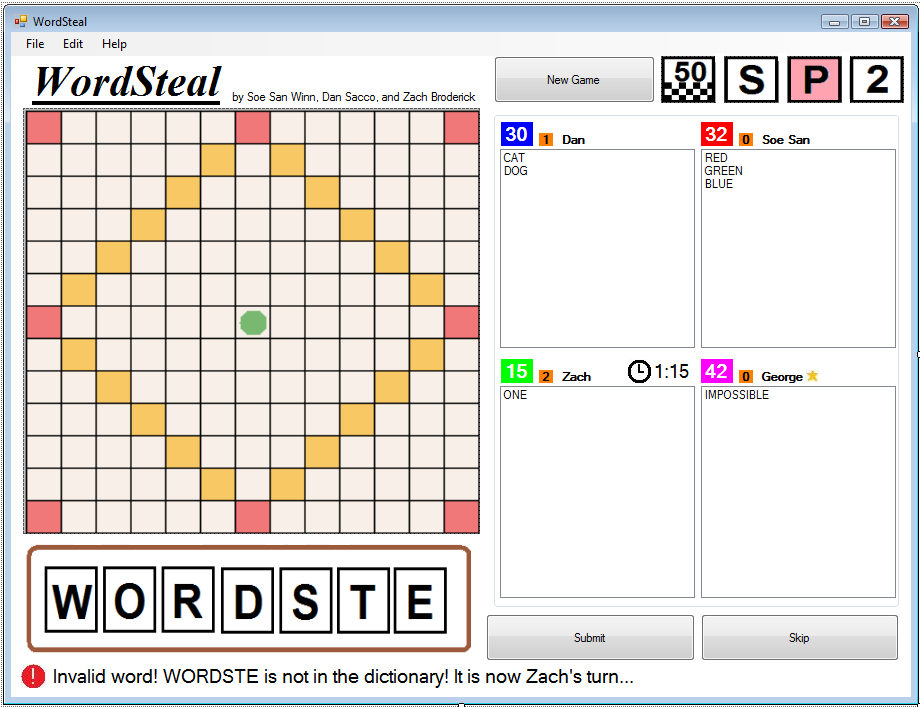
WordSteal

*The best word game you’ve never heard of.*



Team 1: Dan Sacco, Soe San Win, Zach BroderickRequirements Analysis Document (RAD)

# Introduction

## Purpose of the system

There are countless word games available to be played. One of the best ones is known as WordSteal (also known as Wordox. For the purpose of this assignment, I will always refer to the game as WordSteal).

* [Wordox Web site](http://www.angelfire.com/wizard/sandshrew777/wordox_info.html)
* [WordSteal](http://wordsteal.com/index.html) ([video](http://www.youtube.com/watch?v=coYLpNqO0H0))

You will learn more about this game during the course of this HW1 task. Indeed, your primary goal is to determine (a) the core functionality that must be provided by your system and (b) the numerous possible extensions. This document available today is minimal since the primary task of each team is to fill in the requisite details as required to complete the assignment.

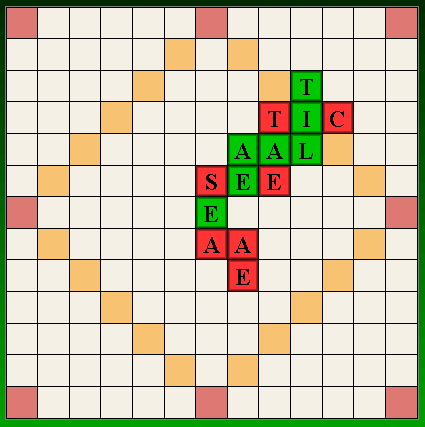


Figure : Sample WordOx layout

## Scope of the system

WordSteal is a game in which players take turns placing tiles on a two-dimensional board as shown in Figure 1. Each individual cell can contain exactly one tile. Up to four players may play a game. The specific rules of the game can be found [here](http://wordsteal.mymesis.com/altpages/wordstealrules.htm) or in the appendix. To keep the program as simple as possible, you will not be required to develop a networked version of the game. Each team will develop a stand-alone application which runs on a single desktop computer. Up to four players can play, each taking their turn as appropriate using the mouse and keyboard when it is their turn to play.

