Chemistry Learning In Progress

Project Plan Document

Nathan Mikeska
Neil Alfredson
Richard Carney
Brian Navarro

Table of Contents

1. Introduction

1.1 Overview

CLIP (Chemistry Learning In Progress) is a new digital system that will be used to help chemistry students to get a better understanding of the organization and patterns in the Periodic Table of Elements. This system is also going to contain functionality to allow a teacher to view the students organizational thought process.

The client is Prof. Susan Wiediger of the Chemistry Department. They have a current system on paper that involves the students moving around paper tiles. Our client wants to make the current system electronic and available to students over the web. She also wants the students to be able to submit their results to her, possibly to be used as an assignment.

The purpose of this system is to allow students to demonstrate their understanding of the organization of the periodic table of elements. This system will show that they understand and find the different patterns in the design of the periodic table of elements. This system will also be designed so the teacher will be able to see the end result of the student's arrangement and view the processes that they used to reach this result. The teacher should be able to use the system to observe the line of thinking that the students used.

1.2 References

C.L.I.P. Project page
http://www.cs.siue.edu/SeniorProjects/f05g6/

1.3 Overview of the Remainder of the Document

The remainder of the Project Plan will discuss the team members, their roles, the client, and the end users. It will take in consideration, the impact that client, end users, and the developers might impose in our project. Project organization and development process are also discussed in detail in this document. This document also covers the development process, which consists of the process model, software documentation and reviews, development methodologies, policies, standards and tools, followed by configuration management.

The project plan will include hardware and software requirement specification that would be necessary for the development of software

covered by Rich. Deliverable deadlines are proposed in order to meet the goal of our project (CLIP) also cover mainly by Rich. The last sections of the documents discuss efforts and schedule based on team input and written by Brian, measurement of the project written by Nate, risk management covered mostly by Brian, and ethical considerations that Neil typed up based on discussions of the group, which will be put into effect to avoid any problems that might arise during development process. The group as a whole was involved in most of the entire document and were involved in coming up with the current understanding of each of the sections.

2. Stakeholders

Client –

The client for the project is Susan Wiediger a chemistry professor at Southern Illinois University of Edwardsville. Along with defining the requirements she will also be a primary user of the system.

Users –

The primary users of the clip system will be chemistry students and chemistry professors. The Students will perform card sorting and arranging activities, save their results, and submit their results via email to the professors. The professors will then observe the students moves and analyze their thought process. The students will gain knowledge of the periodic table of elements and the professors will gain knowledge about the thought process of their students. However the system will not be restricted to only these two groups, anyone can use the system.

Development Team –

The development team consists of four members; Nathan Mikeska, Neil Alfredson, Richard Carney, and Brian Navarro, who will receive grades in CS 425 Computer Science Senior Project Course depending upon the success of the project.

3. Project Organization

There are four team members working on the CLIP system. In order to maximize the effectiveness and organization of our work, each member will be taking certain roles in the project, although all members will still assist in most portions of the project.

<u>Project Manager</u>

Nathan is the Project Manager. As Project Manager, he will oversee all aspects of the project and keep it focused and on track. He will assist the other members in their assigned positions as well as lead any portion of the project not assigned to any other member.

Lead UI Designer & Lead Programmer

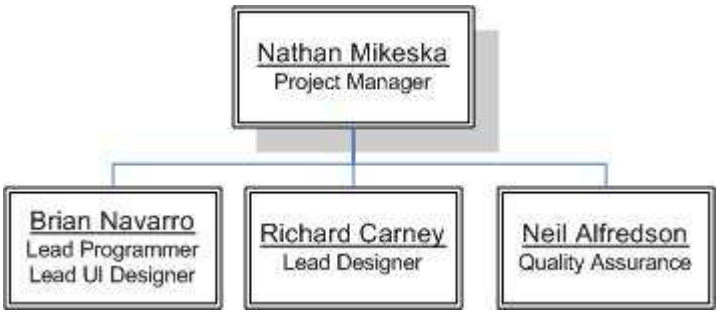
Brian Navarro will fulfill the role of Lead User Interface Designer as well as Lead Programmer. Brian is very talented in as well and interested in interface design and fits the Lead User Interface Designer role well. He is also the only member of the team that has had previous experience with programming in Java. Since he is the only one currently skilled in the programming language, he will also take the role of Lead Programmer. Since Brian is Lead Programmer, he will also take the role of Risk and Development Officer as described in section 9.

