

Chemistry Learning In Progress

Requirements Analysis Document

Nathan Mikeska

Neil Alfredson

Richard Carney

Brian Navarro

Table of Contents

1. Introduction	
1.1 Purpose of the System	2
1.2 Scope of the System	2
1.3 Objectives and success criteria of the project	2
1.4 Definitions, acronyms, and abbreviations	3
1.5 References	3
2. Current System	3
2.1 Overview	3
2.2 Functional Description	3
2.3 Example	4
3. Proposed System	6
3.1 Overview	6
3.2 Functional Requirements	6
3.3 Nonfunctional Requirements	8
3.4 Pseudo Requirements	10
3.5 System Models	10
3.5.1 Scenarios	10
3.5.2 Use Case Models	13
3.5.3 User Interface Mock up	27
4. Addendum	28
5. Glossary	29

1. Introduction

1.1 Purpose of the System

The purpose of this system is to provide students with an automated version of the current system. Therefore, this educational tool will have to retain the same objective of allowing a user to arrange freely a given set to form the patterns that make up the final arrangement. Each set in one way or another reflects the one of many ways the periodic table can be organized.

Furthermore, beyond mimicking the current system, we aim to provide observational data that shows the choices a user made toward finding a final arrangement for a set. This recorded data will help illuminate the various thought processes that occur for the user during their task of finding patterns within a set.

1.2 Scope of the System

The system should be developed with the scope objective of being available to many students within a professor's chemistry class and outside it. The client has suggested that this can be accomplished with applet, a web-application that has been used by many educational tools in the past.

The applet should meet the basic functions of the current system where it allows the user to use the program to organize the cards and arrange them as they wish. The students also need to be able to submit their final arrangement to the professor. The professor will then be able to view the arrangement and see the choices a student made to reach it.

Furthermore, the system should allow for the expansion of functionality and data. Therefore, the system should be well documented to allow expanded functions to be added easily later. Also, we will need to allow the addition of new card sets.

1.3 Objectives and success criteria of the project

The objective of this project is to satisfy and enhance the necessary requirements that are in current system and if possible, add new features. Such new features include allowing the teacher to check the end solution provided by the student against accepted patterns. Therefore, this

project will be successful if it can replace the current paper system. It will go beyond this criteria of success if it can add new features as well.

1.4 **Definitions, acronyms, and abbreviations**

C.L.I.P - Chemistry Learning In Progress

Professor (teacher) – A user that accesses the system from the perspective of a certain role. This role includes the use of the playback and creation functionality.

Student – A user that accesses the system from the perspective of a certain role. This role includes the use of the tile arrangement functionality.

User – Every user has access to the entire functionality of the system. Generally a user fulfills the role of a professor or student as defined above.

1.5 **References**

CLIP Project Page:

<http://www.cs.siu.edu/seniorprojects/2005/fall/CLIP/>

2. **Current System**

2.1 **Overview**

The current system serves two purposes. Firstly, it acts as an educational tool for students as they familiarize themselves with the various periodic elements, relationships, and patterns that those relationships may form. Furthermore, the system provides some insights into a student's thought process for a professor after they review the final arrangement. Presently the system is non-computerized and focuses on the interaction between the professor and the student through a set of paper cards.

2.2 **Functional Description**

Paper Cards

The student is given a set of randomly sorted cards that contain a number of characteristics on each. These characteristics combined, uniquely identify each card. However, the cards all share the same type of attributes, where the different values for them can be used to form the patterns that link them all together.

The Arrangement Area

This can represent any number of surfaces, preferably flat, and is where the student lays out their cards to begin the Card Sorting Activity.

Card-Sorting Activity

After the student has received a set of cards, they are asked to arrange the cards on a table according to how they identify the relationships among the differing valued characteristics on each the card. The student is not limited to any particular form of the arrangement. Any number of cards in the set may be excluded, if the student does not identify the cards as having a position in the final arrangement. In addition to the discarding cards, the student could discover that there are cards not provided in the original set that complete the relationships among the organized set. When this happens, the student creates new cards and places them into the arrangement.

Observation

While the card sorting activity is occurring, the professor observes the student. The professor makes note of each move of the cards and the time spent thinking between the moves. During the process, the professor may ask questions about a student's decision and offer feedback and suggestions.






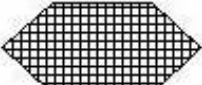





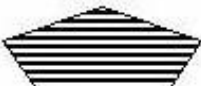
Patterns

A pattern is a final arrangement of the cards. While every card set has at least one best final arrangement, there exist other patterns that are acceptable.








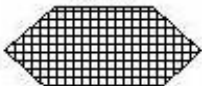



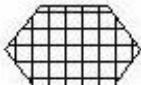
New Sets

Currently there are three sets of cards used in the system, but the professor can create new sets. Each set may focus on a different collection of relationships.





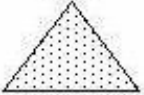

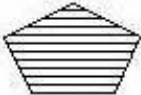
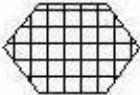
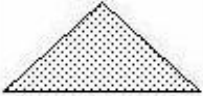


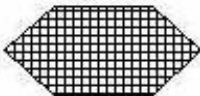
2.3 Example

<p>Reland</p>  <p>24.5</p>	<p>Razil</p>  <p>39.6</p>	<p>Hile</p>  <p>53.1</p>	<p>Aris</p>  <p>55.2</p>
<p>Ework</p>  <p>39.1</p>	<p>Orea</p>  <p>57.0</p>	<p>Labama</p>  <p>23.5</p>	<p>Eirut</p>  <p>20.2</p>
<p>Hana</p>  <p>36.7</p>	<p>Apan</p>  <p>40.9</p>	<p>Oston</p>  <p>23.0</p>	<p>Etroit</p>  <p>55.9</p>

A student is given a set of cards in random order.

<p>Hile</p>  <p>53.1</p>	<p>Aris</p>  <p>55.2</p>	<p>Razil</p>  <p>39.6</p>	<p>Reland</p>  <p>24.5</p>
<p>Eirut</p>  <p>20.2</p>	<p>Ework</p>  <p>39.1</p>	<p>Labama</p>  <p>23.5</p>	<p>Orea</p>  <p>57.0</p>
<p>Hana</p>  <p>36.7</p>	<p>Oston</p>  <p>23.0</p>	<p>Etroit</p>  <p>55.9</p>	<p>Apan</p>  <p>40.9</p>

The student decides to arrange the cards in columns of increasing number of sides.

<p>Eirut</p>  <p>20.2</p>	<p>Oston</p>  <p>23.0</p>	<p>Labama</p>  <p>23.5</p>	<p>Reland</p>  <p>24.5</p>
<p>Hana</p>  <p>36.7</p>	<p>Ework</p>  <p>39.1</p>	<p>Razil</p>  <p>39.6</p>	<p>Apan</p>  <p>40.9</p>
<p>Hile</p>  <p>53.1</p>	<p>Aris</p>  <p>55.2</p>	<p>Etroit</p>  <p>55.9</p>	<p>Orea</p>  <p>57.0</p>

The student sorts each column by increasing number.

The final figure represents the best possible pattern to sort the cards in. We see several patterns here: every column has one more side than the last, numbers increase from left to right, numbers increase from top to bottom, and the card patterns become darker from top to bottom.