## Core System Functionalities

Your application must be able to:

* Allow up to four users to begin playing a game. Note that there is no ability to save a game in progress; the players must either complete the game or exit it.
* Allow users to select from a number of potential WordSteal variations prior to game play.
* Provide an external dictionary to validate words. This will enable the developers (i.e., YOU) to readily produce a non-English version of the game given a different dictionary.
* Provide a GUI that allows users to drag tiles from their “hand” onto the board, and move tiles around the board that they have placed.
* Provide an alternative keyboard-only mechanism for playing tiles and moving them on the board.
* Provide multi-level undo and redo for a player during his turn.
* Allow users to print a full and accurate recording of the game after it has played. This record will describe the full set of moves for each player, together with the final scores for each player (and running total for all players’ scores) and the final board state.

## Objectives and Success Criteria of the Project

The success of the application depends upon meeting the following core set of objectives:

* The design of a command mechanism to encapsulate all changes to the model as a series of commands to be executed on the model.
* The design of a multi-level undo/redo capability.
* The design of a GUI to enable the proper playing of a game.
* The use of automatic test cases to validate that 80% of the written code executes according to specification.
* The design of a mechanism to select at runtime from at least three different plays variations and the implementation of these variations during play.

## Definitions, Acronyms, and Abbreviations

Important terms and concepts are listed here. More will be added by the project teams.

*Model* A schematic description of a system that accounts for its known or inferred properties [http://www.answers.com]

*View* A visual representation of a model which might also enable a user to manipulate properties of the model

*Game* A game is an instance of WordSteal.

*Board* A 13x13 two-dimensional structure divided into rows and columns. For labeling purposes, the vertical columns are labeled ‘A’ through ‘M’ while rows are 1 through 13.

*Rack*A 7 cell structure used to hold tiles not yet placed on the board.

*Cell* A unique cell within a Board, identified by its Column (‘A’ to ‘M’) and Row (1 to 13). A cell is either empty or contains a single tile.

*Tile* A letter from ‘A’ to ‘Z’ that is placed within a Cell (see [distribution](http://wordsteal.mymesis.com/altpages/englishtiles.htm)).

*Variation* A rule designed to be in effect during game play. There are a [number of potential variations](http://www.angelfire.com/wizard/sandshrew777/faq-oxfest.html) that you must consider.

*Dictionary* Potential set of words that can be used (from 2–13 characters). I will provide one to the class in English. [Here](http://clubefl.gr/games/wordox/2.html) is a place to start.

*Player* Any user that interacts with the system.

*System* Any interactions performed by the application are considered to be performed by the system.

*Score* The number of points earned by each player against other players in the game.

*WordList* The list of words associated with each player for which the player is receiving points.

*Submit* The action of the player indicating that they have placed tiles in a configuration for which the player wishes to receive points.

*Turn* A period of time where a player is allowed to place tiles, submit a word, or skip the turn.

*Skip* The action of the player indicating that they do not wish to submit any tiles for their turn.

*Timer* The system process keeping track of the amount of time a player has left in the turn.

*Log* A record of all of the game’s major interactions.

*Undo* During a turn, if a player has placed tiles, they may reverse their actions.

*Redo* If a player has used the undo option, they may reverse that action.

*GUI* Graphical User Interface

## References

* <http://www.angelfire.com/wizard/sandshrew777/wordox_info.html>
* <http://wordsteal.mymesis.com/altpages/wordstealrules.htm>
* <http://wordsteal.mymesis.com/altpages/englishtiles.htm>
* <http://www.angelfire.com/wizard/sandshrew777/faq-oxfest.html>
* <http://clubefl.gr/games/wordox/2.html>

# Current System

No system is in place but you can visit and play WordSteal at an online site (for Free). I encourage students to do this so they can get a sense of the game behavior and logic.