Lead Designer

Richard will be handling the role of Lead Designer. He will focus on the design of the system and make sure that it follows what was outlined in the System Design Document. He will also take the lead in any addition or removal of features that require changes to the system design.

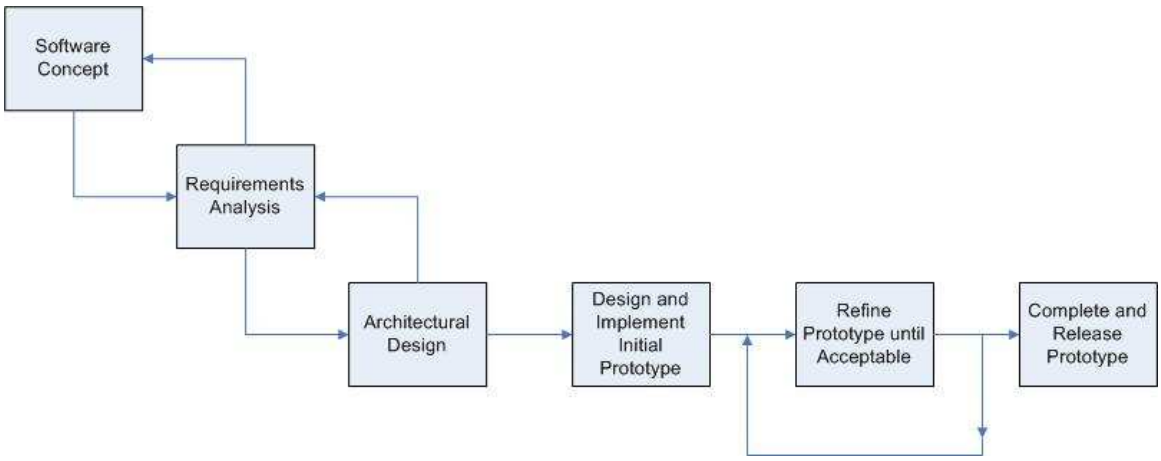
Lead Quality Assurance

Neil will take the role of Quality Assurance. He will be responsible for taking the lead in testing and bug finding.



4. Development Process

4.1 Process Model



The lifecycle model that we will use for the entirety of project will be the waterfall model. However, within our coding and testing phases, we will gear our model to a more evolutionary prototyping approach.

Given the small team size and project milestone deadlines, the waterfall model provides a solid lifecycle that we can follow without needing to make compromises to our goals or deadlines. During the coding and testing phases of the lifecycle, our project will take an evolutionary prototyping approach. It is very important that the system be easily understood and used by the users. This will require a strong focus on providing a good user interface for users to interact with the system.

Therefore, the evolutionary prototyping will allow us to gather important feedback on each iteration of the prototype so that we may provide the best possible interface for the users.

4.2 Software Documentation and Review

We have several important milestones for the project as outlined below:

Requirements Analysis Document	09/26/2005
Requirements Analysis Presentation	10/05/2005
System Design Document	10/10/2005
System Design Presentation	10/12/2005
Contract First Draft	10/24/2005
Project Plan Document	10/26/2005
Project Plan Presentation	11/02/2005
Final Contract	11/09/2005
Prototype Plan	11/14/2005
Final Report	12/05/2005

4.3 Development Methodologies, Policies, Standards, Tools

4.3.1 Methodologies

The system will be developed in an object oriented, modular design as laid out in our Design Document. By keeping the system object oriented and modular, it will help in translating the current system to the new system as well as easily allow team members to work on different aspects of the system in parallel and seamlessly integrate them with on another.

4.3.2 Policies

Most of the code written for the system will be made by team members working on their own or in small groups within the group. Every member is expected to thoroughly comment and test their code on their own. After the code writer believes the code is ready, then it will be reviewed and tested by the team at a team meeting so that everyone understands the code and finds no problems or performance issues with it.

