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/2005/fall/CLIP/

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.

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.

Project Manager

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. He will also handle communication between upper management and the team as well as communication with the client and the team.

Lead UI Designer & Lead Programmer

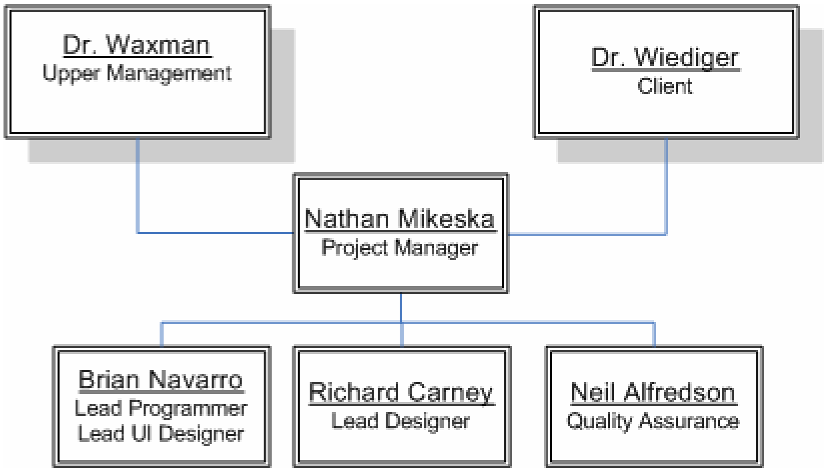
Brian Navarro will fulfill the role of Lead User Interface Designer as well as Lead Programmer. Brian is very talented in as well and interested in interface design and fits the Lead User Interface Designer role well. He is also the only member of the team that has had previous experience with programming 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 take the lead in any addition or removal of features that require changes to the system design.

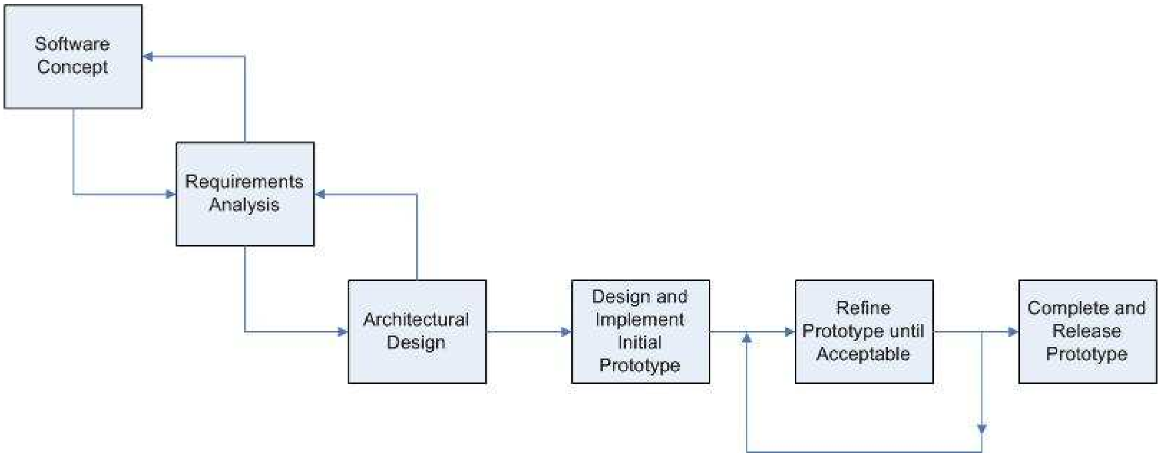
Lead Quality Assurance

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


Software Concept
Requirements Analysis
Architectural Design
Design and Implement Initial Prototype
Refine Prototype until Acceptable
Complete and Release Prototype

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	12/12/2005

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

Module Testing

As detailed by the lifecycle model our coding and testing phases will follow an evolutionary prototyping approach. Therefore code will be written onto the latest approved version of the prototype. This eliminates the need for imitation drivers or stubs as the modules completed will be tested using the latest accepted version of the prototype. An example of this would be the testing of the playback module. Since the grid and playarea modules are already part of the prototype by the time the playback module is scheduled to be completed, we can test the module by loading a log file that contains tile moves from the play area to the grid. Then the test will be to see if these moves are visually displayed.