3. Proposed System

3.1 Overview

The Chemistry Learning In Progress (C.L.I.P.) tool will be web-based. Its purpose is to replace the current paper version with an automated application that can handle the same card-sorting activities that are meant to mimic the organization of the periodic table of elements. The tool is educational based and will be used primarily by students and professors. The students will be presented with various sets of tiles that they must organize in best arrangement to create the pattern that links them all together. After the student has submitted their best answer, the professor will review the results.

3.2 Functional Requirements

The Grid

The Grid is where tiles are dragged to and organized to create the

pattern that forms the final arrangement. It should provide a similar feeling to that of the Arrangement Area used in the current system. This means that the Grid should feel as natural as possible and allow for any spot on the grid to be filled or left empty.

The Play Area

The Play Area is the area to the right of the Grid. It will display a scrollable list of tiles (they initially start here) along with a mini-map for viewing the grid more conveniently, and finally the playback controls for the tile arrangement recordings. The main component of the Play Area is of course the tile list and its purpose is to hold the currently unused tiles and potentially allow a student an area for organizing his/her thoughts on the sub-patterns of a set. In this list, the tiles should be fully visible similar to the current system so that user can easily differentiate between items.

Dragged Tiles

Any tile of a set can be dragged from its current location to a spot either on the Grid or the Play Area.

Dropped Tiles

When a tile is dropped, it must be on an empty grid space in the Play Area, else it will be returned to the location it was originally dragged from.

Creating a Final Arrangement

The student will be presented with a set of tiles that start out in the Play Area. They will have the task of forming the pattern that creates the best final arrangement. To do this, they will drag and drop all tiles they feel follow the overall pattern, leaving out any in which they feel does not adhere to the suggested pattern. Tiles are considered left out when they remain in the Play Area. Upon submitting the pattern, for each tile left out, the student will be asked to explain their reasoning for that by means of a dialog box. Likewise, the student will also be able to add blank tiles to take the place of missing tiles. For every blank tile added, they will need to include a description of how that tile fits in the pattern. They will do this via a dialog box that appears when they click the add blank tile button.

Tile Movement Recorded

As soon as the first tile is moved, the recorder will begin tracking the movements until the student is satisfied with his/her arrangement pattern and has saved the pattern for submission. Every tile move will be stored in memory and then written to a log file upon saving. The starting position and ending position of a moving tile will be what is recorded in memory. The professor will have the ability load a file into memory and to fast-forward, rewind, pause, etc. all of the recorded movements. This will allow the professor to carefully analyze how a student came up with a final arrangement pattern for a set.

Creation of New Sets of Tiles

The professor will be allowed to create new sets of tiles that can be added to the available sets. In all cases, the tiles will display images. These images will be created outside the system. The professor will also have the ability to create acceptable patterns as well as set rules for each new tile set. These rules include tile set name, description, size of tiles, ability to add blank tiles, ability to check pattern, ability to leave tiles out of the pattern arrangement, a show grid option.

Additional Sets of Tiles

An initial set of tiles will be preloaded by the application every time it is used. However, this is limiting, so the ability for the professor to add additional sets of tiles will be available.

Modification of Existing Sets of Tiles

The professor will be allowed to modify any existing set of tiles given that all the necessary information is properly provided.

Results Submitted

After the application is used with any set of tiles, those results may be submitted online. All the necessary data related will be sent to the professor through email is outside the scope of the system. Therefore, when the user saves his/her final pattern, they will be instructed that they simply must attach that file as an attachment in an email.

3.3 Nonfunctional Requirements

User Interface and Human Factors

The user interface will be intuitive and easy to use. Since the UI mimics the physical card sorting activity, it should feel as natural to the user as

the physical version. Little user training should be necessary and new users should quickly come to understand how the interface works.

Documentation

A set of online tutorials will be available to help explain the program to any users who require a better understanding of how the system works. A full manual will be made available with thorough explanations of every bit of functionality that the program provides. On the same page as the web application, a set of tutorials and documentation on the system will be provided through a series of html links.

Hardware Considerations

Since the system will be web based, users will need an internet connection and a standard internet browser that supports the Java Virtual Machine. The system will be designed for and tested to run on any type of machine that supports the Java Virtual Machine. Our primary goal is to allow systems that use Windows 2000, Windows XP, and Mac OS X.

The system will require a web server. The web server will need the bandwidth to support multiple file transfers for when students load up the applet initially. During this initial stage, the file transfers are simply the loading up of image data from the web server hosting the applet.

Performance Characteristics

The system should respond to user input with minimal delay once the applet has loaded onto the user's machine. There will be an initial delay that is dependent upon the user's connection as the applet class files that make up the system will be downloaded to their computer during that time.

Since most of the user interaction will involve dragging tiles with the mouse, the system should not exhibit any sort of lag so as not to frustrate or annoy the user. Users with dial-up connections should be able to load the system within a reasonable amount of time.

Error Handling and Extreme Conditions

In the event that the user loses connection to the internet, the system will support the ability to continue the user's session without any loss of work since the applet will be loaded on the user's local PC.

Quality Issues

The user interface will need to be easy to use and consistent throughout, only changing in form when expected and necessary. This system is primarily a front-end system with a small back-end for minimal processing, computing, and configuration. Therefore, a considerable amount of effort must be spent on testing and maintaining an understandable, useable, useful, and robust.

System Modifications

The system should support an easy and reliable way to create and use additional tile sets. The users should be able to use these additional tile sets with minimal effort. The teachers should be able to easily create or modify tile sets and provide them for students to use.

Physical Environment

No specific physical environment is required for the system.

Security Issues

The primary security issue is keeping the user safe while they use the applet. This means that we will only be allowed to read/write their local machine with their permission which is asked for before the applet starts loading. Therefore, the applet must be digitally signed by the team verifying it is safe to use.

3.4 Pseudo Requirements

Our client has indicated several constraints. These include the application being web-based, preferably as an applet since she has viewed many educational tools take this form.

In addition, the User Interface needs to demonstrate similar ease of use that is accomplished by hand with the paper cards.

3.5 System Models

3.5.1 Scenarios

Scenario: Tile Arrangement

Participating Actors: John: student user

Goal: Move the tiles into the desired arrangement and submit results.

Conditions: John is at a computer and ready to use the program.

Outcome: John arranges the tiles, saves his results and submits them using email.

- 1) John first opens up his web browser and types in the web address that the program is located at. John loads up the program.
- 2) The program asks John which tile set he would like to use for this session. John selects one of the three pre-loaded sets or a set stored on his own machine.
- 3) The program gives John all the tiles for the set that he selected and places them in the playarea. John begins examining and arranging the tiles in the playarea. When he has a decent idea of how he is going to try arranging the tiles, he starts moving his tiles from the playarea to the grid area. John makes many changes to the arrangement before he is satisfied.
- 4) John decides that one of the given tiles does not belong in his arrangement and he opts to leave it in the playarea excluding it from his arrangement.
- 5) John also decides that the set needs another tile to complete his arrangement. He selects the option to add a blank tile to the set. A blank tile is added and the system then prompts him to enter information about it regarding the values of its attributes. John enters this information.
- 6) John feels that he has completed his tile arrangement to the best of his abilities and decides that it is time to save his results. He selects the option to save his arrangement. The system prompts him to enter the reasoning why he left that specific tile out of the arrangement. The system then stores a log file onto his computer. John then sends this file to his professor via email.

Scenario: Results Playback

Participating Actors: Jane: teacher

Goal: To examine the arrangement and final solution of a student.

Conditions: A student has sent Jane an email containing a log file and Jane is ready to examine them.