4.3.3 Standards

The coding standards and conventions that we will use for the project can be found at http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html. These standards have been examined and we decided that they will work well for our project. Any modifications or additions to these standards that may arise are made by the Lead Programmer and followed by the rest of the team members.

4.3.4 Tools

Many tools have been and will be used for the project as outlined below.

Microsoft Word – Microsoft Word is the primary word processor the project will use for all of its documents. In addition, it will be used to help us design the digital cards for the application.

Microsoft Visio – Microsoft Visio is the primary tool we use for diagrams that are included in our documents.

Microsoft PowerPoint – Microsoft PowerPoint is the primary presentation application that we use to develop our document presentations.

Microsoft Paint – Microsoft Paint is a simple graphics program that we will use to take our card designs that are made in Word and make them into images that we can manipulate.

Net Beans – Net Beans is an open-source development tool that will allow us to create our Java Applet and its supporting classes.

WinSCP – WinSCP is a secure file transfer application that we use to upload/download all our files to/from the our Project Web Site and the Applet Website.

GIMP – GIMP is an open-source graphics program that we may rely upon for developing a more aesthically pleasing applet environment.

4.4 Configuration Management

Due to the small team size and modular design of the system, we feel that we do not need any specialized software for version control. Using specialized software would likely complicate the issue more than necessary. Instead, the Lead Programmer will handle the version control. Team members will be able to work on assigned modules and send them to the Lead Programmer when completed. The modules will then be implemented by the Lead Programmer generally during a scheduled team meeting.

The Lead Programmer will be responsible for clearly labeling each version with a version number and uploading the latest version to a secure online location accessible to all team members. All previous versions of the software will also be stored online in an organized manner.

4.5 Test Plan

We will use four types of testing with our system. We will use module testing, integration testing, system testing, and acceptance testing. Generic details on the types of testing can be found in our System Design Document in section 7.

Module testing will be preformed by the individual or individuals that created the module of code being testing. During a team meeting, the module may be tested more thoroughly by the team as a whole if it is a complicated module or has possible issues that need to be examined.

Integration testing will be preformed by the team during team meetings. Any member who has written a piece of code being integrated needs to be present for the meeting.

System testing will be done by the team during team meetings as well as by every team member individually. Due to the nature of the program, there is not a good way to develop a set of test cases since most aspects of the system rely on mouse clicking and dragging interactions with the user. During the initial testing stages, a small number of tiles will be used when testing the main arrangement mode. The movement of these tiles will be recorded by the tester. The tester's records will then be compared to the log files to make sure the logging is working correctly. These logs will then be organized, documented, and stored for use as test cases when testing the playback module. Once these modules have been determined to work correctly, more advanced testing will take place with the full number of tiles in the sets. Since it is likely that some of these aspects of the system will be tested before the creation subsystem is fully functional, it may be necessary to manually create a few small tile sets for testing. The creation subsystem will likely be a difficult system to test due to its complexity. Some general test cases will be created that thoroughly cover the full range of the subsystem. While testing the creation subsystem, the testers will be expected to document all the actions that they have made. These test cases with the additional documentation will be organized and stored with the created test sets. These sets they have created will then be tested with the other system modules once they are functional. The tester will be expected to compare each test set with what was expected of that set from the creation test's documentation for that set.

Each version of our prototype will go through acceptance testing with both our client as well as some volunteer students. Student testing will involve students using different sets to arrange their tiles into patterns. Once they have completed this, the results will be saved to files, organized, and stored for additional testing. Testers will quietly observer students as they use the system. Once each student is finished, the testers will question each student on the system gaining some valuable feedback on its usability and performance. Client testing will be handled in the same manner for the first portion. After that has been completed, the client will then test the playback system in the same manner using the result files from the student testing. The client will also test the creation subsystem similarly. Since creating a tile set will require a lot of time and effort, it is likely that the client will just create or

modify some small test sets during the creation subsystem acceptance testing.

5. Hardware and Software Requirements

Software –

The clip system is an online system. The system will require the users have a Java technology-enabled browser using version 1.4.2 of the Java Runtime Environment or higher. All users will access the system via the web browser.

Hardware –

No special Hardware will be required to use the system.