Integration Testing

Once members have finished coding and testing the individual modules that they are working on, the code will be written into the main prototype. Even though members will be using their own copy of latest version of the prototype while writing their code, it is not unreasonable to expect that different modules will be coded by different team members concurrently. Therefore this testing is to assure that the code being written into the main prototype is compatible with another member's code and previously coded modules. An example of this would be testing the integration between the grid, tile arrangement, and playarea modules by moving a tile from the play area and placing it in the grid. The result would be immediately visible on the screen. The result can also be checked by looking at the log file created if the save arrangement button is clicked after the move.

System Testing

In our evolutionary prototyping development the system test will be a complete test over the systems expected functionality with the modules it contains at the time of testing. Since module and integration testing will be performed first, system testing is the last test to be performed before the prototype is advanced to the next acceptable version. This will be the version that the members will use as the driver to their new modules. The actual testing will consist of a repetition of all the module testing and integration testing that has taken place between version upgrades. System testing I will contain the tile arrangement, grid, and playback modules. System testing II will contain the previous modules plus the data module.

Acceptance Testing

After all of the functionality has been added to the system we will begin acceptance testing. During this testing the team will observe selected chemistry students using the system in the HCI lab. For this testing we are gong to enlist the help of ten students and six teachers that will be categorized into two separate groups and tested using two different methodologies.

Task Oriented

During these tests the user will be seated next to a team member in front of a computer running the system. The other team members will be in the other room of the HCI lab operating the cameras, VCR, and observing the test. The team member leading the test and next to user will be asking the user to perform a series of specific tasks on the system

Acceptance Testing I

User: Chemistry Student
Number of Users to be tested: 5

These tasks to be performed are:

- open the applet
- load the first tile set into the playarea
- reorganize the play area by taking the third tile and placing it in the first spot
- move all of the tiles to any locations on the grid
- arrange the tiles on the grid in the most logical arrangement according to the student
- move a specific tile from the grid to the playarea
- add a blank tile to the set of tiles
- save the final arrangement of the grid
- fill in the pop up form asking why they left the tile placed back into the playarea out of the final arrangement

During these sessions will be focusing primarily on the use of the system through the perspective of the student. Therefore, tasks that relate to the teachers role such as the playback of arrangements and tile set modifications will not be included. The team will record any mistakes the user makes, as well as their response as to why they made the mistake, while performing the tasks. Examples of mistakes may include clicking on the wrong menus or moving the tiles to wrong locations. We will also be keeping track of the time it takes the student to complete the task. From these tests we hope to discover any problems with interface that prevents the students from accessing the functionality of the system.

Acceptance Testing II

User: Chemistry Professor
Number of Users to be tested: 3

The tasks to be performed are:

- the same tasks as students
- open the recording the user submitted earlier
- move ten steps forward
- move a step backward
- auto play the rest of the moves till the last recorded move
- modify a tile set by adding another tile
- modify the same tile set by removing a different tile from the tile set
- create a new tile set from images preloaded onto the computer
- create a new best arrangement pattern for the new tile set
- adjust the rules of the new tile set to not allow students to input blank tiles
- adjust the rules of the new tile set to not allow students to add new best arrangement patters
- turn off the gridlines

These sessions will primarily focus on the use of the system through the role of the teacher. However they will need to complete the tasks the students have, because their understanding of the systems is more complex. The teachers will be evaluated and observed the same way as the students. During these tests we hope to locate any misunderstandings the teachers may have due to the interface organization.

Open Environment

These tests will be conducted similar to tests above. However the user will not be asked to perform or be guided through specific tasks. They will be given a brief description on the purpose of the system and be asked to simply use it. While they are guiding their way through the system the lead tester will be asking them questions about the language of the menu options, size of the forms, difficulty of scrolling, difficulty of using the minimap, colors of the forms, visibility of the information on the tiles, organization of the menus, and overall comfort with using the system. From these tests we hope to discover anything in the interface that is unappealing to the users.

Acceptance Testing III

User: Chemistry Student

Number of Users to be tested: 5

Acceptance Testing IV
User: Teacher
Number of Users to be tested: 5

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.

