**ChemXplosion Report**

Team ChemJam

Software Developement II

April 28, 2018



**Overview**

Thomas Gluick, the client of our product, Assistant Professor of Chemistry at GGC, is very passionate in his profession, which is why he would love to see his educational game succeed. He dabbles in gaming and programming and enjoys the humor found in Rick and Morty.

ChemXplosion is a chemistry game which focuses on chemical reactions. The game was made in collaboration with our client. The purpose of the game is to provide an entertaining and educational approach to chemistry. Our goal was to fix what the previous team had completed and improve and expand on the foundation which has already been placed.

**Team Members:**

Hieu Dinh - Data Modeler Lead / Code Architecture Lead  
Rae Vroman - Team Manager/ Client Liaison   
Alex Croghan - Testing Lead  
Humberto Michael Lopez - Documentation Lead / Sound Engineer Lead  
Evan Lamkie - UI/UX Design Lead



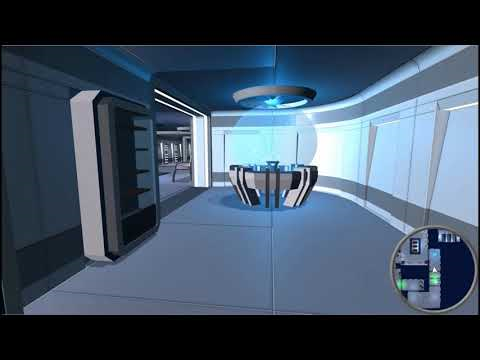
**Project Overview Screencast**

Below is a link to a video briefly describing ChemXplosion, testing, and team performance.

<https://www.youtube.com/watch?v=ok9eoLujoB8&feature=youtu.be>

**New Features**

* Smoother Walking/Running Animations
* Sound FX To Doors
* Background Music (BGM)
* Chemical Models
* Chemical Thumbnails
* Tooltip System
* New Chemistry Assets
* Chemical Pouring Animations
* Working Puzzles
* New Puzzles
* Correct/Incorrect Puzzle Animations
* Inventory System
* Ability To Complete Level
* Minimap

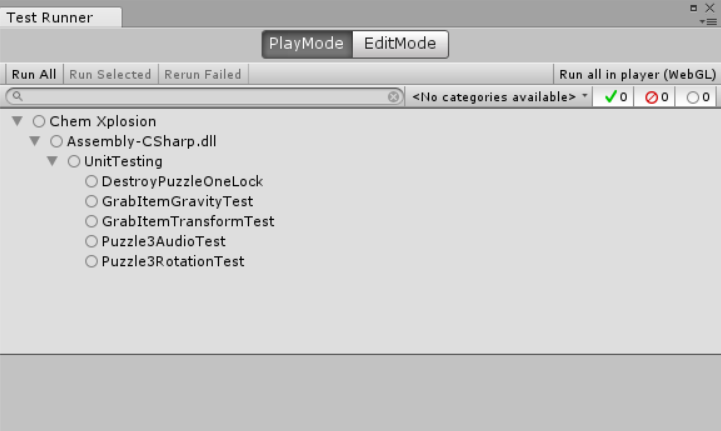
**In-Game Example**

**Known Issues**

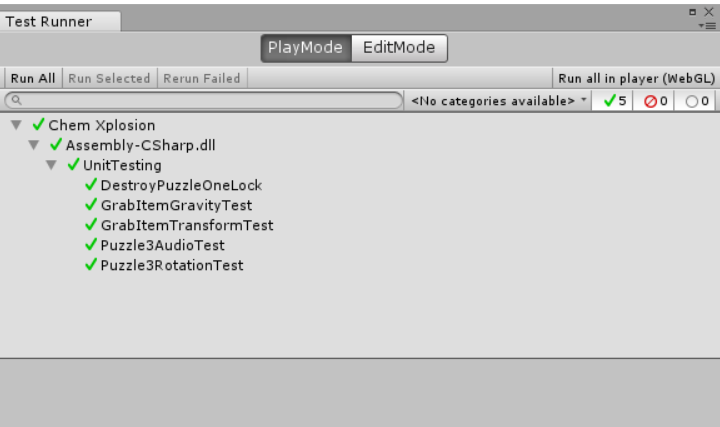
* While holding chemical element, look straight down; Will fly up (also way to access hidden easter egg)
* Few locations with possible fall through floor/ceiling/wall
* Generator - Final puzzle - Top fixture - does not rotate smoothly
* Restart option will cause player to be unable to move
* Clicking on quit game causes cursor to disappear
* Few GUI related issues