6. Deliverables

These documents are online and are available for the client to view at any time. They serve as the project documentation and an evolutionary understanding of the product between the development team and the client.

Requirements and Analysis Document –

This document describes the functionality of the system required by the client.

Design Document –

This document defines the architecture of the system and decomposes the system into smaller subsystems and objects. It also describes the methods of design and testing that will be used and illustrates the user interfaces.

Client Prototype –

This prototype will be located on a website that will later evolve into the permanent site for our client. It will only be accessible to our client and specific chemistry students at this time. The prototype will be used for demonstrational purposes and to test the user interface and system functionality. It will be available to our client after 12 / 07 / 2005.

User-Manual –

This manual will be presented with the software and will describe the functionality of the system in detail. It will explain how to utilize the system from the teacher and student perspectives. It will also include a developers

section that will contain the results form the testing, developing, and coding stages of this project.

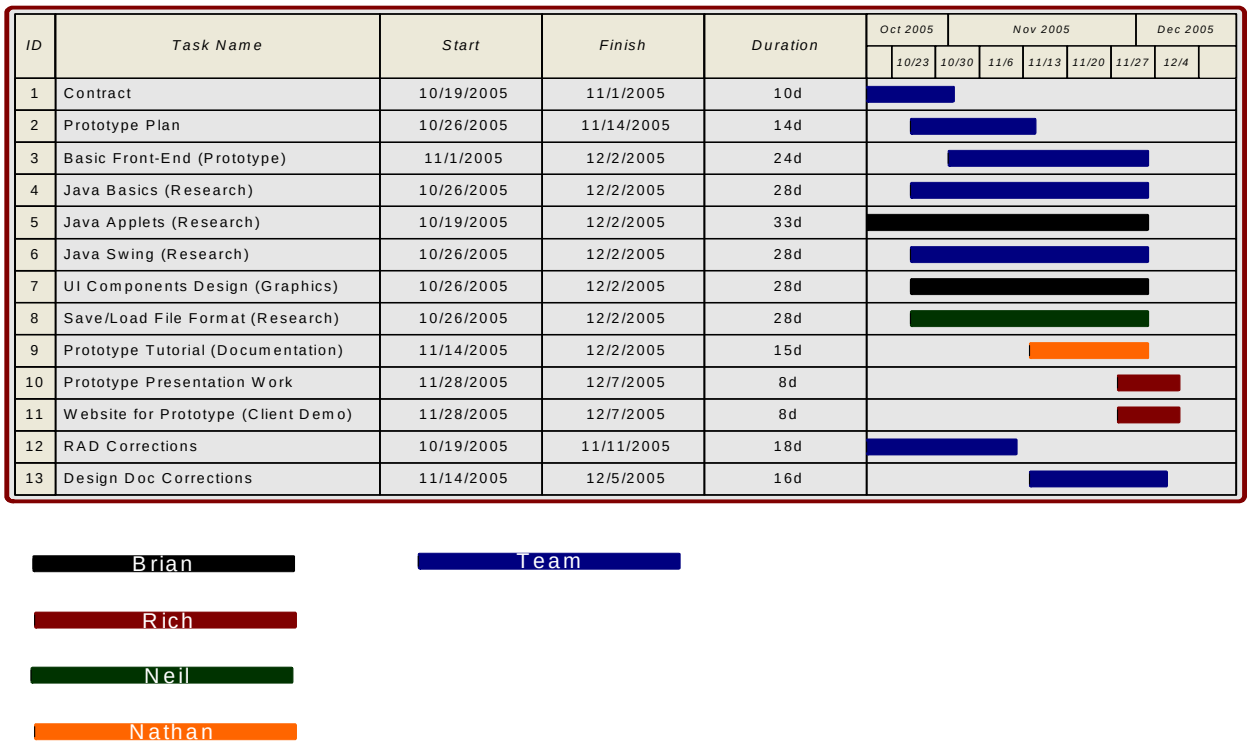
Clip System –

This is the final product. The clip system will replace the current system be online and accessible to anyone after 5 / 06 / 2005.

