3D Molecular Visualization Game

Project Plan

**The RIsoTopes**

Justin Cotner, Andrew Popovich, Forrest Shooster, Christopher Timmons

## Table of Contents

[Table of Contents](#h.lfyi42kzqcja)

[Revision History](#h.68hvkk7lvt3t)

[Project Overview](#h.ugmg45mjsthm)

[Goals and Scope](#h.6cxayt4dh6s1)

[Goals](#h.igxpiajw8sbz)

[In Scope](#h.jz04x2esqxtk)

[Out of Scope](#h.k28k3bmgw3zl)

[Technical Process](#h.p68kbpoyxjvs)

[Scheduling and Estimates](#h.dnih7lqkprg)

[Risk Management](#h.vnvrrqy657gg)

[Measurements and Metrics](#h.957tyiy4770f)

[Deliverables](#h.se2b4hapfyv5)

[Fall Semester Deliverables](#h.17ta6bqcm7hx)

[Spring Semester Deliverables](#h.e6ow93ynn4qc)

## Revision History

## 

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date of Revision** | **Reason for Revision** |
| Version 1.0 | 9/15/15 | Initial Draft |
| Version 2.0 | 1/27/16 | Added Spring Semester Schedule |

## Project Overview

This project involves creating a 3d molecule visualization game for chemistry students to link their in-class studies with a virtual representation. This is meant to help the students understand how the molecules are supposed to look and how they interact with other molecules and forces. The concepts taught in the students’ classes will be reinforced with this game.

This project continues the work of the last year’s senior project team and aims to expand upon the levels and game content of which that team created. In addition, this project aims to expand upon the basic server structure in place so that chemistry instructors can gain a better understanding as to how students are progressing in the game and how that progress correlates to the students’ understanding of specific chemistry concepts. The top two priorities in this project are playability and realistic modeling. These priorities may conflict with each other at times since the computational effort to display realistic molecules and their interactions may detract from the overall playability of the game. Therefore, a balance between playability and realism is desired for this project.

## Goals and Scope

### Goals

* Establish a baseline for the game from which future project teams can build off of in an easier manner

### In Scope

The Game

* Establish and implement a core gameplay mechanic for the game
* Establish and finalize a storyboard for the game
* Improve existing levels and content
* Add new content in terms of levels and zones

We plan to add more contiguity and cohesion to the gameplay that currently exists for the game by adding in mechanics that help to make the gameplay enhance the learning experience provided rather than tangentially associate itself with learning. To this end we will create a storyboard of new scenes, new UI, and new level content to be added. Likely, we will end up changing the way the levels are currently done to fit the new ideas being developed so that they may feel more relevant.

Some ideas that suggested for new content were:

* Pokedex style molecular analyzer device (Molecudex?); effectively a mobile database the player carries around in-game that they can use to identify molecules by observing their main features, such as molecular geometry and bond angles
* Turning the core of the game into a Role-playing-esque collection game based in crafting using specific chemicals / molecules
* Other Collectors existing in the world
* Acidic/Basic Molecules that may cause things to happen to the ship

Server-side

* Improve student/teacher accounts
  + Add more metrics to student accounts
  + Enable teachers to set up a class of students
  + Enable teachers to view students’ progress

Platforms

* Windows, Mac, Linux
* Ensure that RIT computers can download and run the game

### Out of Scope

* Game ported to mobile platforms
* Multiplayer content
* Level customization by teachers

## Technical Process

The team has decided to use the Scrum methodology for this project. The only real requirements we have are that the game has a high level of “playability”, that the player interacts with single molecules only, and that the molecules behave in a realistic fashion. Other than those three requirements the sponsor has given us free reign to make the game however we want. Because playability is highly subjective and given the few requirements it is important to get a product in front of the sponsor as often as possible to receive feedback.

Tools we will be using include Trello to keep track of user stories, Slack for communication, and Git for our version control system and issue tracker.

To facilitate the Scrum process we will hold bi-weekly sprint planning meetings, bi-weekly retrospectives, and daily stand-ups via Slack if in person is not viable.

There will be several internal project artifacts that will be maintained. Items that will be updated frequently are the four-up charts, sprint backlog, product backlog, and issue tracker (bugs/defects). Items that will be updated less frequently include the project plan, domain model, and the risk management document.