# Proposed System

## Overview

This section provides a functional overview of the system. This will again be properly be divided into two parts

## Functional Requirements

## The game must be playable by 2-4 players simultaneously.

## Players must be able to enter their names, and choose specific rules for the variant of the game.

## Players must be able to start a new game when the program is loaded.

## Players must be able to start a new game during the progress of a game with confirmation that they want to cancel the old game.

## The score and the words each player possesses must be displayed at all times on a score board.

## The words each player possesses must be high-lighted with the player's color on the board.

## Players must be able to move tiles around on the rack, move the newly placed tiles on the board and move tiles from the rack to the board, and from the board back to rack.

## Players must be able to skip a turn.

## Players must be able to undo and redo their moves during their turn.

## Players must be able to save the log of the game.

## Nonfunctional Requirements

### Usability

* Game must be playable without mouse
* Each player must have unique color representation.
* Tiles on the board should reflect the PLAYER who owns them by being displayed in that PLAYER's color.
* “About” and “Help” information about the game must be able to be retrieved at any point during the game or the run of application.
* Players must be able to choose from a set of several different language dictionaries.
* SYSTEM will use a standard dictionary format so that third party dictionaries that exist in that format may be used.
* SYSTEM will provide status messages to the player to inform the player of the state of the game.
* SYSTEM must play a sound when player has 10 seconds or less during their turn. (Single beep)
* SYSTEM must play a sound when player runs out of time. (Alarm clock)
* SYSTEM must play a sound when player submits word that is not in dictionary. (Two note, low pitch sound)
* SYSTEM must play a sound when player submits a word that is in dictionary. (Ding ding)
* SYSTEM must play a sound when player steals opponent’s tile(s). (Cha ching!)
* SYSTEM must play a sound when player submits a word on a pink square and the board is cleared. (Ding ding ding)
* SYSTEM must play a sound when a player wins the game. (Triumphant horns)

### Reliability

### Components of the project code will be tested alongside the implementation phase to ensure that they are functional.

### Final, integrated project Code will be tested with EclEmma to ensure that greater than or equal to 80% of the integrated code is covered at run-time, and is functioning properly. The remaining 20% will be inspected through manual testing to ensure the highest chance of being quality code.

### Performance

### Drag and drop of the tiles must be smooth without graphical lagging.

### Validation of the words from the dictionary or keeping track of players’ words will take little or unnoticeable amount of time.

### Supportability

### The application must not be platform dependent, i.e., it should be able to run on any platform supporting JAVA.

### Implementation

* Project will be implemented in JAVA.
* All project graphical user interfaces will be created using a JAVA GUI editor.

## System Models

### Use case model

|  |  |
| --- | --- |
| **Name:** | StartNewGame |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | Application is running.  A Game is not currently in progress. |
| **Flow of**  **Events:** | 1. PLAYER initiates new game function  2. SYSTEM presents PLAYER with NewGameForm  3. PLAYER submits NewGameForm |
| **Exit**  **Conditions:** | Game is now in a new state. |

|  |  |
| --- | --- |
| **Name:** | StartNewWhileGameInProgress |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | A Game is currently in progress |
| **Flow of**  **Events:** | 1. PLAYER initiates new game function  2. SYSTEM confirms that the PLAYER wishes to continue.  3. SYSTEM presents PLAYER with NewGameForm  4. PLAYER submits NewGameForm |
| **Exit**  **Conditions:** | Game is now in a new state. |

|  |  |
| --- | --- |
| **Name:** | PlaceTileFromRack |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  There is at least one tile on the rack  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selects a tile from the rack and places it in a cell on the board.   2. SYSTEM refreshes to show modified rack and board on display. |
| **Exit**  **Conditions:** | The tile is now in the destination cell on the board and no longer in the rack. |