7. Efforts and Schedules

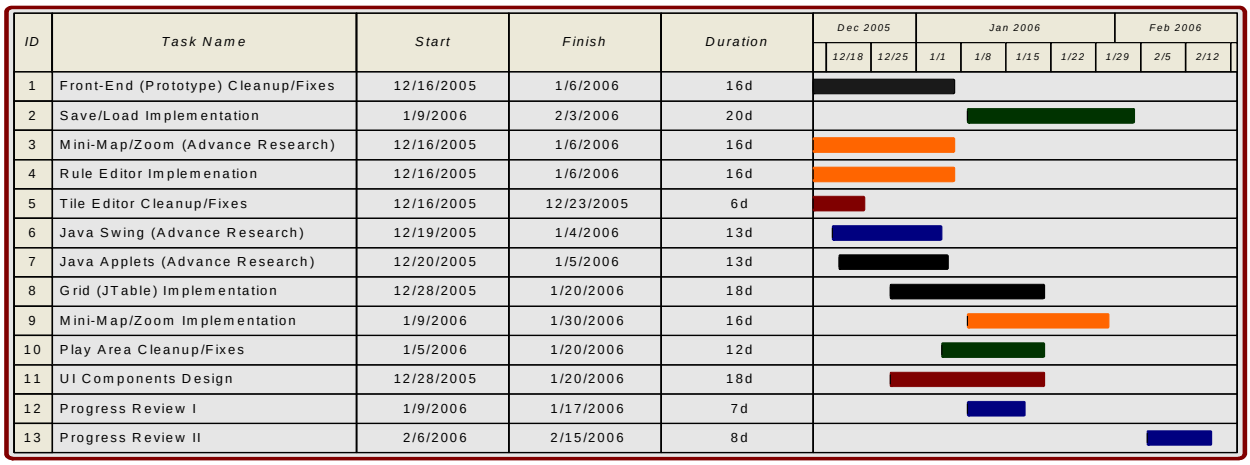
Schedule Part I

ID	Task Name	Start	Finish	Duration
1	Contract	10/19/2005	11/1/2005	10d
2	Prototype Plan	10/26/2005	11/14/2005	14d
3	Basic Front-End (Prototype)	11/1/2005	12/2/2005	24d
4	Java Basics (Research)	10/26/2005	12/2/2005	28d
5	Java Applets (Research)	10/19/2005	12/2/2005	33d
6	Java Swing (Research)	10/26/2005	12/2/2005	28d
7	UI Components Design (Graphics)	10/26/2005	12/2/2005	28d
8	Save/Load File Format (Research)	10/26/2005	12/2/2005	28d
9	Prototype Tutorial (Documentation)	11/14/2005	12/2/2005	15d
10	Prototype Presentation Work	11/28/2005	12/7/2005	8d
11	Website for Prototype (Client Demo)	11/28/2005	12/7/2005	8d
12	RAD Corrections	10/19/2005	11/11/2005	18d
13	Design Doc Corrections	11/14/2005	12/5/2005	16d



Schedule Part II

ID	Task Name	Start	Finish	Duration
1	Front-End (Prototype) Cleanup/Fixes	12/16/2005	1/6/2006	16d
2	Save/Load Implementation	1/9/2006	2/3/2006	20d
3	Mini-Map/Zoom (Advance Research)	12/16/2005	1/6/2006	16d
4	Rule Editor Implemenation	12/16/2005	1/6/2006	16d
5	Tile Editor Cleanup/Fixes	12/16/2005	12/23/2005	6d
6	Java Swing (Advance Research)	12/19/2005	1/4/2006	13d
7	Java Applets (Advance Research)	12/20/2005	1/5/2006	13d
8	Grid (JTable) Implementation	12/28/2005	1/20/2006	18d
9	Mini-Map/Zoom Implementation	1/9/2006	1/30/2006	16d
10	Play Area Cleanup/Fixes	1/5/2006	1/20/2006	12d
11	UI Components Design	12/28/2005	1/20/2006	18d
12	Progress Review I	1/9/2006	1/17/2006	7d
13	Progress Review II	2/6/2006	2/15/2006	8d