**Testing Coverage, Methods, and Results**

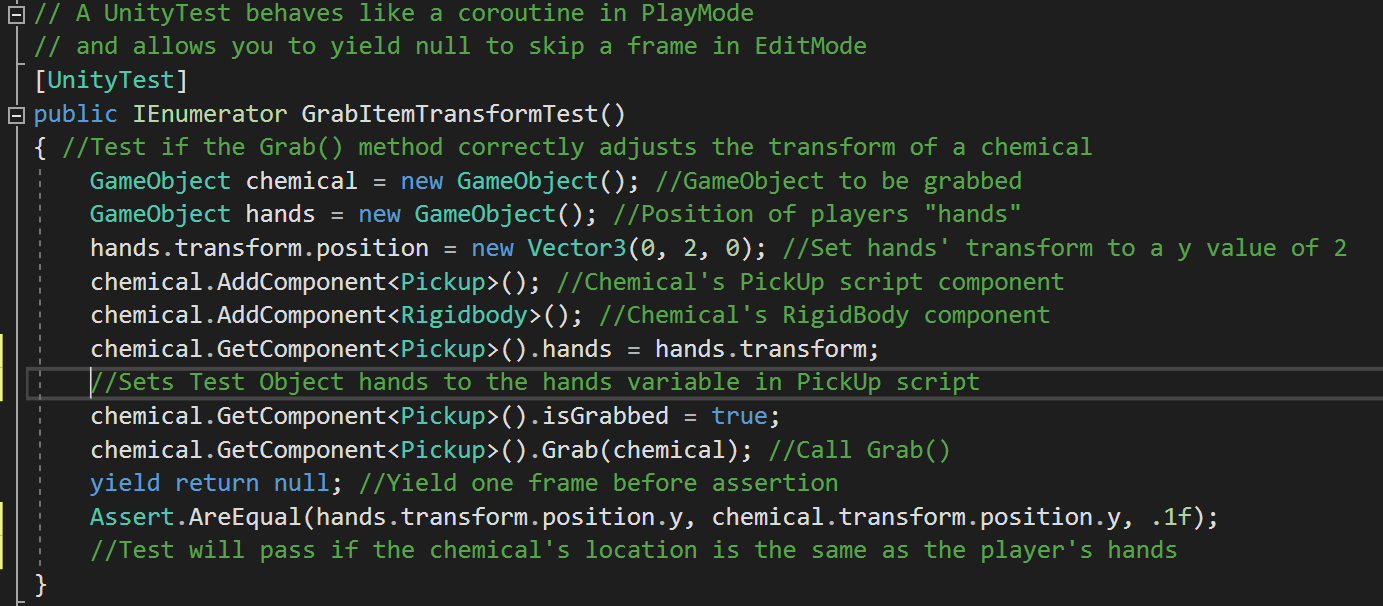
Automated Testing was performed via the Test Runner within Unity. Unity’s test runner uses assets known as PlayMode tests which are written in C# and behave like a coroutine when the game is run, allowing frames to be skipped before running tests. Testing coverage and methods included the grabbing of items (testing both an item’s gravity and position), puzzle animations and puzzle sound effects.