Outcome: Jane thoroughly examines the student's results until satisfied.

- 1) Jane first opens up her web browser and types in the web address that the program is located at. Jane loads up the program.
- 2) Jane selects the option to view a results file. She then browses to the location of the students file she wishes to examine and selects it. The program loads up the file.
- 3) Jane then watches the tile movements that the student made. She often rewinds the playback or fast-forwards it. She eventually reaches the end and then closely examines the result of the student. At the end Jane is also shown the students reasoning behind the left out tile. Jane feels she has sufficiently examined the student's results.

Scenario: Create Tile Set

Participating Actors: Jane: teacher

Goal: To create a new tile set for the program.

Conditions: Jane has the program loaded up and is ready to design a new tile set.

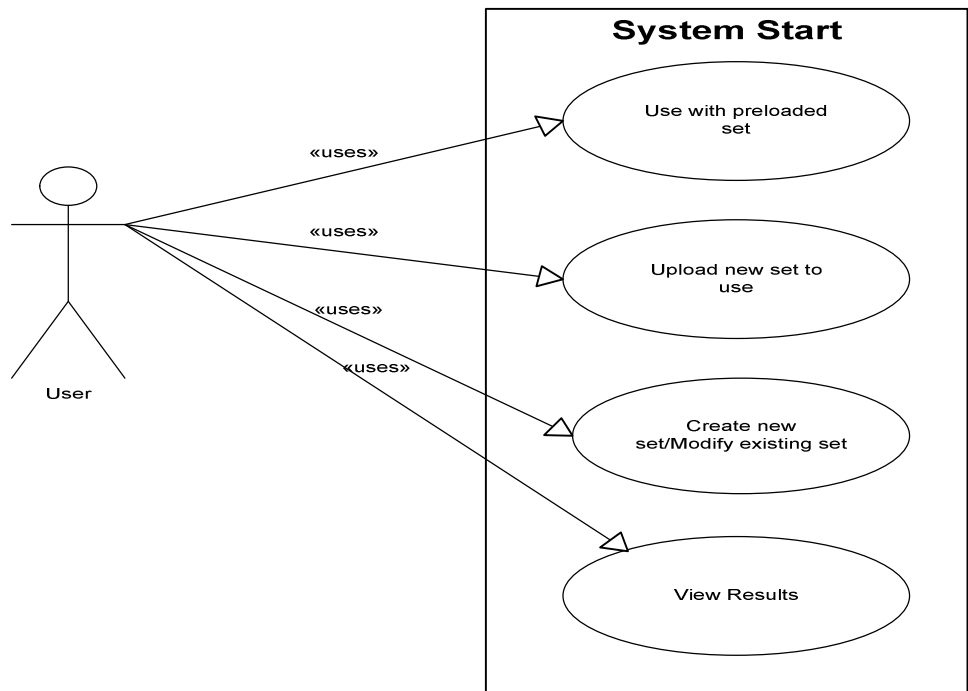
Outcome: Jane has created and saved a new tile set.

- 1) Jane has the program running and decides to create a new tile set. She selects the option to create a tile set located in the tools file menu.
- 2) A form is displayed that has three tabs on it. Jane is on the tile editor tab. Jane now has an empty set of tiles. She selects the option to add a new tile. She browses to the location of the images she created with a third party graphics program and selects one. The name of the text file is now displayed in the tile set list and the image is shown below in the image preview box.
- 3) Jane repeats step two until she has created all the tiles that she wishes the new tile set to contain. She then hits save.
- 4) Jane then selects the rule editor tab. There Jane enters the name of the new tile set and a description about the

tile set. A set of rules for the tile set (such as to allow addition of blank tiles or not) is also located here. Jane edits these rules until she is satisfied and hits the save button.

- 5) When Jane has finally created all of her tiles, she then begins to arrange them in different patterns. When complete with each pattern, Jane selects the option to add the pattern to the tile set. With each pattern, the program asks Jane to set the 'correctness' level of each. When satisfied she hits the save button.