Brian

Team

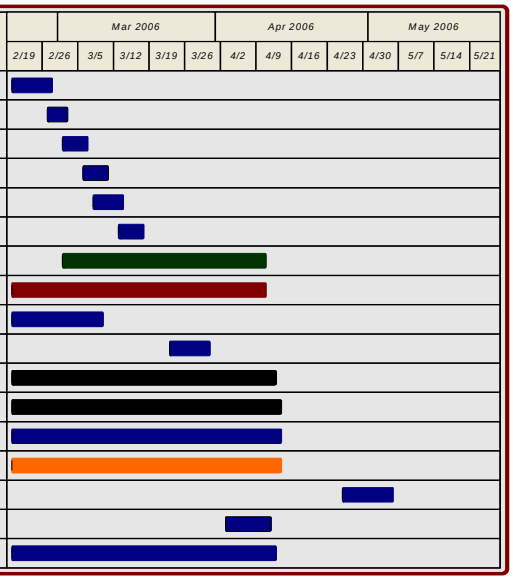
Rich

Neil

Nathan

Schedule Part III

ID	Task Name	Start	Finish	Duration
1	System Testing I	2/20/2006	2/27/2006	6d
2	Acceptance Testing I	2/27/2006	3/2/2006	4d
3	Acceptance Testing II	3/2/2006	3/6/2006	3d
4	Acceptance Testing III	3/6/2006	3/10/2006	5d
5	Acceptance Testing IV	3/8/2006	3/13/2006	4d
6	Progress Review III	3/13/2006	3/17/2006	5d
7	Full Manual	3/2/2006	4/10/2006	28d
8	User Tutorials	2/20/2006	4/10/2006	36d
9	Simple Patterns (Research-Impl)	2/20/2006	3/9/2006	14d
10	System Testing II	3/23/2006	3/30/2006	6d
11	User Interface Fixes	2/20/2006	4/12/2006	38d
12	Grid Fixes	2/20/2006	4/13/2006	39d
13	Recording Fixes	2/20/2006	4/13/2006	39d
14	Rule Editor Fixes	2/20/2006	4/13/2006	39d
15	Delivery Testing	4/26/2006	5/5/2006	8d
16	Acceptance Testing V	4/3/2006	4/11/2006	7d
17	Mini-map/Zoom Fixes	2/20/2006	4/12/2006	38d



Brian

Team

Rich

Neil

Nathan

Below is a description of the tasks for the team. The task names correspond to the previous diagrams.

Contract
This defines our Project Vision that lists the Client's requirements necessary and optional for the system.

Prototype Plan
This is a blueprint for the framework of our system. The framework will contain limited functionality but will show how the end product will look at this time.

Basic Front-End Development (Prototype)
A simplified version of the end-product that demonstrates for our client the virtual environment we are working to create. This will include basic functionality and a well-developed User Interface.
Some features include:
a. Menu System with Working options
b. Simplified Grid
c. Play Area (List of Cards, Mini-map outline, record-playback controls)
d. One type of card (Picture-Text)

Java Basics (Research)
Most of the team is unfamiliar with the Java Language, so a considerable amount of time will be spent early-on getting up to speed on the basic elements of Java.

Java Applets (Research)
Some time needs to be spent understanding the mechanics of the Applet. Also, we will be looking at the differences that exist between a stand-alone application and an applet in order to map out possible roadblocks that may prevent certain features from being fully implemented. Some research has already been done in this area concerning how to transform an applet to a stand-alone application.

Java Swing (Research)
The graphics package we are going to primarily use is called Swing. We are looking to use this because it provides many lightweight components (meaning speed friendly) that can form the foundation of our UI. This research will

primarily focus on getting the team up to speed on the various UI components of this package and how they can be used.

UI Component Design (Graphics)
A fair amount of graphical design will be necessary for developing the digital versions of the cards. This is because the limited space on the screen requires that we take careful time to maximize our use of it by not having too large of tiles. There are two possible types of cards right now. The first one is a combination of picture and text, and second is one that only uses text. The second one has varying amounts of text and we will have to figure out how to allow for our client's current cards to be mapped over without losing any information.

Save/Load File Format (Research)
We will be research how the recording file can be structured and what content it will have. In addition, we are going to be looking into what information our tile set files will have in them.

User Tutorials (Documentation)
We are going to develop a basic step by step of what the user can do with our prototype. This simple tutorial will form the foundation for the end-system tutorials.