|  |  |
| --- | --- |
| **Name:** | MoveTileOnRack |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  There is at least one tile on the rack  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selects a tile from the rack and places it in another position on the rack.   2. SYSTEM refreshes the rack on display to show the new state. |
| **Exit**  **Conditions:** | The tile is now located at the destination indicated by the PLAYER and not in its previous location, and any other tiles have been shifted accordingly. |

|  |  |
| --- | --- |
| **Name:** | MoveTileOnBoard |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  The PLAYER has placed at least one tile on the board this turn.  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selects a tile from the board that he or she placed this turn and moves it to another value destination cell on the board.   2. SYSTEM updates the rack to show the new state on display. |
| **Exit**  **Conditions:** | The tile chosen is now in the destination cell indicated by the PLAYER and is no longer where it was previously. |

|  |  |
| --- | --- |
| **Name:** | MoveTileFromBoardToRack |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  There is at least one tile on the board that the PLAYER placed there this turn.  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selects a tile he or she previously placed on the board this turn and moves it to the rack.   2. SYSTEM updates the state of both rack and board on display. |
| **Exit**  **Conditions:** | The chosen tile is now in the indicated position on the rack and is no longer on the board. |

|  |  |
| --- | --- |
| **Name:** | PlaceIllegalTile |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | Extends PlaceTileFromRack, MoveTileOnRack, MoveTileOnBoard, and MoveTileFromBoardToRack |
| **Flow of**  **Events:** | 1. After selecting a valid tile, PLAYER attempts to move it to an invalid position (outside the board/rack, on top of another tile on the board).  2. SYSTEM denies the move, and plays the relevant sound. |
| **Exit**  **Conditions:** | The tile is returned to its original position |

|  |  |
| --- | --- |
| **Name:** | SubmitWord |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  PLAYER has placed at least two tiles on the board this turn  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selected the Submit option. |
| **Exit**  **Conditions:** | SYSTEM plays the relevant sound.  PLAYERs’ scores and wordlists are updated, and Timer is reset.  Tiles on the rack have been refilled to 7 if there are enough tiles remaining. If not, fill up the rack with the remaining tiles.  It is now the next PLAYER’s turn. The next Player is defined to be the next player in the list as entered in StartNewGame form. |

|  |  |
| --- | --- |
| **Name:** | SubmitInvalidWord |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | Extends SubmitWord |
| **Flow of**  **Events:** | 1. PLAYER places a word on the board that is invalid (not in dictionary, not contiguous, violates variation rule, or not in green square on first turn). |
| **Exit**  **Conditions:** | SYSTEM plays the relevant sound.  SYSTEM updates the status window on the main GUI with the appropriate message.  Tiles placed this turn have been returned to the rack. No points or words have been added to PLAYERs’ scores.  It is now the next PLAYER’s turn. The next Player is defined to be the next player in the list as entered in StartNewGame form. |

|  |  |
| --- | --- |
| **Name:** | SubmitOrangeCellWord |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | Extends SubmitWord  PLAYER has placed a tile on an orange cell |
| **Flow of**  **Events:** |  |
| **Exit**  **Conditions:** | Bonus points have been added to the PLAYER’s information.  Bonus points will be added to the PLAYER’s score if and only if the PLAYER plays a word on a Pink square. |

|  |  |
| --- | --- |
| **Name:** | SubmitPinkCellWord |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | Extends SubmitWord  PLAYER has placed a tile on a pink cell, and the game is not in “noPink” variation. |
| **Flow of**  **Events:** |  |
| **Exit**  **Conditions:** | SYSTEM plays the relevant sound.  SYSTEM updates the status window on the main GUI with the appropriate message.  The board has been cleared of tiles.  The bonus points added to the PLAYER’s information has been added to the PLAYER’s score.  Bonus points of ALL OTHER players are erased. |

|  |  |
| --- | --- |
| **Name:** | SkipTurn |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  Game is not finished |
| **Flow of**  **Events:** | 1. PLAYER selects the Skip Turn option.  2. SYSTEM refreshes the board and rack as necessary. |
| **Exit**  **Conditions:** | All tiles placed on the board this turn have been returned to the rack.  It is now the next PLAYER’s turn. The next Player is defined to be the next player in the list as entered in StartNewGame form. |