**Test Runner**  


**Test Results**

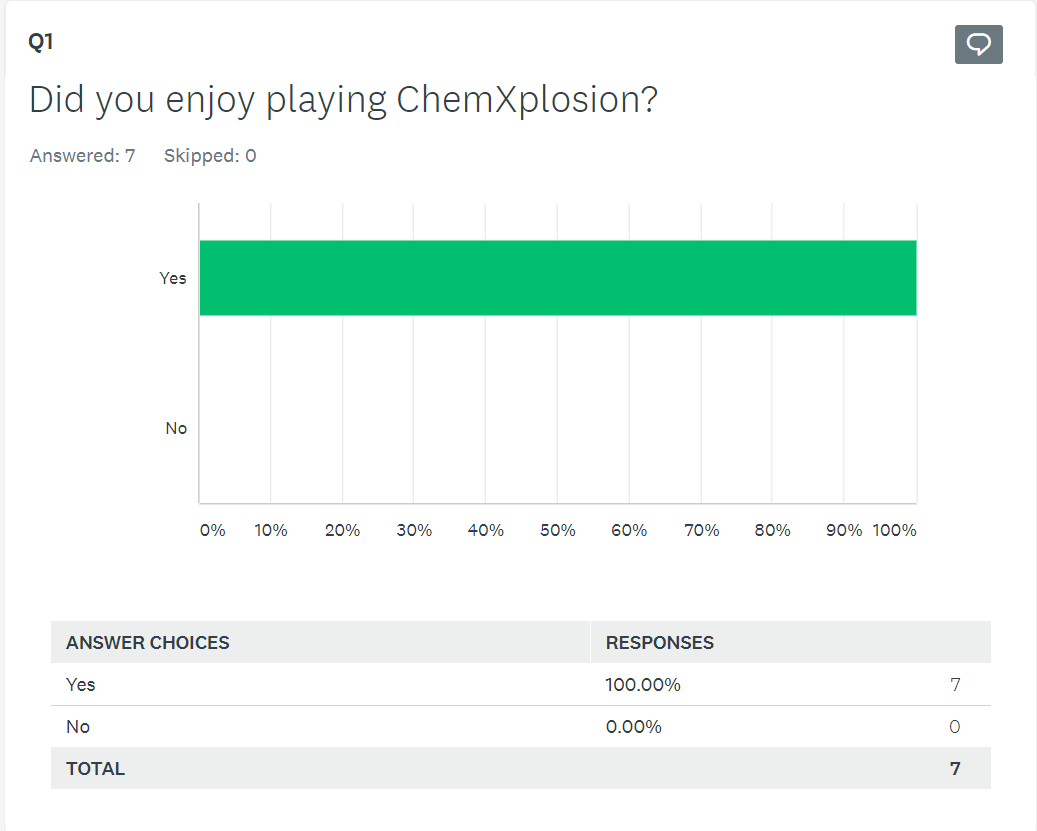