## Scheduling and Estimates

The team will be utilizing two week Scrum sprints. What exactly will get done in each sprint will be decided on at the sprint planning meetings. The two week sprints will begin in week six and continue until the end of the second semester (taking a break during intersession). In the first five weeks we will complete the Synopsis, Project Plan, a Domain Model, and the initial product backlog. When planning sprints we will use story points as an estimate of how much time a task will take (compared to other tasks). Each team member will be assigned some number of story points to work on during each sprint. Story point values will be determined during the sprint planning meeting. While meeting individual sprint deadlines is important it is not as important as meeting the overall deadline of the end of second semester. Therefore if a team member is unable to complete a user story in a given sprint they will keep working on it in the next one.

## Risk Management

This project is not without its share of risks. Most of the risks come from two areas. The first of these areas is with the requirements elicitation. Specifically, the project sponsor has a goal in mind for the game, but he does not have specific requirements that enable us to to reach that goal. Our mitigation strategies for this case are to meet with the sponsor to elicit specific enough so that there is more of a framework to work with. The other area is having to follow a previous team’s work. Depending on the requirements that we elicit and how the other team followed them, we may need to modify or redo their work. Our mitigation strategies for this are to find what worked for the other team and focus on that instead of what didn’t work.

We are using a basic risk management system with the risk exposure based on the probability of a risk happening and how much it would impact the project if it occurs. We will be monitoring our risk management document for any changes on how our prevention and mitigation strategies are working to limit the exposure of the risks and change them if need be.

## Measurements and Metrics

### Velocity

**Description:**

Due to the team’s use of Scrum the team will be able to use project velocity as a metric to measure the team’s progress. Project velocity is a process metric which measures how many units of work are completed in a set time interval.

**Calculation:**

Velocity in our case will be calculated as story points per sprint, in which the story point measure will be determined for each user story as a part of the product backlog creation process.

**Application:**

With this metric the team will be able to gain an understanding of how progress is being made on over a wide period of time. Therefore, this will help the team determine which sprints went well and which sprints did not get so well. As a result this will help the team determine the reasons behind the success or lack of success with each sprint.

**Drawbacks:**

Unfortunately however, the team will only have a limited number of sprints per semester, so this metric may not prove to be useful until a few sprints have passed.

### Defect Density

**Description:**

Another metric that the team will track is the defect density in the project. Defect density is a product metric which measures how many defects are in a particular portion of code.

**Calculation:**

The defect density will be tracked by measuring the number of defects found per user story.

**Application:**

This metric will help the team by providing a measure which shows how many defects are present in each user story. With this information we will be able to categorize user stories by how many defects were found in each user story, and then use those classifications to help provide better story point estimates when working on related user stories.

**Drawbacks:**

The defect density metric is prone to bias due to how defects are defined and how granular the user stories are. Therefore, we will not be able to use this metric as a means of estimating how many defects will be found per user story or how many defects will be found per story point.

### Requirements Volatility

**Description:**

Another metric that the team will generate is requirements volatility. Since the team has chosen Scrum as its process methodology due to the lack of specific and concrete requirements, it is expected that specific requirements will change as the project progresses.

**Calculation:**

Requirements volatility will be measured as the percentage of user stories that change per sprint. This will include user stories that were completed but not matching what our project sponsor wants in terms of what the game or server should be doing.

**Application:**

Requirements volatility could be a useful metric that would help both the team and the project sponsor determine if the requirements are too volatile to make development progress, and if so we could then address ways to correct requirements volatility and continue to track it in the future.

**Drawbacks:**

Requirements volatility may not be a useful metric if no user stories are modified during our development process.

## Deliverables

The team’s use of Scrum requires that the team creates and maintains some internal artifacts. These artifacts are:

* Product Backlog - A collection of features that we plan on implementing throughout the project and they are represented as user stories with story points. This will serve as our requirements document from which we will use to discuss new features with our project sponsor.
* Sprint Backlog - A collection of user stories that we will be working on throughout a single sprint. These user stories will be considered to have the highest priority and value to our project sponsor, and therefore these documents will be discussed with him before sprints.
* Project Burndown Chart - This burndown chart will track the amount of story points completed in each sprint.
* Sprint Burndown Charts - These burndown charts will track the amount of story points completed in each day of our sprint. Story points are not completed until a their corresponding user story is complete.
* Sponsor Progress Deliverables - We plan on delivering working code to our sponsor at the end of each sprint. Also, while doing so we plan on deploying our game and server through some team independent mechanism for the sponsor to keep.