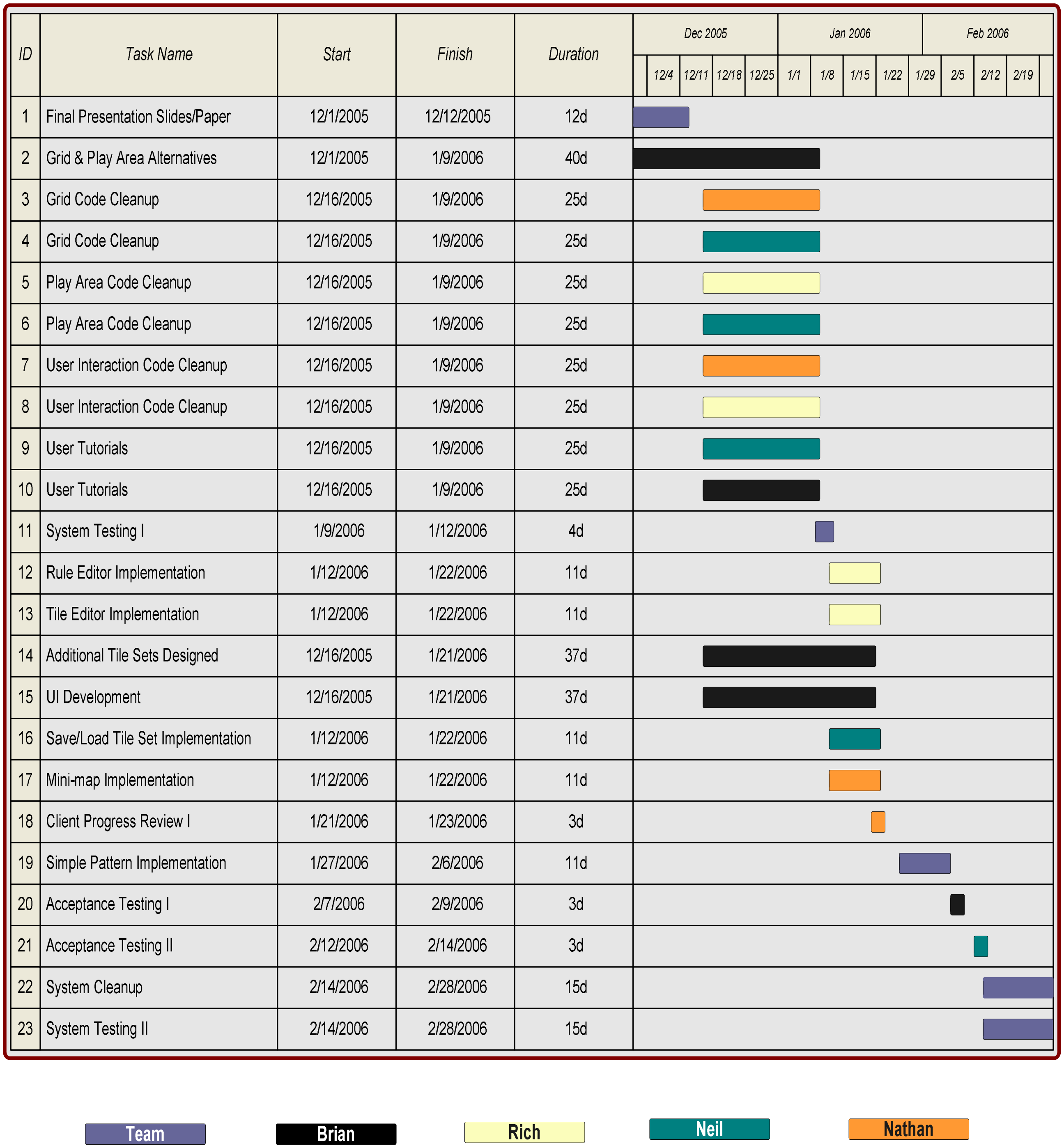
User-Manual –

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 –

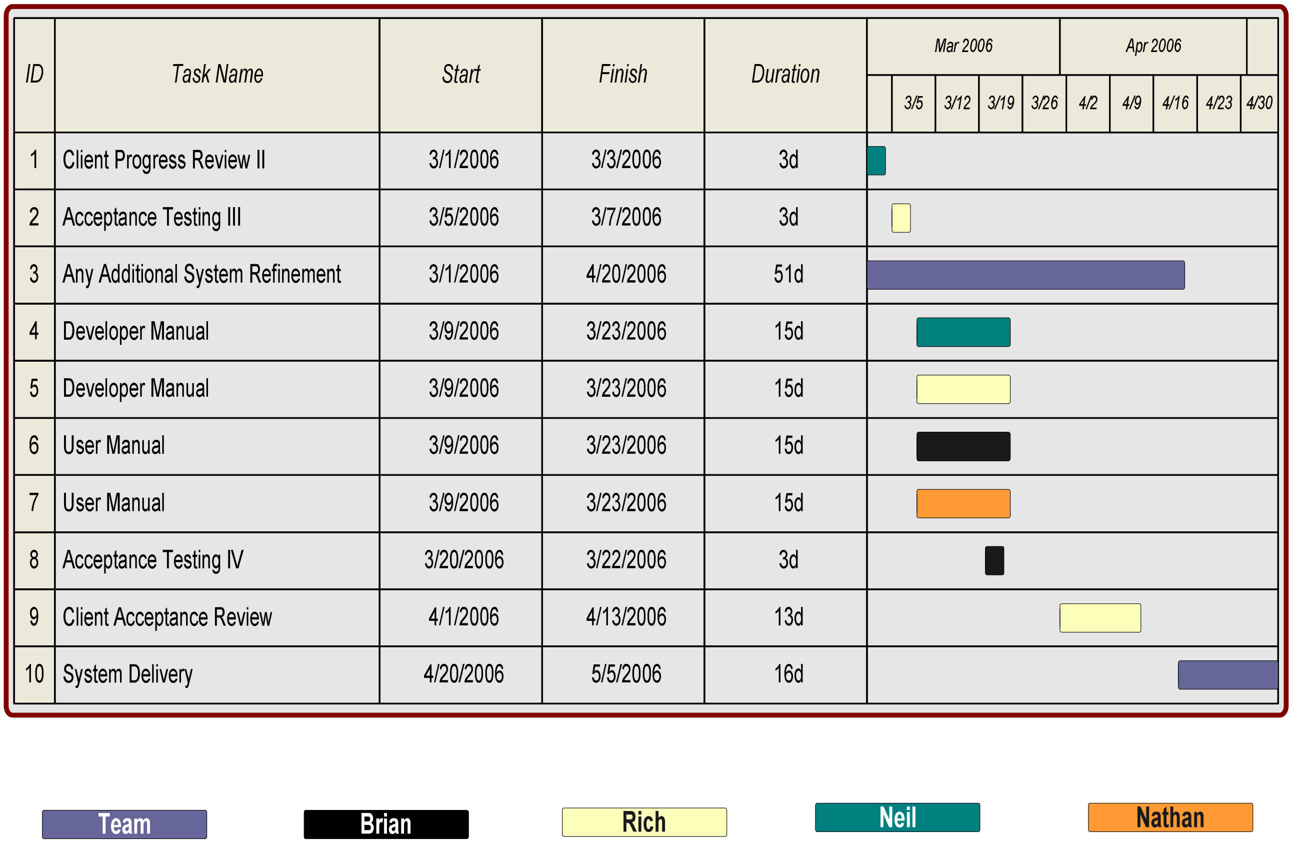
This is the final product. The clip system will replace the current system be online and accessible to anyone after 5 / 06 / 2005.

7. Efforts and Schedules



ID	Task Name	Start	Finish	Duration	Dec 2005 Jan 2006 Feb 2006
1	Final Presentation Slides/Paper	12/1/2005	12/12/2005	12d	🟪
2	Grid & Play Area Alternatives	12/1/2005	1/9/2006	40d	⬛
3	Grid Code Cleanup	12/16/2005	1/9/2006	25d	🟧
4	Grid Code Cleanup	12/16/2005	1/9/2006	25d	🟩
5	Play Area Code Cleanup	12/16/2005	1/9/2006	25d	🟨
6	Play Area Code Cleanup	12/16/2005	1/9/2006	25d	🟩
7	User Interaction Code Cleanup	12/16/2005	1/9/2006	25d	🟧
8	User Interaction Code Cleanup	12/16/2005	1/9/2006	25d	🟨
9	User Tutorials	12/16/2005	1/9/2006	25d	🟩
10	User Tutorials	12/16/2005	1/9/2006	25d	⬛
11	System Testing I	1/9/2006	1/12/2006	4d	🟪
12	Rule Editor Implementation	1/12/2006	1/22/2006	11d	🟨
13	Tile Editor Implementation	1/12/2006	1/22/2006	11d	🟨
14	Additional Tile Sets Designed	12/16/2005	1/21/2006	37d	⬛
15	UI Development	12/16/2005	1/21/2006	37d	⬛
16	Save/Load Tile Set Implementation	1/12/2006	1/22/2006	11d	🟩
17	Mini-map Implementation	1/12/2006	1/22/2006	11d	🟧
18	Client Progress Review I	1/21/2006	1/23/2006	3d	🟧
19	Simple Pattern Implementation	1/27/2006	2/6/2006	11d	🟪
20	Acceptance Testing I	2/7/2006	2/9/2006	3d	⬛
21	Acceptance Testing II	2/12/2006	2/14/2006	3d	🟩
22	System Cleanup	2/14/2006	2/28/2006	15d	🟪
23	System Testing II	2/14/2006	2/28/2006	15d	🟪