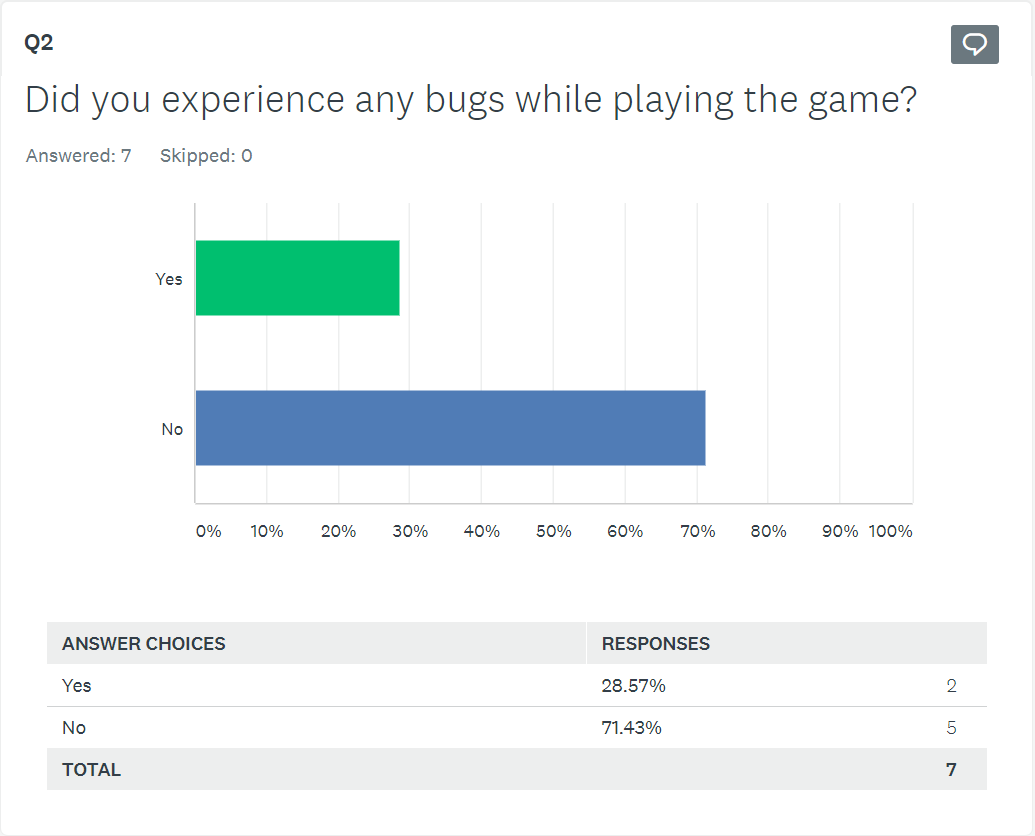
**Example PlayMode Test**

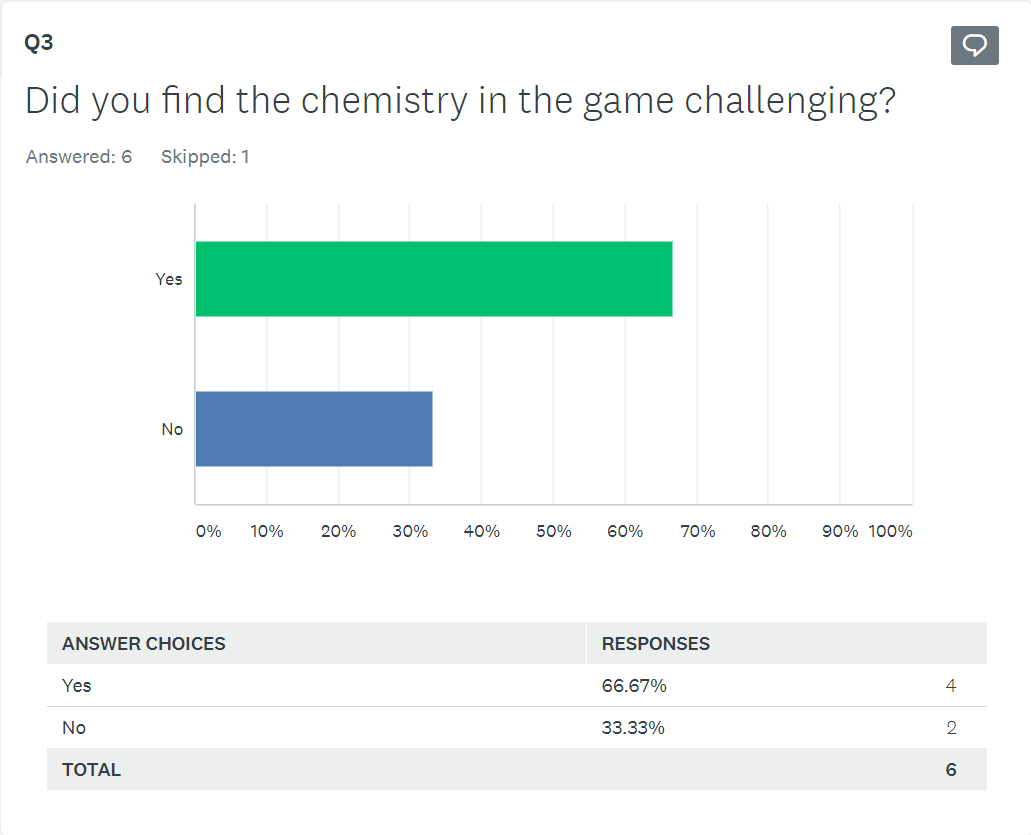