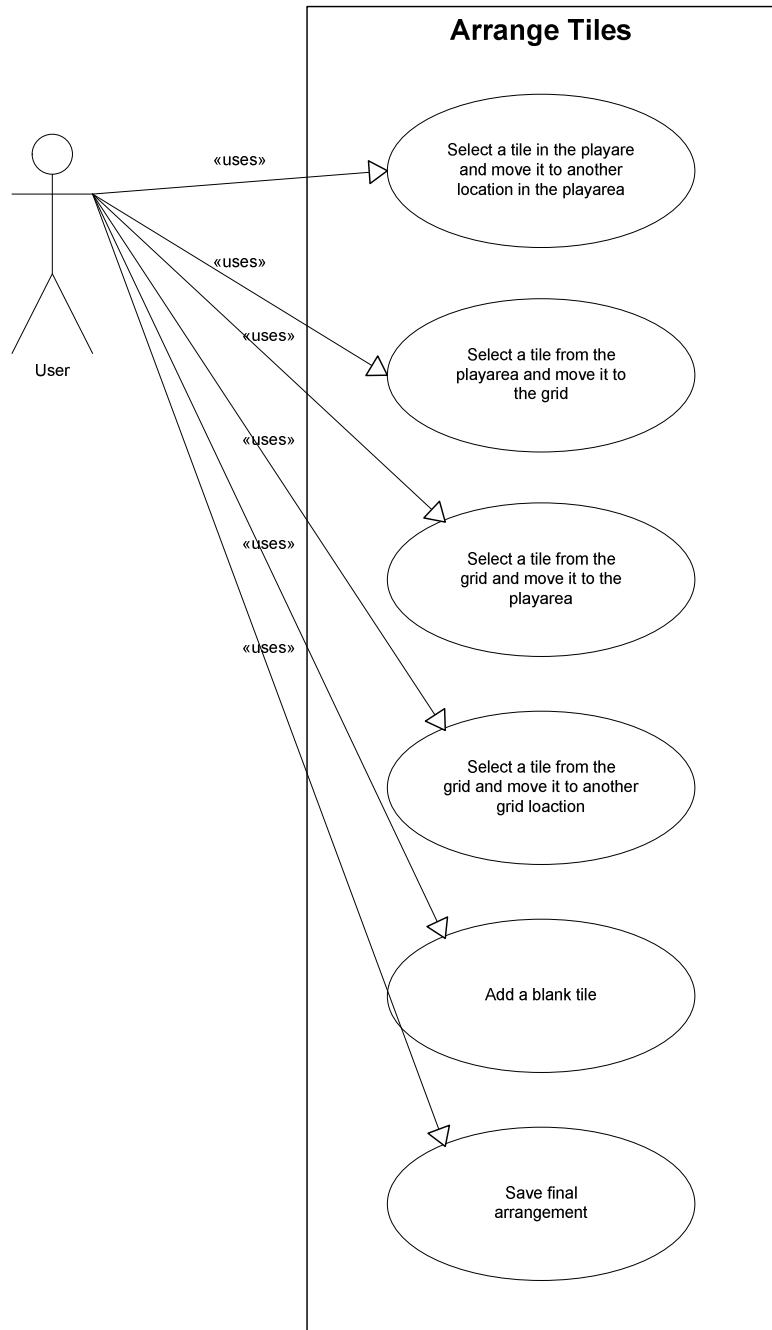
3.5.2 Use Case Models

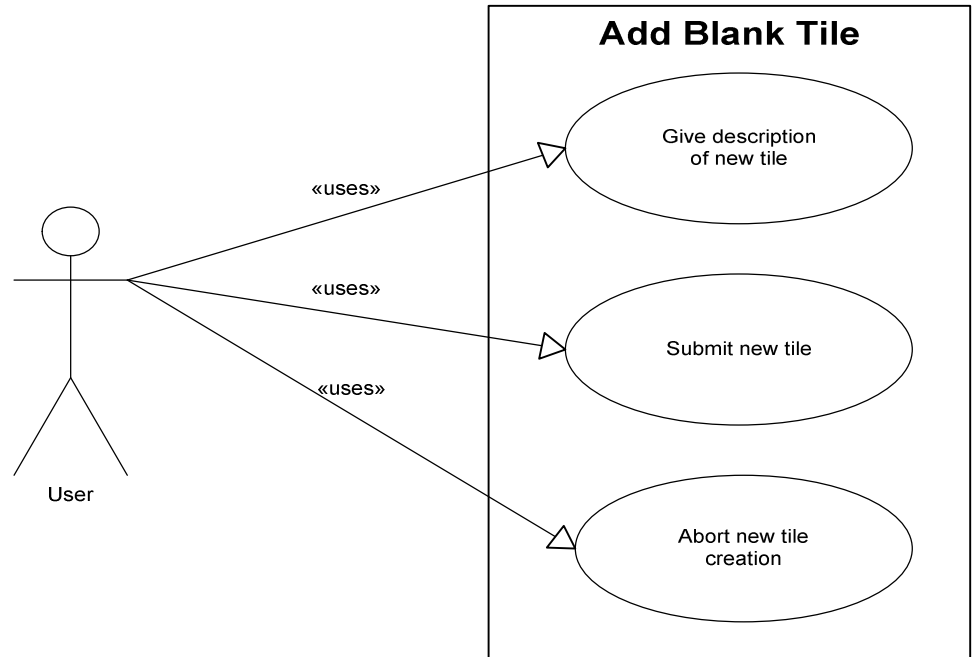


Use Case Name: StartProgram
Participating Actors: User
Entry Conditions: The user is at a pc with internet access and ready to begin using the system.
Exit Conditions: The user is connected to the system and ready to begin using it.
Flow of Events:

- 1) The user opens a web browser and browses to the page where the system is located.

- 2) The user starts the applet up.
- 3) The user is greeted with an empty grid screen and empty playarea screen.
- 4) The user then clicks on the file menu. There he or she has the option to load a tile set or open a recording.
- 5) If the user opts to begin with a preloaded set, he or she is then prompted to select one of the preloaded sets to begin using it. The system enters Tile Arrangement Mode. If the user selects to create or modify a set, the system enters Creator Mode. If the users opts to view a results file, the system enters Playback Mode





Use Case Name: AddBlankTile

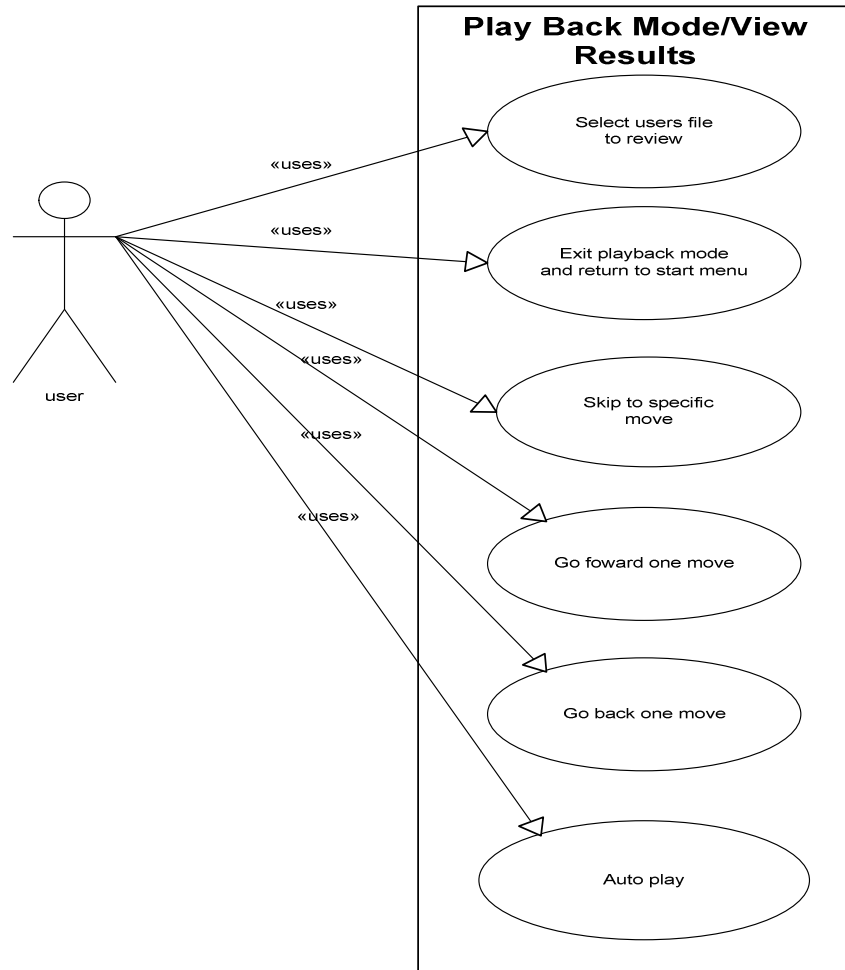
Participating Actors: User

Entry Conditions: The user is in Tile Arrangement Mode and selects the option to add a blank tile.

Exit Conditions: The blank tile has been added to the current set or the user has canceled.

Flow of Events:

- 1) The add blank tile dialog box appears.
- 2) The user is prompted to give a description of the tile (give information concerning the attributes of the tile, which vary depending on the set).
- 3) Once the user feels he or she as given a good enough description of the blank tile, they click the add button and the blank tile is added to the free-floating area. Alternatively, the user can click cancel to abort the blank tile addition.



Use Case Name: PlaybackResults

Participating Actors: user

Entry Conditions: The system enters Playback Mode

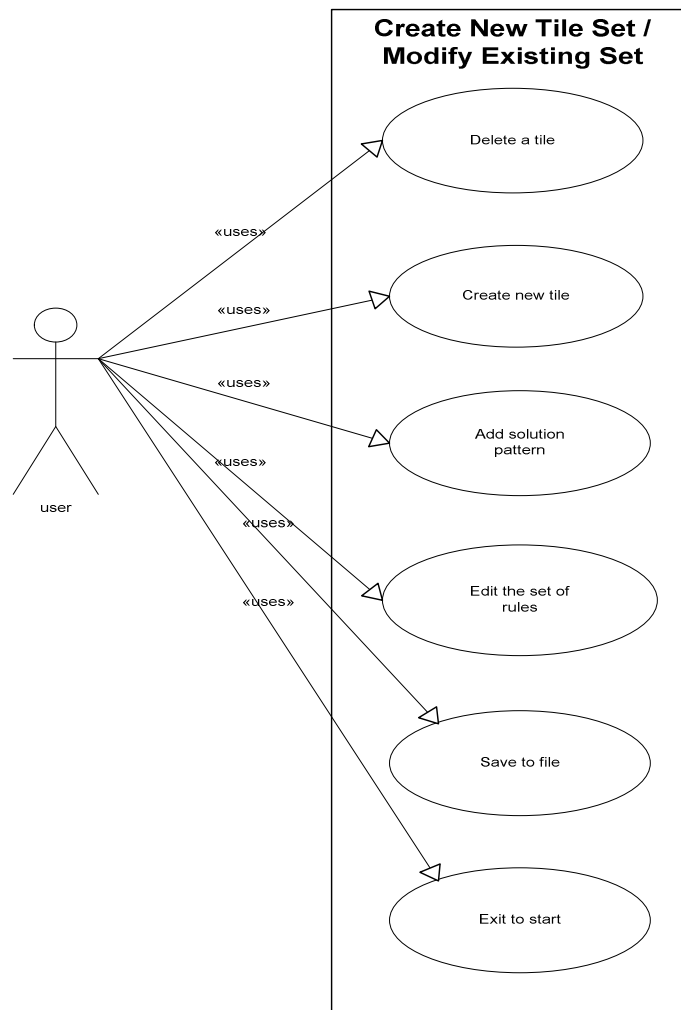
Exit Conditions: The system leaves Playback Mode