🟪 Team ⬛ Brian 🟨 Rich 🟩 Neil 🟧 Nathan



ID	Task Name	Start	Finish	Duration
1	Client Progress Review II	3/1/2006	3/3/2006	3d
2	Acceptance Testing III	3/5/2006	3/7/2006	3d
3	Any Additional System Refinement	3/1/2006	4/20/2006	51d
4	Developer Manual	3/9/2006	3/23/2006	15d
5	Developer Manual	3/9/2006	3/23/2006	15d
6	User Manual	3/9/2006	3/23/2006	15d
7	User Manual	3/9/2006	3/23/2006	15d
8	Acceptance Testing IV	3/20/2006	3/22/2006	3d
9	Client Acceptance Review	4/1/2006	4/13/2006	13d
10	System Delivery	4/20/2006	5/5/2006	16d

Effort Measurement

The duration times for each of the tasks mentioned below are only best-guess estimates. The best-guess estimates are based on normal workload expectations where the individual or individuals will do what is required within a reasonable time given the level of priority of a task. There are three levels of priority: high, normal, and low. Most tasks initially fall under normal priority or low priority, but there status changes as they become more important (or even less important, though rare).

High Priority – Needed as soon as possible for other areas of the project where without it, it would be difficult to move forward.
Normal Priority – Needed as soon as possible for other areas of the project but there is enough flexibility to allow for some delay.
Low Priority – Not necessary for other areas of the project right now but would be nice to have done as soon as possible.

Tasks

Below is a description of the tasks our team will undertake to complete the proposed system. The task names correspond to the previous diagrams. There are five attributes for a task: task name, purpose, resources (team member (s) assigned), priority, and dates active (start to finish dates). In addition, the resources named for a task are considered in charge of that tasks and are responsible for completing it. The persons in charge of a task are always free (in fact encouraged) to ask other team members for assistance.

Task Name: Final Presentation Slides/Paper
Purpose: Compilation of information from the RAD, SDD, and Project Plan and development of presentation slides based on the compilation.
Resources: All Team Members
Priority: High
Dates Active: 12-01-05 to 12-12-05

Task Name: Grid Code Cleanup
Purpose: Quality enhancements to the Grid implementation, which will result in a significant improvement in performance and usability.
Resources: Nathan & Neil
Priority: Normal
Dates Active: 12-16-05 to 01-09-06

Task Name: Play Area Code Cleanup
Purpose: Quality enhancements to the Play Area implementation, which will result in a significant improvement in performance and usability.
Resources: Rich & Neil
Priority: Normal
Dates Active: 12-16-05 to 01-09-06

Task Name: User Interaction Code Cleanup
Purpose: Quality enhancements to the User Interaction implementation, which will result in a significant improvement in performance and usability.
Resources: Nathan & Brian
Priority: Normal
Dates Active: 12-16-05 to 01-09-06

Task Name: User Tutorials
Purpose: Development of step-by-step webpage instructions on using the system. These will be what we use when conducting Acceptance Testing sessions.