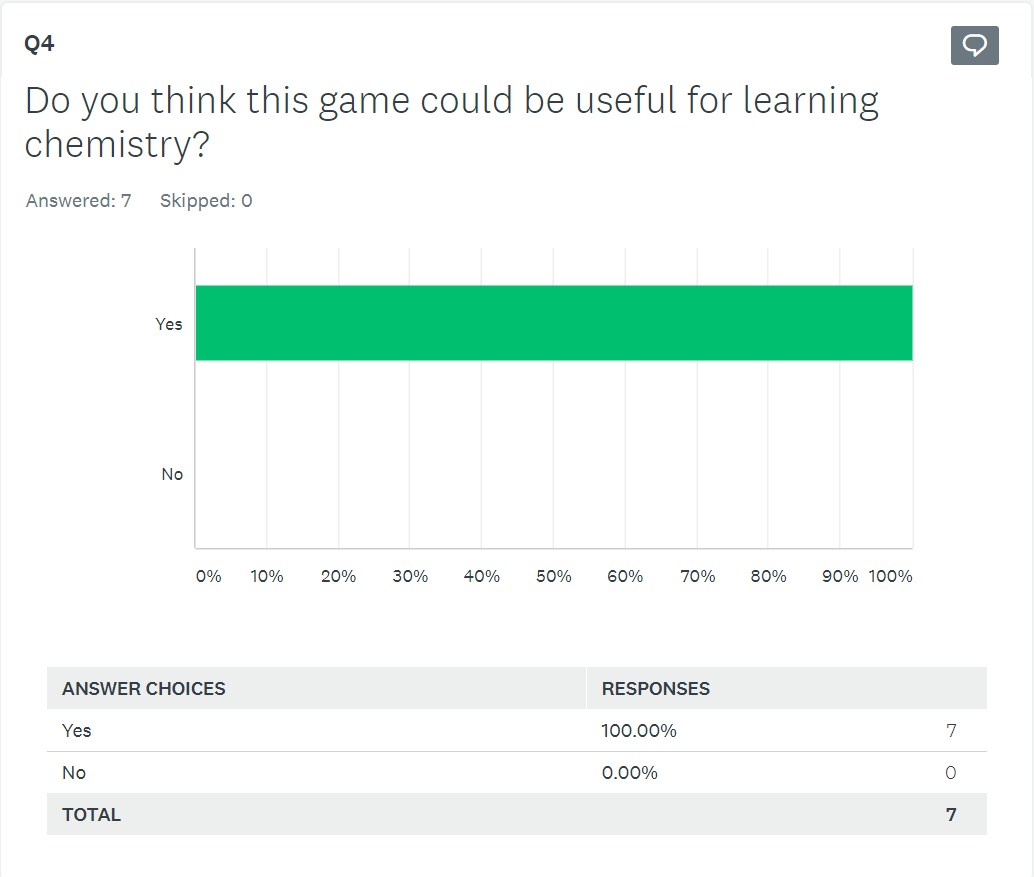
**Usability Testing and Survey Results**

