visual assets for panels, buttons, text, and icon were created separately in an illustrator program or the Unity game engine itself.
  + Task 2: Create Action Bar Class
    - A C# script was written to keep track of the available actions a user can perform, as well as the necessary redirections to the corresponding algorithms. These redirections occur upon the user’s click on the button representing the desired function to perform.
  + Task 3: Create Algorithm for the Action Bar UI
    - UI components were created with fully implemented buttons, self-populating scroll lists that are animated inside the game engine. The animations are programmed with a C# script that acts as a component of the UI.
  + Task 4: Create Algorithm to perform fusion
    - A C# script was written to perform “fusion” of two selected atoms within the workspace of the application. The selected atoms would be used to determine the next appropriate fusion product, obtain its information via SQLite query, and store the newly acquired information in a new Buddy object as well as in our dynamic data structure.
  + Task 6: Create art asset for reward system after performing an action
    - Required text and button visual assets were created within the game engine itself and with a separate illustrator program.
  + Task 7: Create animation for reward system after performing an action
    - The animation and custom particle effects were created using the Unity game engine.
  + Task 8: Set up a database to hold all of the needed Trium data that is needed to perform Fusion, Grouping, and Reactions
    - SQLite executables for Windows, OS X and Android were integrated into the project, in addition to necessary library and DLL files. The C# scripts defined within these files were used to interact with a SQLite database file that was created and populated outside of the Unity Engine.
* User Story # 3: As a user, I would like to be able to perform Group on two or more atoms to create a single-element molecule.
  + Task 1: Create Art Assets needed for the Group menu tabs
    - Visual assets for panels, buttons, text, and icons were created in a separate illustrator program as well as the Unity game engine itself.
* User Story # 4: As a user, I would like to be able to perform a Reaction on two or more Triums to create a compound.
  + Task 1: Create Art Assets needed for the Reaction menu tabs
    - Visual assets for panels, buttons, text, and icons were created in a separate illustrator program as well as the Unity game engine itself.
* User Story #5: As a user, I would like to be able to collect atoms representing the naturally occurring elements.
  + Task 1: Create Wormhole Class
    - A C# script was written to provide relevant logic used when the wormhole UI element was clicked by the user. It includes a range of atoms that can be generated, information to limit that range, and other utility functions used during atom generation.
  + Task 3: Create algorithm for randomly generating atoms
    - The Wormhole C# script was modified to create an atom generating algorithm. The algorithm accesses information regarding a randomly selected atom from the database, creates the newly required Buddy structure, displays it within the workspace of the application, and stores relevant information within our dynamic data structure.
  + Task 4: Connect atom generation to the Cosmic Ranch and in turn the User’s Backpack
    - Existing C# scripts were modified to update our dynamic data structure and display Buddy objects within the workspace of the application to indicate that more atoms had been “generated.”
* User Story # 6: As a user, I would like to be able to refer to a glossary for all of the Triums discovered thus far.
  + Task 1: Create the StateHandler class
    - A C# script was written to keep track of the user state, which helps coordinate the algorithms and actions that are necessary to perform when that specific state is active.
  + Task 2: Create the State class
    - A C# script was written that contains all the required fields that a State will have in order to coordinate making transitions and performing actions between States.
  + Task 3: Create the Glossary class
    - A C# script was written to allow entering and leaving the Glossary scene, as well as algorithms to generate tabs at runtime and edit its contents spontaneously in edit time.
  + Task 4: Create an Algorithm to sort the list of Triums
    - The Glossary C# script was modified to include two new methods that sorts Triums the user has acquired. It is split into two categories, Atom and Molecule, and each are sorted separately. All of the relevant Triums were inserted into a SortedDictionary, which sorted by Atomic Number (Atoms) and Name (Molecules), respectively.