Resources: Neil & Brian
Priority: Low
Dates Active: 12-16-05 to 01-09-06

Task Name: Rule Editor Implementation
Purpose: Implementing the read and write operations of the rule editor portion of the system.
Resources: Rich
Priority: Normal
Dates Active: 1-12-06 to 1-22-06

Task Name: Tile Editor Implementation
Purpose: Implementing the read and write operations of the tile editor portion of the system.
Resources: Rich
Priority: Normal
Dates Active: 1-12-06 to 1-22-06

Task Name: Additional Tile Sets Designed
Purpose: Creating images for new tile sets.
Resources: Brian
Priority: Normal
Dates Active: 12-16-05 to 1-21-06

Task Name: UI Development
Purpose: Improving the entire look and interaction with the user including the placement and organization of UI components based on team consensus, client progress reviews, and acceptance testing sessions.
Resources: Brian
Priority: Normal
Dates Active: 12-16-05 to 1-21-06

Task Name: Grid/Play Area Alternatives
Purpose: Researching an alternative implementation for the Grid and Play Area that uses a JTable for the Grid and a JList for the Play Area. These components are the involved in the typical approach to drag and drop functionality (therefore cleaner and more robust) but they will only be pursued if they are found to be capable of doing what the current implementation can do.
Resources: Brian
Priority: Normal

Dates Active: 12-01-05 to 1-09-06

Task Name: Save/Load Implementation
Purpose: Implementing read and write operations that relate to loading tile sets and previous arrangements, and saving current arrangements.
Resources: Neil
Priority: Normal
Dates Active: 1-12-06 to 1-22-06

Task Name: Mini-map Implementation
Purpose: Implementing a miniaturized rendering of the Grid
Resources: Nathan
Priority: Normal
Dates Active: 1-12-06 to 1-22-06

Task Name: Client Progress Review I, II
Purpose: Official demo sessions with our client to report on our progress and get feedback based on our latest prototype. The person indicated in the resources section leads the review meeting.
Resources: Nathan (I), Neil (II)
Priority: Normal
Dates Active: 1-21-05 to 1-23-05 (I), 3-01-06 to 3-03-06 (II)

Task Name: Simple Pattern Implementation
Purpose: Implementing an optional feature of simple pattern comparisons.
Resources: All Team Members
Priority: Normal
Dates Active: 1-27-06 to 2-06-06

Task Name: Acceptance Testing I, II, III, IV
Purpose: Official demo sessions with potential users to get feedback based on the latest prototype. . The person indicated in the resources section leads the acceptance session.
Resources: Brian (I, IV), Neil (II), Rich (III)
Priority: Normal
Dates Active: 2-07-06 to 2-09-06 (I), 2-12-06 to 2-14-06 (II), 3-05-06 to 3-07-06 (III), 3-20-06 to 3-22-06 (IV)

Task Name: System Testing I, II

Purpose: Functional testing completed by the Team to check that all functionality at the time works as intended.
Resources: All Team Members
Priority: Normal
Dates Active: 1-09-06 to 1-12-06 (I), 2-14-06 to 2-28-06 (II)

Task Name: System Cleanup
Purpose: Improvements made to the system during and after System Testing that relates to improving the performance and usability of the system.
Resources: All Team Members
Priority: Normal
Dates Active: 2-14-06 to 2-28-06

Task Name: Developer Manual
Purpose: A document based on our RAD, SDD, and internal code documentation that fully explains from a technical standpoint our system.
Resources: Rich & Neil
Priority: Normal
Dates Active: 3-09-06 to 3-23-06

Task Name: User Manual
Purpose: A document based on our RAD, SDD, internal code documentation, and user tutorials that fully explain in non-technical terminology how to use the system.
Resources: Brian & Nathan
Priority: Normal
Dates Active: 3-09-06 to 3-23-06

Task Name: Client Acceptance Review
Purpose: The system is presented to the client for final review. It's a sort of last chance review before we focus the rest of our efforts on documentation – both internal and external – and no more coding will be allowed unless absolutely critical. The person indicated in the resources section leads the review meeting.
Resources: Rich
Priority: Normal
Dates Active: 4-01-06 to 4-13-06

Task Name: Any Additional System Refinement
Purpose: Any final refinements to the system are done now.

Resources: All Team Members
Priority: Normal
Dates Active: 3-01-06 to 4-20-06

Task Name: System Delivery
Purpose: The system is officially complete and is delivered to the client via the web site that we establish. We present the client with a hard copy of both the user and developer manuals and list the soft copies of both on the web site.
Resources: All Team Members
Priority: Normal
Dates Active: 4-20-06 to 5-05-06

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 for making the system friendlier when arranging the cards, but how to go about doing this is not known to the team at this time. This requires additional research time.

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, grammar/spelling fixes.
1.0	10/25/05	Team	Reviewed and Accepted as v1.0
1.0 – 1.2	10/25/05 12/07/05	Team	Thorough revision of entire document. Most sections updated