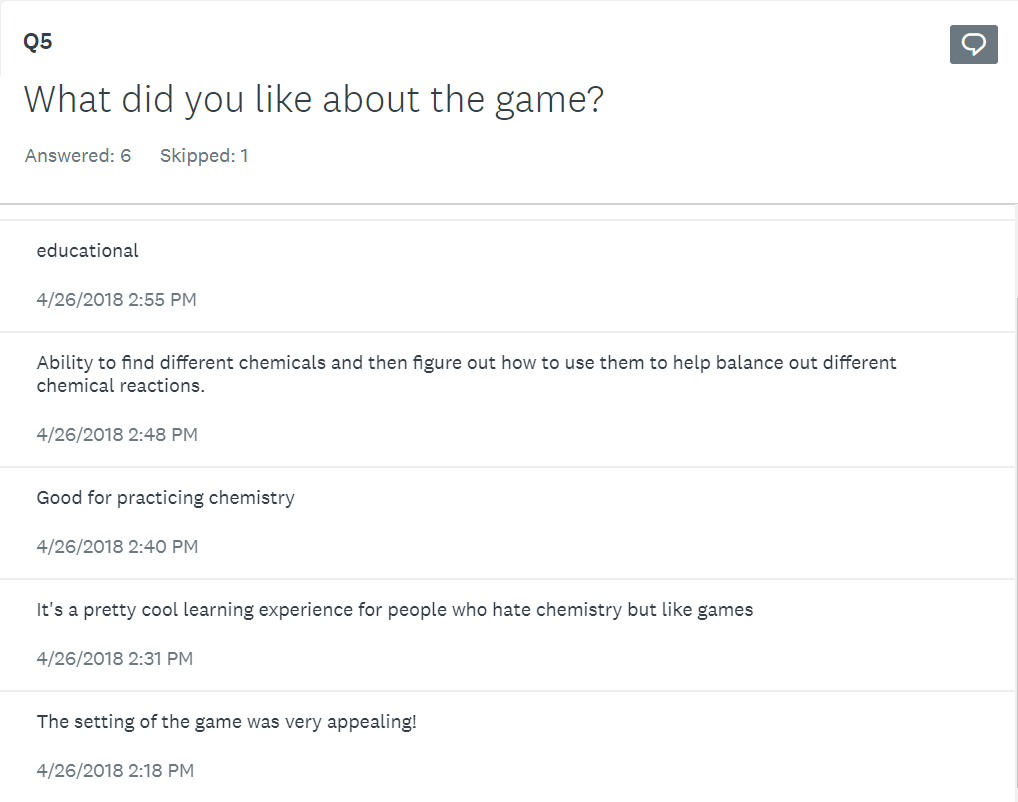
At the CREATE Symposium, we had an opportunity to have students and teachers test our game. All testers were asked to complete a survey so that we could collect feedback about the current state of the game, as well as improvements that could be made in the future.