At the end of each sprint the team plans on delivering progress updates in the form of code and documentation. These progress updates will consist of:

* Sprint Progress Deliverable - The latest code version that is known to be working
* Sprint Burndown Chart - A completed burndown chart for the completed sprint
* Product Backlog updated - The product backlog will be reviewed and updated accordingly
* Project Burndown Chart updated - The project burndown chart will be updated
* Risk and Metrics Trackers updated - Updates to the Metrics tracker and Risk tracker
* Website Updated - Published documentation updates on the website

Also, the following documents will be created at the beginning of each sprint:

* Upcoming Sprint Backlog created - A new backlog for the next sprint
* Upcoming Sprint Burndown Chart created - A new burndown chart for the next sprint

Finally, we plan on completing a one page Four Up Chart weekly, which will document each team member’s completed work, planned work, risks, and needs.

### Fall Semester Deliverables

The following is a schedule of all of the deliverables for the fall semester. These deliverables include the deliverables required for the Software Engineering department, as well as our Scrum deliverables discussed above.

|  |  |
| --- | --- |
| **Week Number - Date of Week Start** | **Deliverables** |
| Week 1 - 8/24/15 |  |
| Week 2 - 8/31/15 | * Project Website created * Time/Effort Tracking created |
| Week 3 - 9/7/15 | * Project Synopsis |
| Week 4 - 9/14/15 | * Project Plan (this document) created * Domain Model created * Metrics and Risk Trackers created * Website updated |
| Week 5 - 9/21/15 | * Product Backlog created * Project Burndown Chart created * Website updated |
| Week 6 - 9/28/15 | * Sprint #1 Documents created * Website updated |
| Week 7 - 10/5/15 | * Mid-term Peer Evaluation * Sprint #1 Burndown Chart updated |
| Week 8 - 10/12/15 | * Sprint #1 Deliverables and Updates * Sprint #2 Documents created |
| Week 9 - 10/19/15 | * Sprint #2 Burndown Chart updated |
| Week 10 - 10/26/15 | * Sprint #2 Deliverables and Updates * Sprint #3 Documents created |
| Week 11 - 11/2/15 | * Sprint #3 Burndown Chart updated |
| Week 12 - 11/9/15 | * Sprint #3 Deliverables and Updates * Sprint #4 Documents created |
| Week 13 - 11/16/15 | * Sprint #4 Deliverables and Updates * Draft Interim Presentation |
| Week 14 - 11/23/15 | * Sprint #4 Deliverables and Updates |
| Week 15 - 11/30/15 | * Clean Up Project Artifacts * Prepare for Presentation * Website updated |
| Week 16 - 12/7/15 | * Present the Interim Presentation * Interim Team Self-Assessment |
| Finals Week - 12/14/15 |  |

### Spring Semester Deliverables

|  |  |
| --- | --- |
| **Week Number - Date of Week Start** | **Deliverables** |
| Week 1 - 1/25/16 | * Deliverable Commitment Document * Spring Semester Schedule |
| Week 2 - 2/1/16 | * Start of Sprint #5 * Website Updated |
| Week 3 - 2/8/16 | * Sprint #5 Deliverables and Updates |
| Week 4 - 2/15/16 | * Start of Sprint #6 * Website Updated |
| Week 5 - 2/22/16 | * Sprint #6 Deliverables and Updates |
| Week 6 - 2/29/16 | * Start of Sprint #7 * Website Updated |
| Week 7 - 3/7/16 | * Sprint #7 Deliverables and Updates |
| Week 8 - 3/14/15 | * Start of Sprint #8 * Website Updated |
| Spring Break |  |
| Week 9 - 3/28/15 | * Sprint #8 Deliverables and Updates * Coordinate Final Presentation with Sponsor |
| Week 10 - 4/4/15 | * Start of Sprint #9 * Schedule Final Presentation |
| Week 11 - 4/11/15 | * Initial Project Poster Concept Due |
| Week 12 - 4/18/15 | * End of Sprint #9 * Deliver Project Poster To Project Coordinator |
| Week 13 - 4/25/15 | * Poster Presentation Day * Draft Final Presentation * Technical Report Outline |
| Week 14 - 5/2/15 | * Give/Attend Final Presentations |
| Week 15 - 5/9/15 | * Give/Attend Final Presentations |
| Finals - 5/16/15 | * Final Reflection Meeting |