|  |  |
| --- | --- |
| **Name:** | ReachedMaxScore |
| **Actor:** | SYSTEM |
| **Entry**  **Conditions:** | A PLAYER’s score has reached or exceeded the maximum score |
| **Flow of**  **Events:** | 1. SYSTEM presents the WinningGameFrame and plays the relevant sound. |
| **Exit**  **Conditions:** | The game is now in the finished state.  SYSTEM plays the relevant sound.  SYSTEM updates the status window on the main GUI with the appropriate message. |

|  |  |
| --- | --- |
| **Name:** | ExpireTime |
| **Actor:** | SYSTEM |
| **Entry**  **Conditions:** | The PLAYER whose turn it is has exceeded the time allotted him or her. |
| **Flow of**  **Events:** | 1. TIMER expires.  2. SYSTEM proceeds with skip turn behavior. |
| **Exit**  **Conditions:** | SYSTEM plays the relevant sound.  SYSTEM updates the status window on the main GUI with the appropriate message.  All tiles placed on the board this turn have been returned to the rack.  It is now the next PLAYER’s turn. The next Player is defined to be the next player in the list as entered in StartNewGame form. |

|  |  |
| --- | --- |
| **Name:** | SkipAllPlayers |
| **Actor:** | SYSTEM |
| **Entry**  **Conditions:** | All PLAYERs have consecutively skipped their turn |
| **Flow of**  **Events:** |  |
| **Exit**  **Conditions:** | The rack is now filled with a new set of 7 tiles if there are enough tiles remaining. If not, fill in with as many tiles as there is left.  SYSTEM updates the status window on the main GUI with the appropriate message. |

|  |  |
| --- | --- |
| **Name:** | ViewHelp |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | The application is running |
| **Flow of**  **Events:** | 1. PLAYER selects the View Help option  2. SYSTEM stops the clock if there is a game in progress.  3. SYSTEM presents HelpFrame. |
| **Exit**  **Conditions:** |  |

|  |  |
| --- | --- |
| **Name:** | ViewAbout |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | The application is running |
| **Flow of**  **Events:** | 1. PLAYER selects the View About option  2. SYSTEM stops the clock if the game is in progress.  3. SYSTEM presents AboutFrame. |
| **Exit**  **Conditions:** |  |

|  |  |
| --- | --- |
| **Name:** | ExitApplication |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | The application is running |
| **Flow of**  **Events:** | 1. PLAYER selects the Exit option  2. SYSTEM asks for confirmation with confirmation form.  3. PLAYER submits exit confirmation. |
| **Exit**  **Conditions:** | The application has been terminated. |

|  |  |
| --- | --- |
| **Name:** | SaveGameLog |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | The game is in a finished state |
| **Flow of**  **Events:** | 1. PLAYER selects the Save Log… option  2. SYSTEM presents standard OS save file dialog  3. PLAYER submits form |
| **Exit**  **Conditions:** | Log of game now saved |

|  |  |
| --- | --- |
| **Name:** | Undo |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  PLAYER has placed at least one tile this turn |
| **Flow of**  **Events:** | 1. PLAYER selects the undo option  2. SYSTEM refreshes the rack and board to match the state of the game. |
| **Exit**  **Conditions:** | The last tile moved is now in its original position. |

|  |  |
| --- | --- |
| **Name:** | Redo |
| **Actor:** | PLAYER |
| **Entry**  **Conditions:** | It is the initiating PLAYER’s turn  The PLAYER’s last action was an Undo |
| **Flow of**  **Events:** | 1. PLAYER selects the redo option.  2. SYSTEM refreshes the rack and board to match the state of the game. |
| **Exit**  **Conditions:** | The tile moved by the previous Undo action is returns to its originally intended destination. |