Flow of Events:

- 1) When the system enters Playback mode, it prompts the user to select the file to view.
- 2) The user browses to the file he or she wishes to view and selects it. The user can also cancel which leaves Playback Mode and returns the system to the starting menu.
- 3) The system loads up and analyzes the file.
- 4) The system displays the tile arrangement starting from the first move. The user can move forward or backward one move at a time with the forward or backward buttons. The user can also skip forward and

backward a specific number of moves. The user can put it in auto mode, which plays the moves one after another without input from the user. Auto mode can move forward or in reverse at a few different speed levels. The user can also opt to skip to a specific move number. The system displays the total time that has passed since the last move as well as the total time that has passed since the first move.

- 5) When the user has viewed the results file as much as he or she wishes, they can select the option to leave Playback Mode and then the system takes them back to the starting menu.



Use Case Name: TileSetEditor

Participating Actors: user

Entry Conditions: The system enters Creator Mode.

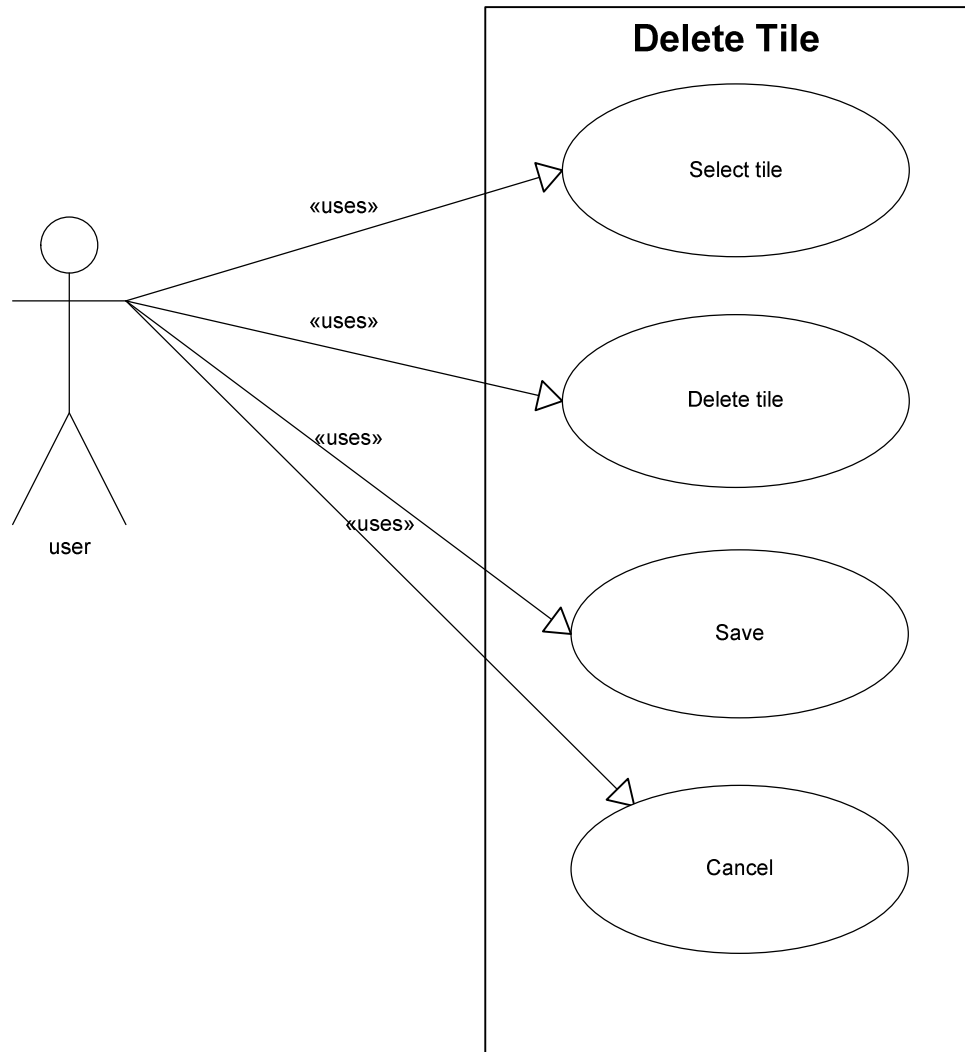
Exit Conditions: The system leaves Creator Mode.

Flow of Events:

- 1) After entering Creator Mode, the user has the option to open an existing set to edit or to create a new set.
- 2) Selecting to create a new set calls EditTileAttributes. Selecting to modify a set prompts the user to select an existing set to modify. When the user selects a set, it

loads up all tiles associated with that set into the playarea area.

- 3) From here, the user has several options:
 - Delete a tile – Calls DeleteTile
 - Create a new tile – Calls CreateNewTile
 - Add a solution pattern to the set – Calls AddPattern
 - Edit the rule set for a set – Calls EditSetRules
- 4) The Instructor has the option to save the tile set or return to the starting menu forfeiting all changes he or she has made.
- 5) If the user chooses to save the tile set, the system will prompt him or her to enter a filename and choose the location (on the local PC) to save the tile set.
- 6) The user then has the option to continue in Creator Mode or exit to the starting menu.