Prototype Presentation Work
Before the Final Report is due, we will need to spend some time working on our presentation slides.

Website for Prototype (Client Demo)
Since we are developing an applet, we require the need for a website. Therefore, we will spend some time developing a simple website to host the system, which will later be the permanent site for our Client. However, for now the website will be basic enough to do a Demo session for our client to get some necessary UI and functionality feedback.

Front-End Cleanup
After our prototype is completed, we will begin working to refine our code and look for places to make it more reusable throughout. In addition, this is when we will work on our internal documentation for what we have done so far. In other words, we will focus on solidifying the framework of the system.

Save/Load Implementation

As soon as our research is done on the file formats, we will begin creating the format of the recording file to allow for proper saving and loading a user's solution. In addition, a small amount of time will be used to refine our format and structure for the tile set files.

Mini-Map/Zoom (Advance Research)
Research on will need to be done concerning how to implement a mini-map and how to implement zooming in order to decide which is more doable. As soon as that research is complete, we will create plan for the more practical implementation. This research will help us solve the screen space limitation problem by providing an alternative viewing method of the Grid.

Rule Editor Implementation
Here we will begin implementing the rules for any set created by the user. This will require us to fully implement a rule editor and modify our existing sets to conform to using this. Before the rule editor is fully implemented, the rules will be hard-coded into the sets. Rules include a description of the set, the type of cards, and how many there are in the set.

Tile Editor Cleanup
We will continue to work to clean up the code that handles tile creation for sets. This will include fixing any user interface discrepancies and user difficulties.

Java Swing (Advance Research)
In our prototype, we plan on using a general JTable implementation for our drag and drop grid. However, as soon as our research into the use of the more advance concepts of Swing is done, we will set up an implementation plan for a similar functioning arrangement area to the current system.

Java Applets (Advance Research)
Any problems we encounter with Applets, we will work to research them and discover solutions to those problems. This may not be necessary, but time is allotted for it. So far we have solved the security problem.

Grid (JTable) Implementation
Given the research goes well for advance Java Swing concepts, we will begin implementing our plan for a more realistic Grid to help with the feel of the new system.

Play Area Cleanup

We will need to periodically refine the UI of the Play Area and its method of listing cards based on how well our prototyping goes with actual users.

UI Component Design

We will continue to work on improving our tile designs and overall graphics of the system to satisfy our client's needs.

System Testing

There will be two major system testing sessions done. The first will occur after we have completed a working Grid System that works similar to the current system during the coding stage and the second will be once we enter the testing stage.

Acceptance Testing

There will be several sessions of acceptance testing done. These are focused on getting feedback from both our client and potential users. We plan to run these tests to focus on gathering how users feel the system works in comparison to the current system and where the system is lacking and exceeds. The number of users is undetermined, but it will be about 10 users per session so that we can monitor them using the system. The sessions include the tasks: Acceptance Testing I – V, Progress Review I – III, and Delivery Testing.

Integration Testing

During our weekly code reviews, we will work on integration testing of the changes made for that week. This will allow us all to discuss both erroneous and misleading events. The erroneous events are actual exceptions while misleading events are incorrect displaying of data by the system.

Full Manual

After most of the System has been developed and approved, the Team will spend some time slowly putting together a complete manual of the system. This will include both a Developers section and an 'all users' section. The developers section will be based on documents compiled during Testing, Design, and Coding. Another major source for this section will be the RAD and Design document. The 'all users' section refers to a user who wants to do everything with the system, like the professor. The information for this section comes from Acceptance Testing.

User Tutorials

A set of user tutorials will be available to help the student user get acquainted with the system. The information for this section comes from Acceptance Testing.

Simple Patterns (Research-Implementation)
This is an optional feature where we will allow the professor to create best-matching patterns that can be used to check a student's solution against. This will be researched and implemented given a fair amount of time remaining.

Code Fixes
These are special times set aside throughout the course of the project to allow us to fix any minor bugs we put off or were unaware of. Also, if we happen to change our minds on how something should be done and the team agrees, then this would be the time to change the code. In other words, this is a way to further improve our code's efficiency and speed by going beyond our usual code reviews. Lastly, this group of code fixes includes the tasks for Tile Editor Cleanup/Fixes, Play Area Cleanup/Fixes, User Interfaces Fixes, Grid Fixes, Recording Fixes, Rule Editor Fixes, and Mini-map/Zoom Fixes.