### Object model

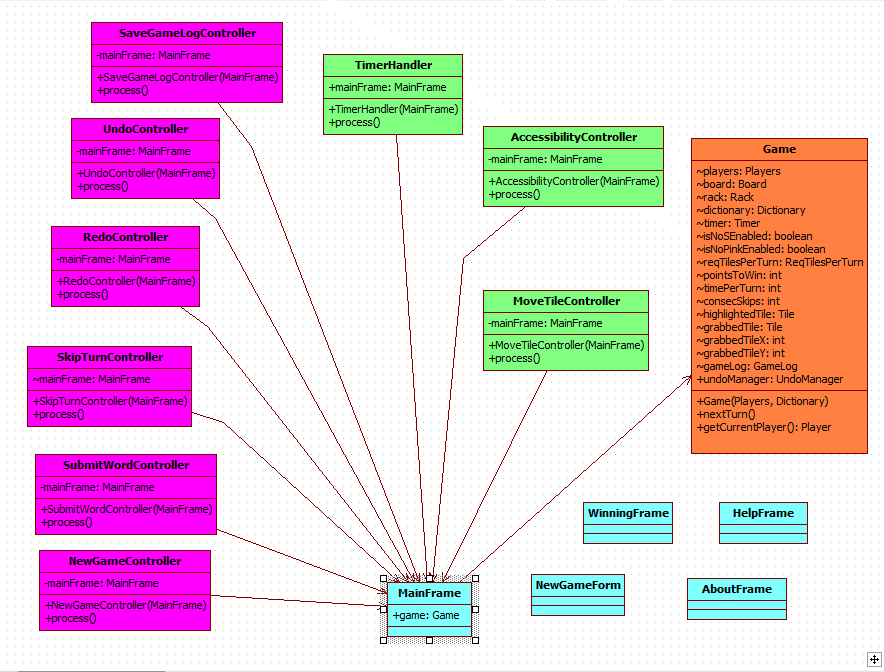


Figure .1 - Object Model – Boundaries/Controllers

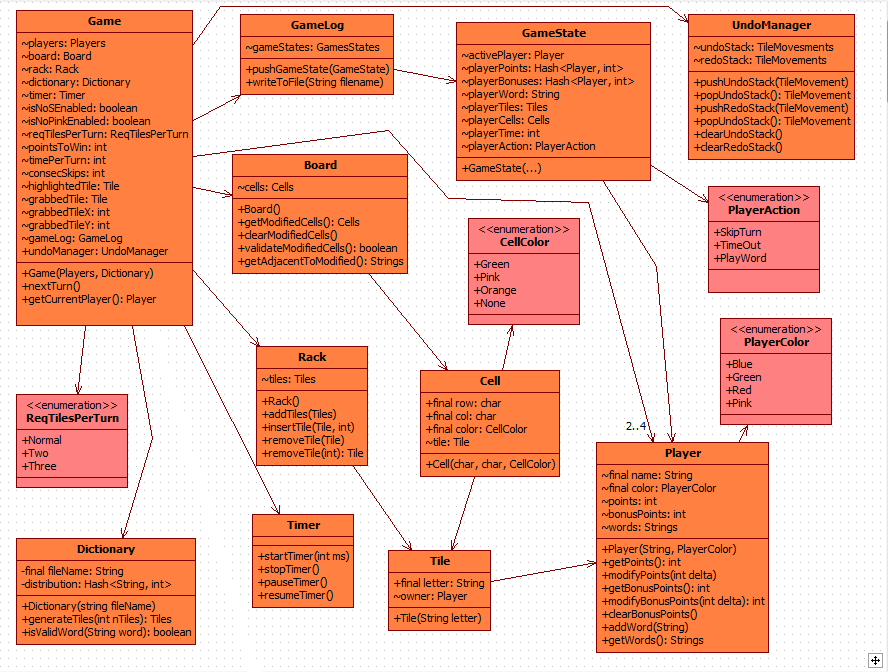


Figure .2 - Object Model – Entities (except Undo)