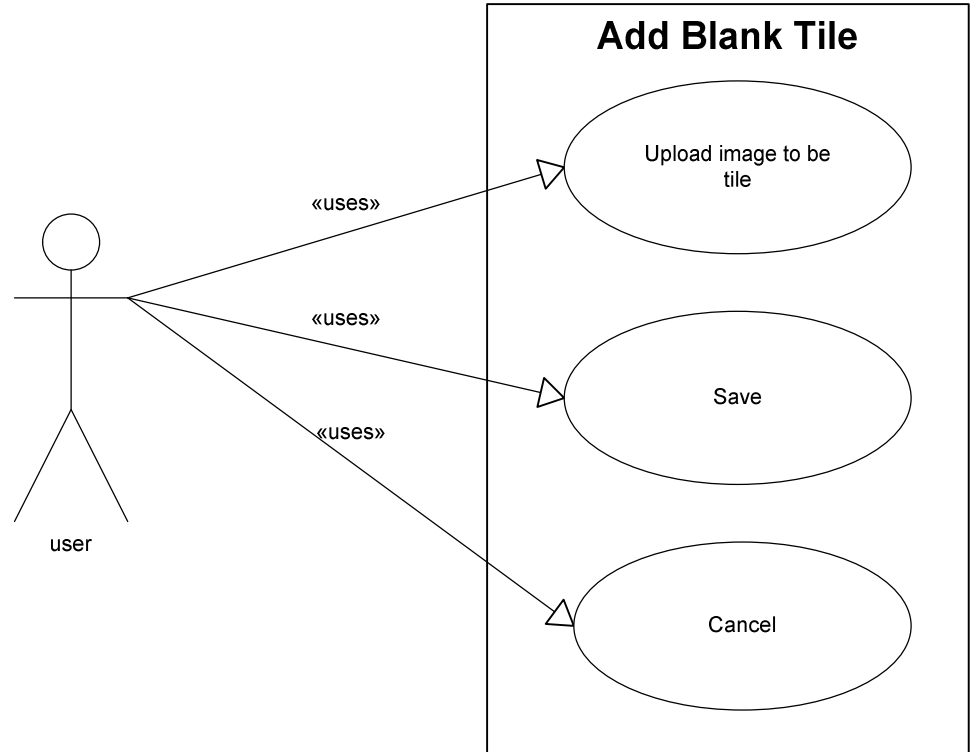


Use Case Name: DeleteTile
Participating Actors: user
Entry Conditions: The system is in Creator Mode and has a tile to be deleted.
Exit Conditions: A Tile has been deleted or the operation is canceled

Flow of Events:

- 1) The user selects the tile to be deleted from the list of tiles in the set.
- 2) The user selects the delete option.
- 3) The system then prompts the user asking if he or she is certain that they want to delete the tile.

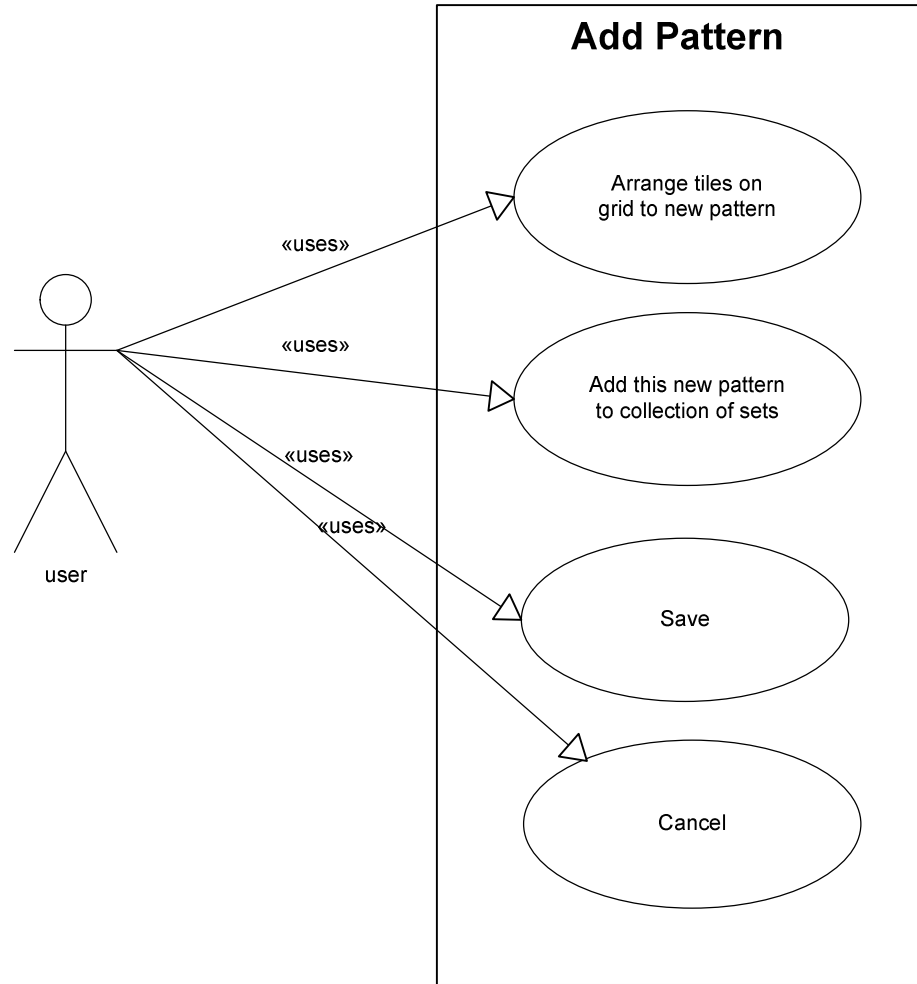
- 4) If the user selects yes, the tile is removed from the tile set. If the user selects no, the operation is aborted. The user can then save or cancel.



Use Case Name: CreateNewTile
Participating Actors: user
Entry Conditions: The system is in Creator Mode.
Exit Conditions: A new tile has been created or the operation is aborted.

Flow of Events:

- 1) The user selects the option to add a new tile to the set.
- 2) The user uploads an image to serve as the tile.
- 3) User saves or cancels tile set.



Use Case Name: AddPattern

Participating Actors: user

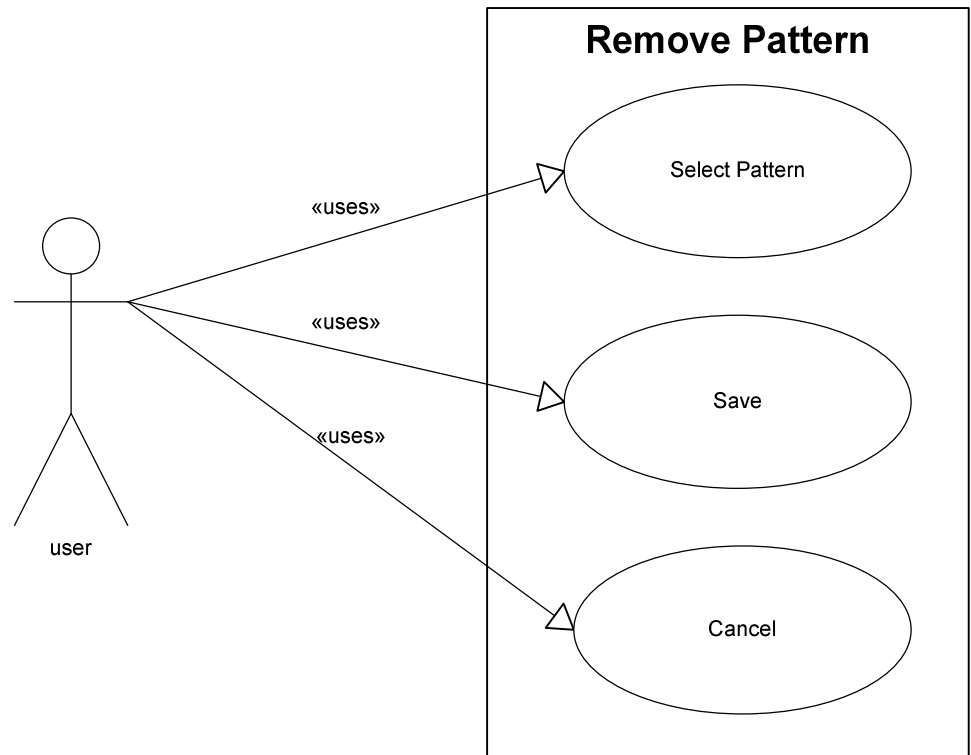
Entry Conditions: The system is in Creator Mode and at least 1 tile is in the set.