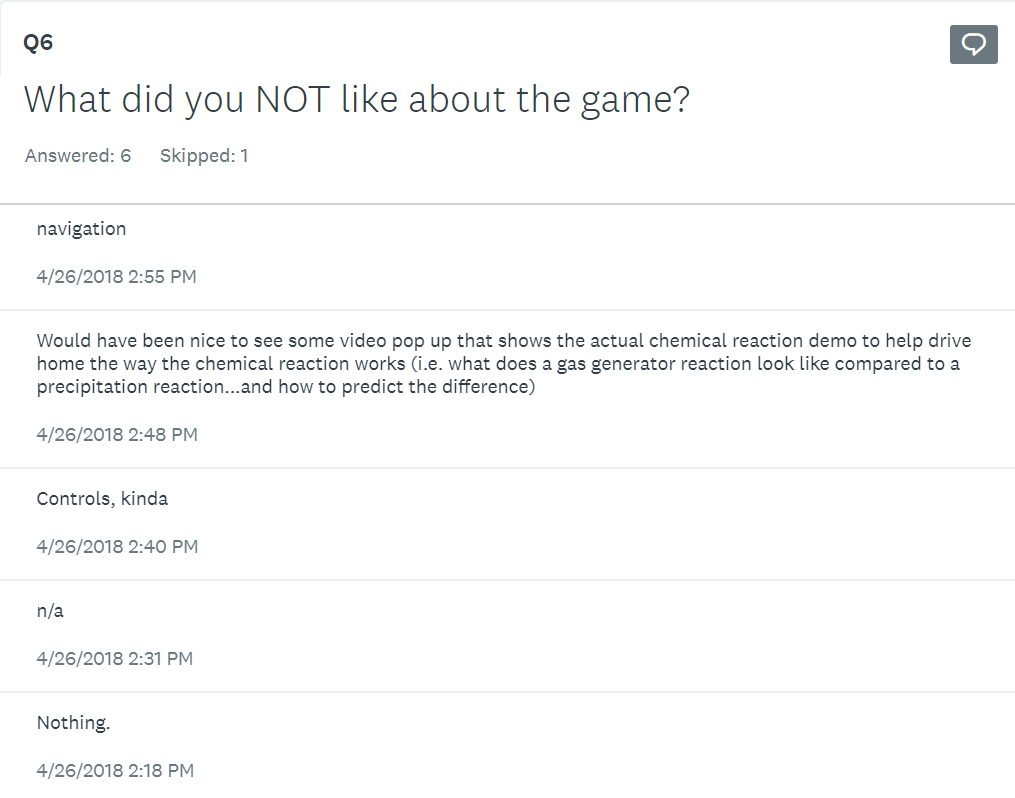
****

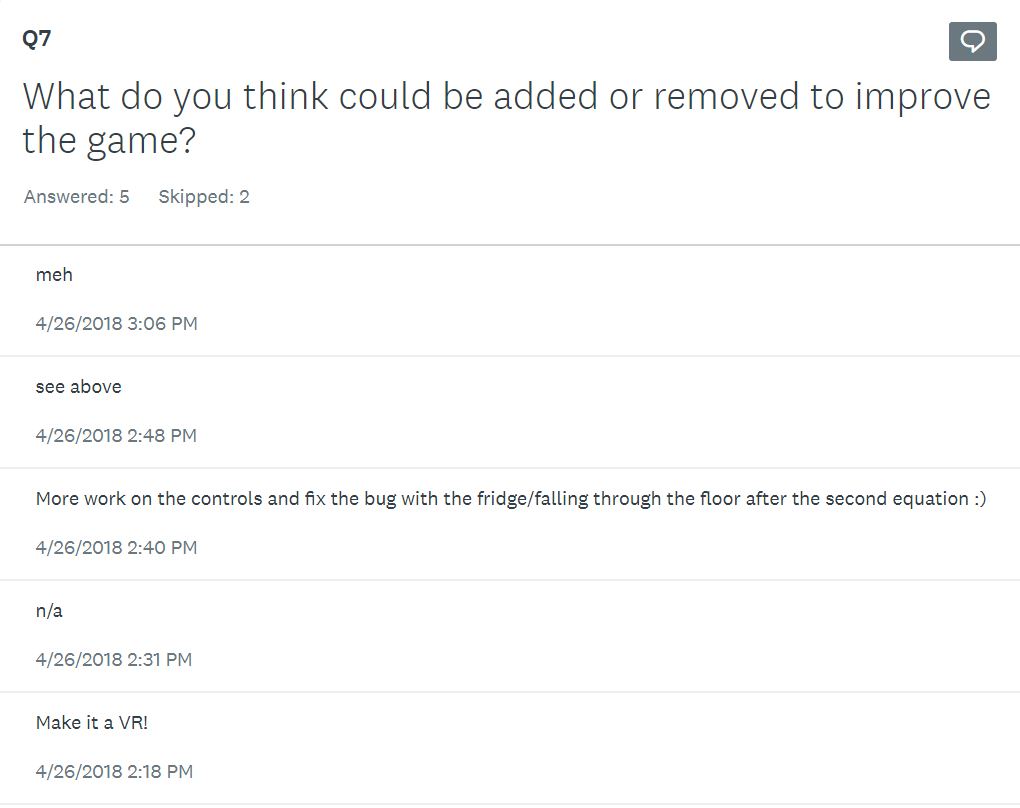
****

****

****

****

****

****

The results of this survey suggest that the ChemXplosion is moving in the right direction. We hope that future developers will be able to use this feedback to improve what we were able to create.