8. Measurements

Project progress will be tracked via two methods. The first method will be team log reports and the second method will be manager meetings.

Team Log Reports –

Each member of the team will write up a weekly log report. In each member's log, they will report on the work they have done throughout the week. At the end of each week, the team members will send their log reports to the project manager. The project manager will look through the logs of the other members and then write up a weekly overview on the progress of the team. This report will then be sent to the Senior Project Instructor.

Manager Meetings –

Every other week, the project manager will meet with the Senior Project Instructor and provide verbal progress reports and team reports. This is an opportunity for the Instructor to catch up on the current progress on the team as well as clear up any issues in which the instructor is unclear on.

9. Risk Management

Our biggest risk is the research we have to do before we even get into the developing our prototype. Since we chose to use Java, most of the team

has to read up on it and get familiar with how it works and where it differs from what they know already.

Furthermore, we have encountered a number of user interface difficulties, which revolve around dealing with the limited screen space. A contributing factor to the limited space problems deals with trying to accommodate for cards that have varying amounts of text (namely, the element cards have all types of information that vary in size) and how we can make the system environment more relatable to the real world. This requires us to spend additional time in the design stage while we code, therefore, potentially holding up some of the coding we need to get done but cant. Also, in order to deal with the UI issues we have, we are looking into developing a mini-map or zoom option for making the system more friendly when arranging the cards, but how to go about doing either one is not known to the team at this time. This requires additional research time.

Finally, the concept that we have of the Grid, which is to be a table that allows multiple targets to be used for a single card, is at the center of our difficulties. Its complexity is increased due to the lack of team knowledge concerning the area of Drag and Drop in general and in specific to how it works in Java. This will cause more time to be spent researching into how to bring the concept to reality.

In summary, all our risks revolve around requiring research into areas we have not yet been exposed to. Taking the time to research steals time from other areas that may need more time down the road. However, we have decided to combat this by assigning a team member to be a Risk and Development Officer to coordinate a modularized development schedule that allows us to work on multiple areas at once. In addition, this team member will continually monitor our progress with individual task checklists handed out to the other members and he will lead the integration process during our weekly code reviews.

10. Ethical Considerations

The ethical considerations that our group had to consider on this project included keeping the results secure. We did not want everyone to be able to see the results of the other students in the class. This is because a student could get the results from another student in order to copy their solution. The way we are planning to deal with this situation is to have the students e-mail the

solution to the professor. This will allow the e-mail service to make sure that the transfer of the file is secure.

Another possible ethical consideration is who we would allow to use the system. The client Susan Wiediger of the Chemistry Department wants other professors to be able to use the system. In order to deal with this we are planning on not requiring any sort of log on or user/passwords in order to access the program. This means that we will allow anyone to use the system, regardless of if they are a member of our client's chemistry class.

Another ethical consideration involving the available users is what platform we will allow this program to run on. If we developed it only for a Windows machine some of our client's co-workers may not be able to use the program. Because of which, she wants the system to be usable on different platforms besides Windows. In order to deal with this we are planning on developing the system with java.

The online nature of the planed java applet may require us to access the user's hard drive. We may need to be able to read and write specific files on the user's computer. These read/write privileges require us to think ethically. We should make sure that the user is aware of what we are doing, and that the program is not trying to do anything that may be detrimental to the user, or the user's computer (such as viruses or spy ware).

11. Glossary

Version #	Date	Author	Description
0.1	10/23/05	Nathan Mikeska	Initial creation and formatting of the document. Sections 3, 4, and 8.
0.2	10/23/05	Brian Navarro	Sections 7 and 9.
0.3	10/23/05	Brian Navarro	Additions to tools section, some formatting.
0.4	10/23/05	Neil Alfredson	Sections 1 and 10.
0.5	10/25/05	Richard Carney	Sections 2, 5, and 6.
0.6	10/25/05	Neil Alfredson	Additions to Section 10, grammer/spelling fixes.
1.0	10/25/05	Team	Reviewed and Accepted as v1.0