Exit Conditions: A new pattern has been added to the solution set for the current tile set or the operation has been canceled.

Flow of Events:

- 1) The user arranges the tiles in the grid area until they form the pattern they wish to add to the tile sets solution set.
- 2) The user then selects the option to add the pattern to the solution set.

- 3) The system prompts the user to enter the ‘correctness’ rating of the current pattern as well as a name for the pattern.
- 4) The system then prompts for confirmation. If save is selected, the pattern is saved in the solution set for the tile set. If cancel is selected, the pattern is not added and the system returns to creator mode.



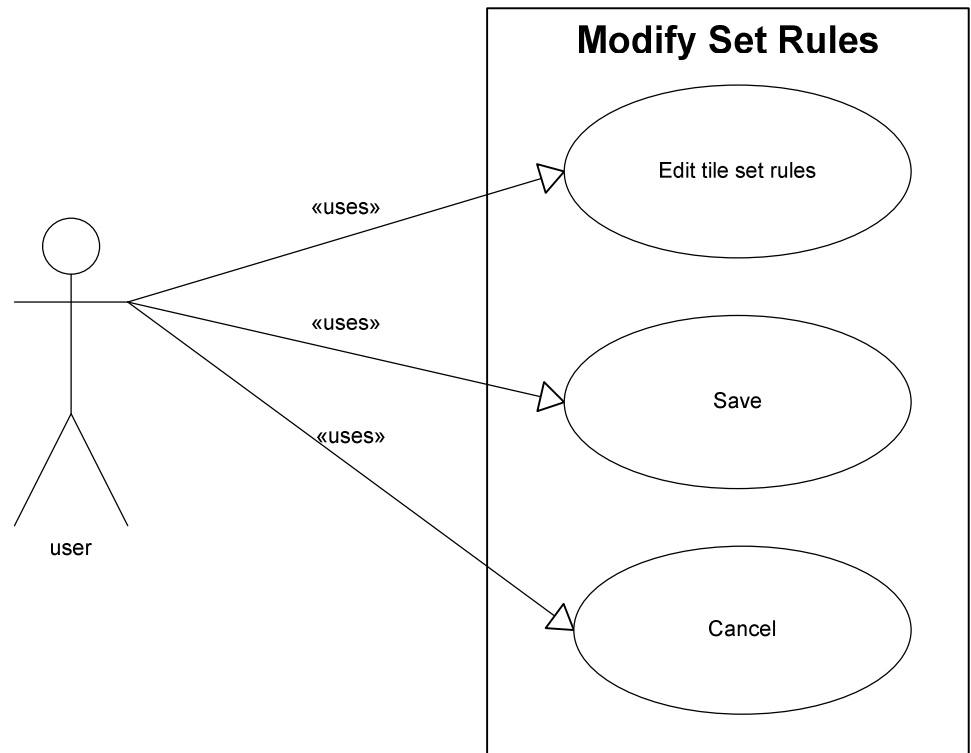
Use Case Name: RemovePattern
Participating Actors: user
Entry Conditions: The system is in Creator Mode.
Exit Conditions: A pattern has been removed from the set’s solution set, the operation has been canceled, or the user opts to view a pattern.

Flow of Events:

- 1) At any time in Creator Mode, the user can select the option to remove a pattern from the set’s solution set.
- 2) A list of patter names is then displayed. The user has the option to cancel and return to Creator Mode or they can select any one pattern and select to view the pattern

or remove the pattern. If the view pattern is selected, RemovePattern exits and the tiles are arranged in the grid of Creator Mode to match the pattern that was selected.

- 3) If the user opts to remove a pattern, a prompt asks for confirmation. No will return to step 2. Yes will remove the selected pattern from the set and then return to Creator Mode.



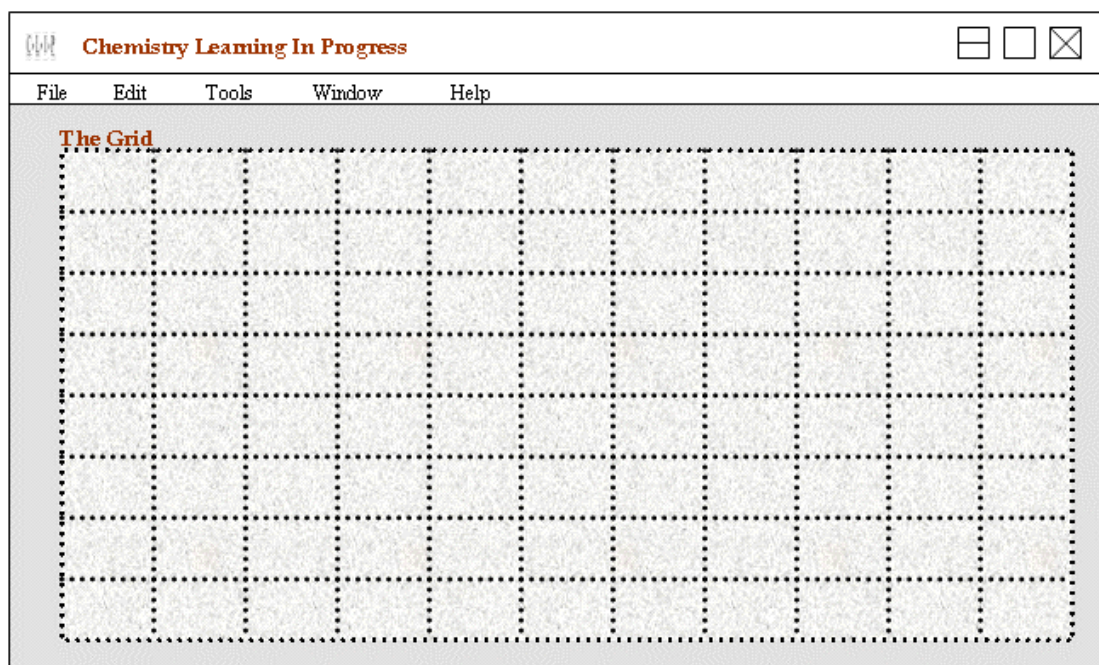
Use Case Name: ModifySetRules
Participating Actors: user
Entry Conditions: The system is in Creator Mode
Exit Conditions: Set rules have been modified or the operation has been canceled.

Flow of Events:

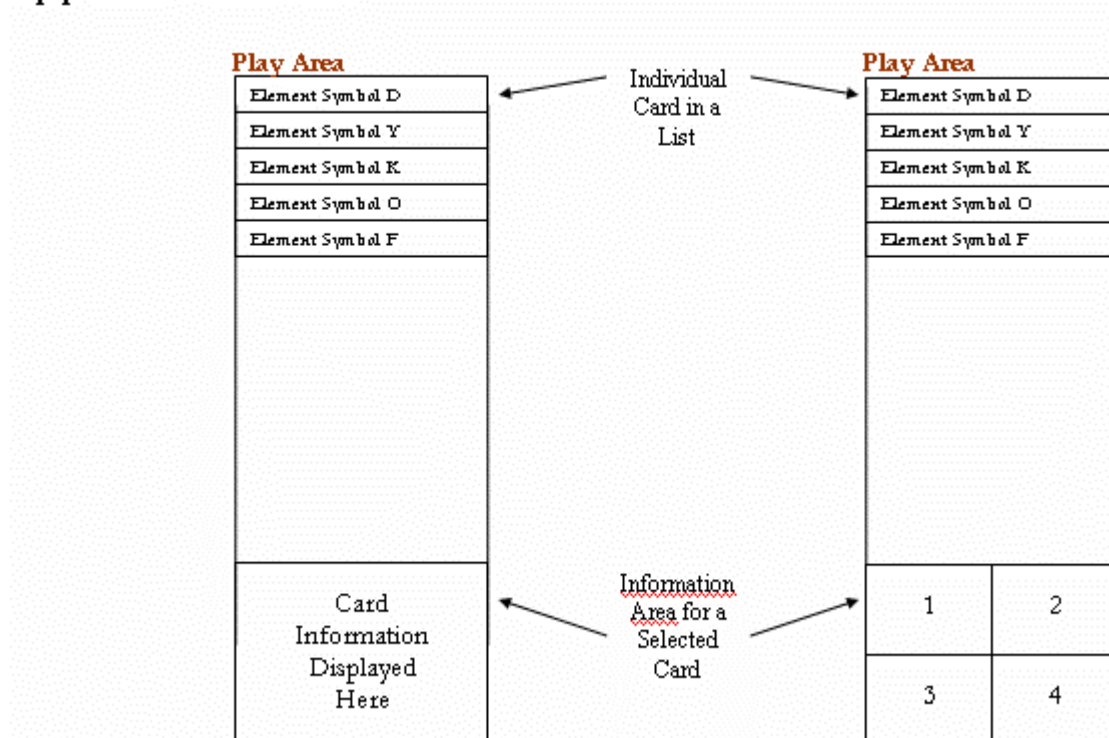
- 1) The user can select the option to modify the tile sets rules while in Creator Mode.
- 2) This brings up a list of tile set rules and their current settings. The user can edit the rule settings at will.

- 3) The user then has the option to cancel and return directly to Creator Mode or to save the changes to the set rules and then return to Creator Mode.

3.5.3 User Interface Mock up



Popup-Window – Version 1 and 2



4. Addendum

During implementation of the application some of the requirements and specification for the software that we were developing changed. Through discussion and meetings with our client it was decided that the pattern checking system would not be a necessary feature for the program. The team and client came to the agreement that the functionality that would be provided in this feature would not be adequate to make pattern checking a useful feature. Do to this and the fact that it was never a required feature for the team to implement it was removed from the application.

There were also additional features that were suggested to the team by the client in the middle of implementation. The client came up with these additional features which the team decided were able to be implemented. The team decided that some of these features could greatly help the use and effectiveness of the program. Therefore, these additional features were added to the functionality of the program. The added features included were printing, continue/save sessions, resizable and random tilebin, randomly remove tiles, and overall comments.

The printing of tile arrangements is an option that would allow the user to print their results and be able to turn in a printout of the arrangement. The printing capabilities also include the ability to send all of the information about an arrangement and create an image file of it. The team also believed that the application should provide the ability to print out additional information that the teacher might want to know. This additional information that can be printed includes the name and description of the blank tiles and all of the information a user may have provided about left out tiles.

The client suggested that a user might want to be able to start working on an arrangement and stop with the intention of coming back later. This means we would need to provide a way for the user to save the information that they currently have and allow them to load up what they had and continue to work on it. This is why continue / save session option was added to the features of the program.

In order to help make the program better it was decided that the tilebin should be resizable in case a user finds it easier to make an arrangement inside of the tilebin instead of the grid area. Our client also suggested that when the tilebin is loaded it should be randomly sorted.

Another feature that our client decided she wanted in the program is the ability to randomly remove tiles from a set. This could be used to see if the student would notice that a tile is missing and add it into the pattern as a blank tile.

One of the last features that we added to the program was the ability for a user to comment on the overall arrangement that they came up with. This would be a general comment about why they arranged the tiles in the set the way that they did.

5. Glossary

Attribute: An attribute refers to an attribute on a tile or a card. Each set has its own set of attributes and each tile or card within a set has a set of values for those attributes that make that card or tile unique.

Blank Tile: A blank tile is a tile added to a set in Tile Arrangement Mode. A blank tile is not a permanent addition to the set. The user can edit a blank tiles attributes. The purpose of a blank tile is to fill in holes in a user's pattern (similar to undiscovered elements on the periodic table).

Card: A card is the physical version of a tile. A card belongs to a set. Every card in a set has the same list of attributes. The values for the attributes vary from card to card. Multiple cards can have the same values for some attributes but the whole set of values for each card should uniquely define that card for that set.

Comparison Pattern: This is one rule of a set that determines whether or not a pattern will be provided to allow the comparison of a user's tile arrangement against what is considered a 'best' or logical tile arrangement.

Creator Mode: The system provides the ability for sets to be created and modified given the user-chosen rules, tile images, and comparison pattern(s).

Play Area: This is the area to the right of the Grid. It contains a list of tiles, a mini-map, and a series of playback controls. The primary component of the play area is the list of the tiles where the tiles are shown when the system starts up. Tiles in this list can be moved around by the drag and drop action. Using the drag and drop motion, a user is allowed to move tiles to another position in the list of cards or to the Grid. The mini-map component of the Play Area is for showing a miniaturized version of the Grid. Finally, the playback controls allow for the playing of a recorded tile arrangement.

Pattern: A pattern refers to any kind of arrangement of tiles on the grid.

Playback Mode: The program enters this mode when the instructor is viewing a student's results. In this mode, the instructor can move through the steps the student made while forming a pattern. It supports fast forward and reverse functions as well. Its purpose is that of a possible grading technique as well as a research tool to observe the thought processes of students while performing this activity.

Pre-Loaded Set: A pre-loaded set is a set of tiles that has been pre-loaded on the server. This allows users to load up the applet and start using it with the provided data.

Rules: These are attributes that are a part of the set to distinguish them from other sets. Such rules include a set name, description, type (2 types only), and whether or not comparison pattern(s) are present.

Set: A set is a collection of tiles or cards. Each set has a list of attributes that are common to all tiles or cards within it. Furthermore, the system supports the option to create or modify sets.

Tile: A tile is the electronic version of a card. A tile belongs to a set. Every tile in a set has the same list of attributes. The values for the attributes vary from tile to tile. Multiple tiles can have the same values for some attributes but the whole set of values for each tile should uniquely define that tile for that set.

Tile Arrangement Mode: Tile Arrangement Mode is a mode the program enters when a users wants to select a set and try to form a pattern with the tiles from that set. The main function within Tile Arrangement Mode is dragging tiles from the play area and grid to locations on the grid in order to form a pattern.

Version #	Date	Author	Description
0.1	9/19/05	Brian Navarro	Proposed system overview and functional requirements
		Nathan Mikeska	Proposed system nonfunctional requirements
		Richard Carney	Current System
0.2	9/20/05	Nathan Mikeska	Some formatting and minor changes
0.3	9/21/05	Brian Navarro	Cleaned up doc. Lots of formatting and minor changes
0.4	9/25/05	Nathan Mikeska	Scenarios, use cases, insertion and formatting, some consistency checking, glossary
		Brian Navarro	UI Mock-up, pseudo requirements
		Richard Carney	Use case diagrams
0.5	9/25/05	Neil Alfredson	Introduction, table of contents
		Brian Navarro	UI Mock-up fixes
1.0	9/25/05	Team	Reviewed by team and accepted as version 1.0.
1.0 – 1.3	9/25/05 12/07/05	Team	Thorough revision of entire document. Most sections updated
2.1	4/23/06	Neil Alfredson	Addendum