**Location of Installation, Developer, and User Documentation**

**Code Documentation**

\*\ChemXplosion\Documentation\Spring 2018\Code Documentation

**Developer Documentation**

\*\ChemXplosion\Documentation\Spring 2018

**User Documentation**

\*\ChemXplosion\Documentation\Spring 2018

**Performance Overview**

**Iteration Plans and Rational**

For our first iteration we focused on finishing the first level. When we got the game, it had one complete puzzle in it, so we added three more puzzles. We also filled up the rest of the level with assets (3D models) after the second puzzle. We fixed a lot of the bugs that were in the game like being able to walk through walls and door, and made the player controller move more smoothly.

The second iteration we improved the puzzles and the first level. We made the puzzles more visually appealing by adding a 3D model of a chemical for the user to use in the puzzle, adding 3D models of chemistry items like flasks, bunsen burners, etc, added screens on the puzzles to show when the puzzle was completed properly and what type of reaction it was, and added sound effects and animations when solving the puzzles. We also created an inventory system, created a minimap, added more sound effects, and added visual cues for interacting with the environment.

Future iterations might include adding saving/loading the game, traps or other game obstacles, converting to VR, making more difficult chemistry problems, or generally increasing the complexity of the game.

**Iteration Overview**

In the first iteration, we completed 4 user stories and 28 story points. Most of the time was spent learning Unity, because there is such a high learning curve. According to our agreement, each story point was worth 2 working hours so we should have completed 56 hours of work. We ended up having a much higher amount of working hours in this sprint because the story points were underestimated. This could probably be attributed to the fact that Unity is a very difficult and complex program to learn.

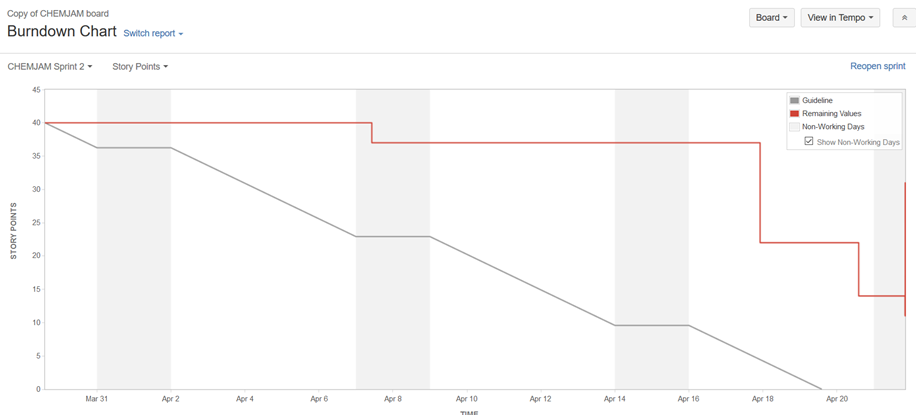
In the second iteration we completed 10 user stories and 49 story points. We were able to complete a lot more because we had more knowledge and experience with Unity. The story points were underestimated because the hours put into each user story still exceeded the estimated amount of hours based on the story points.

**Velocity**



In each iteration, the team completed more story points than were originally estimated. This tells us that even though we underestimated how long each user story would take, the group still was working at a faster pace than was expected. If we were doing more iterations, I would definitely increase the number of story points done in an iteration, and I would also increase the story point estimation for each user story.

**Burndown**



The burndown chart definitely doesn’t look like it’s supposed to. At the end, more user stories were added (that had already been completed but that weren’t included in the sprint), so there is a big spike. The chart also does not decline at the ideal rate, which makes it seem like the group finished a lot of tasks at the last minute instead of steadily throughout the sprint. This is actually because the group wasn’t recording their work in Jira as they completed it, instead we tended to record our work after we had completed a lot of tasks which made the chart dip down a significant amount in a short period of time.