# What did not go well?

### **General**

* Group understanding & acceptance of assigned tasks
  + During our planning phase for Sprint 1, our tasks were assigned on a “first come first served” basis. This left some members with tasks they did not understand or tasks that their skillset did not necessarily align with.
* Underestimated required time to complete tasks
  + Throughout Sprint 1, we found that the estimated time to complete some of our tasks was drastically underestimated. This is due in part of not understanding how to solve the problem and making the solution compatible with the Unity Engine, as well as bringing some of our components together with others. These underestimated tasks put several extra hours of strain on our overall schedule and took away from time needed to debug and organize our solutions.
* We did not allocate specific amounts of time to test and debug
  + Our time allotment for Sprint 1 tasks implicitly included testing and debugging along with general script development. This also led to spending more than the allotted time for tasks as well as running out of time at the end to test, debug and combine our work.
* Underestimated time required to combine our work
  + As with testing and debugging, the time we needed to combine our scripts and assets was not properly allotted in our planning document. This led to a majority of our user stories not being fully completed and key features not being present for our sprint review, even though most of our individual tasks were completed.
* Team members fell behind on work or started late
  + Due to complications from other courses, exams, interviews, and overall workload, some of the tasks in our group were started late or were not completed. This led to a very stressful rush to completion in the final hours before our Sprint Review, which is something that we did not anticipate nor enjoy.
* No organized group meetings until the final days of the Sprint
  + Our group had no organized group meetings until the final days of the Sprint, leading to a group-wide panic and rush to completion effort before our Sprint Review. In addition, the attempted meetings and communication about the project were not orchestrated by the scrum master, but by other members of the group, which added to the overall disconnect we experienced during the Sprint.

### **Tasks**

The following are User Stories that were neither successful nor fully implemented, as well as the tasks that were not completed for the aforementioned story. *Tasks successful to each user story are listed in the above section titled “What went well?”*

* User Story # 2: As a user, I would like to be able to perform Fusion on two atoms to create a new element.
  + Task 5: Create Algorithm to select existing Triums
    - An algorithm was supposed to be written to highlight a Buddy inside the user’s workspace, but failed to be delivered in the final build. Lack of time for debugging led to this algorithm to not be completed for the Sprint Review.
  + Task 9: Create the author program to allow for developers to add new data to the database
    - An external program was to be created to interact with a SQLite database file, which would populate the database’s tables with segments of information about Triums, Reactions, Elements among others. The program was not entirely finished when the database needed to be populated, and an alternative program had to be used.
* User Story # 3: As a user, I would like to be able to perform Group on two or more atoms to create a single-element molecule.
  + Task 2: Create Algorithm to perform Group
    - A C# script was to be created to group two Triums selected within the workspace of the application. It would follow similar to the Fusion script, obtaining database information about the resulting grouped molecule, creating a new Buddy object within the application workspace and updating our dyanmic data structure. Its completion was barred due to lack of time spent elsewhere for debugging and collaboration purposes.
* User Story # 4: As a user, I would like to be able to perform a Reaction on two or more Triums to create a compound.
  + Task 2: Create Algorithm to perform Reaction
    - We were unable to write the algorithm due to the lack of time and the required knowledge of interaction with the static SQLite database.
* User Story #5: As a user, I would like to be able to collect atoms representing the naturally occurring elements.
  + Task 2: Create UI assets for element selector
    - The UI we desired to create would use an algorithm that allows a user desired selection of Triums to be randomly generated inside the workspace. Debugging and collaboration issues prevented the UI’s development progress.

Although we were able to address many of the separate components, we were not able to link them together to create a functional build in the time constraint. The following is a summary of what we lacked in Sprint 1:

* Time management
* Combining the individual parts to each other, especially backend to front end
* Team coordination and communication, not meeting weekly like stated

# How should we improve for Sprint 2?

* Require more consistent and frequent group meetings
  + In Sprint 1, the group hardly followed the meeting plan and arranged meetings. This led to miscommunication and miscoordination that slowed down the project’s progress. In the following sprint, we should enforce the meetings to keep track of each member’s progress and better collaborate with each other for the separate work of each member to connect more smoothly.
* Actively communicate as a group on a consistent basis
  + In Sprint 1, there was not a lot of communication in the team, which left members with lack of detail that they might have needed to carry out the required work. In the following sprint, communication should be more frequent to keep everyone on track and on the same page, should there be misunderstanding regarding the project.
* Accurately estimate the time required for each task
  + Since many of us were not familiar with the game engine we were utilizing, and additionally accounting for the sequence of events and outside obligations that the members had, such as midterms and interviews, we did not manage time well in this sprint and found ourselves overtasked. Now that we have had hands-on experience, we should plan and allocate time more accurately to prevent such from happening again, ensuring that we have the right amount of time to complete the right amount of tasks.
* Ensure full completion of User Stories
  + Although many of the tasks were implemented for each user story, we ran out of time to link the components to create the functional build required for sprint 1. In this sprint, we should plan according and keep track of everyone’s progress to ensure that tasks are completed as well as leaving required time to gather and collaborate on merging each member’s parts to create the final product.