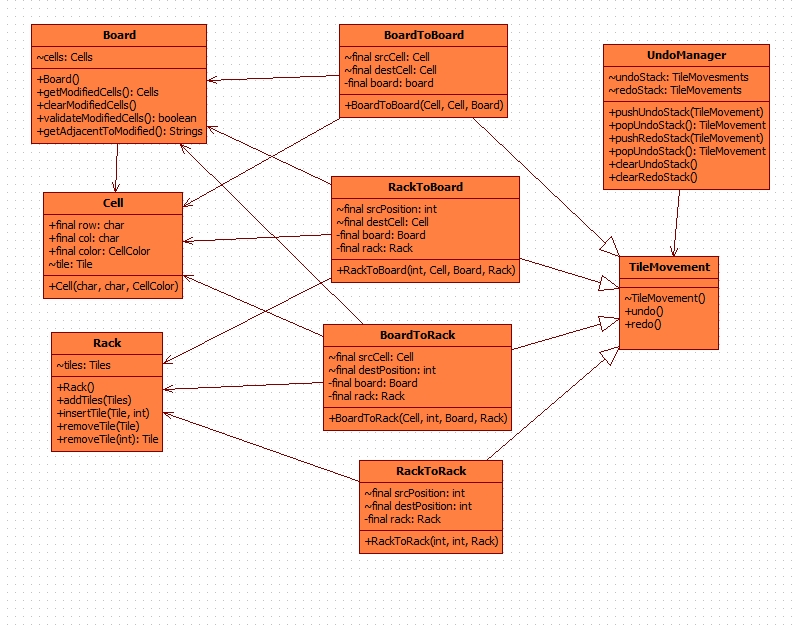


Figure .3 - Object Model – Undo Entities

### User Interface – Navigational Paths and Screen Mock-ups

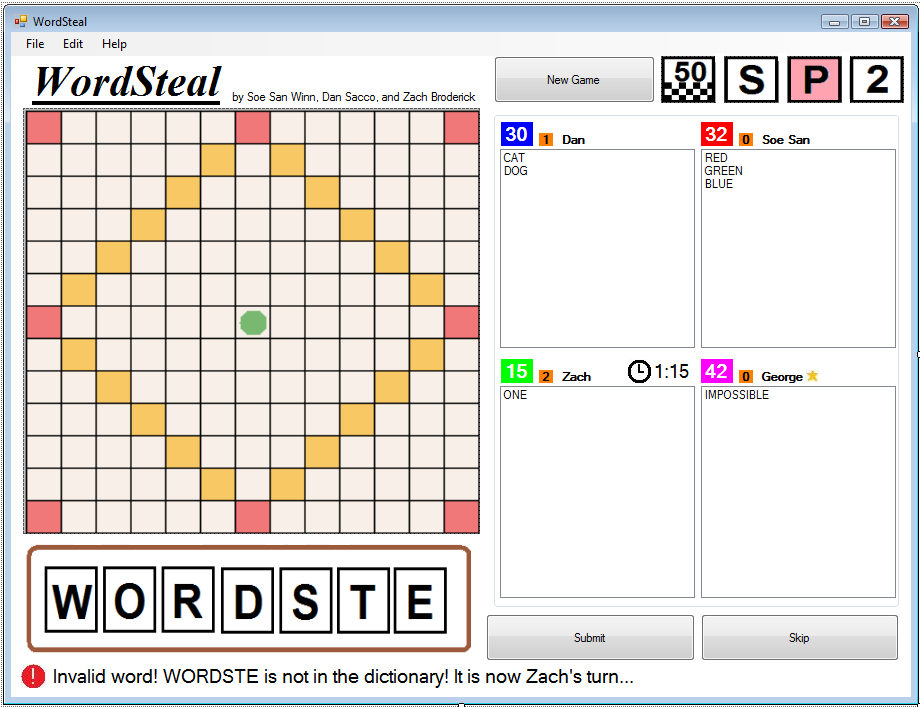


Figure 3 - Main Form

Menu Bar:

* File
  + New Game... (Ctrl+N)
  + Save Log... (Ctrl+S)
  + Exit... (Ctrl+Q)
* Edit
  + Undo (Ctrl+Z)
  + Redo (Ctrl+Y)
* Help
  + About (F1)
  + Instructions (F2)

Board/Rack:

* PLAYERs select tiles from either the board or the rack by holding down the left mouse button. Tiles are dragged in the standard drag-and-drop fashion, smoothly animated, as the PLAYER drags the mouse with the left mouse button down. A tile is release when the PLAYER releases the left mouse button.
* Rack has room for exactly 7 tiles, if PLAYER wants to re-arrange them, simply drags and drops tile in between two other tiles and they are moved to reflect new ordering.
* Tiles on board are colored to reflect owning PLAYER's color.

Scoreboxes:

* Each player has a box containing the WordList he or she has acquired through the game.
* Above the WordList to the left is a box containing the player’s score and colored to reflect the player’s color.
* To the right of the score box is a smaller orange box numbered to reflect the amount of bonus points the player currently has that have not yet been added to their score.
* To the right of the bonus points box is the player’s name as specified in the NewGameForm.
* If a player is currently in the lead, a gold star icon appears to the right of his or her name. If there is a tie, all tied players have a gold star next to their name.
* The player whose turn it is designated by a clock icon and the current time remaining in the turn displayed above his or her WordList and right justified.

Status text (bottom of frame):

* Displays various messages when a user hits Submit:
  + <WORD> is not in the dictionary!
  + You must place tiles contiguously horizontally or vertically!
  + You must place at least 2 tiles!
  + No\_S: You cannot simply append S to a word!
  + <#> Tiles: You must play exactly <#> tiles!
  + It is now <PLAYER's> turn.
  + Pink tile hit! Board is cleared!
  + Times up!
  + <PLAYER> wins!
  + All PLAYERs have skipped. Tiles reset!

Buttons:

* Submit corresponds to Submit action in the use cases, as does Skip and New Game.

Game Icons:

* Display information reflecting the game settings specified in the new game form. The first icon, to the right of the New Game button, displays how many points are needed to win the game. The next 3 icons represent the No S, No Pink, and 2/3 Tile Rule respectively and whether or not they are enabled (represented by different icons).

Keyboard-only play:

* Alt accesses the menu bar in standard OS fashion.
* Tab moves between the New Game, Submit and Skip buttons in standard OS fashion. Enter presses these buttons when selected.
* Pressing any arrow key will cause the UI to highlight the first tile in the rack in the player’s color. Pressing the right and left arrow keys will move among the tiles on the rack, highlighting the currently “selected” on in the player’s color.
* Pressing the up arrow key will select the tile on the board, if there is one, that is nearest to the tile previously selected on the rack and has been placed by the player this turn. If there are none, the previously selected tile remains highlighted. At no time can the player select an empty cell—tiles only.
* If the player has selected an tile on the board and presses any of the arrow keys, the system will highlight whichever tile is deemed closest on the board that that player has previously played this turn. If a tile on the rack is closest, that tile is highlighted instead.
* A player may press the space bar to “grab” a currently selected tile (no visible response).
* If a player presses the arrow keys while a tile is “grabbed”, that tile is moved to the nearest cell in the direction of the arrow key, either on the board or the rack. This only happens graphically, however—actual game state is not changed, but the grabbed tile is rendered in its new position and not in its previous position. If it is in a cell occupied by another tile, it is drawn over that tile.
* A player may “ungrab” a grabbed tile at any time by pressing the ESC key. This will return the grabbed tile to its original position.
* Pressing the space bar while a tile is grabbed will attempt to place that tile in its current graphical location. If that location is illegal (occupied by another tile on the board), the tile is returned to its original position. If the cell is on the rack and occupied by another tile, the tiles are shifted accordingly. The tile is no longer in the grabbed state.

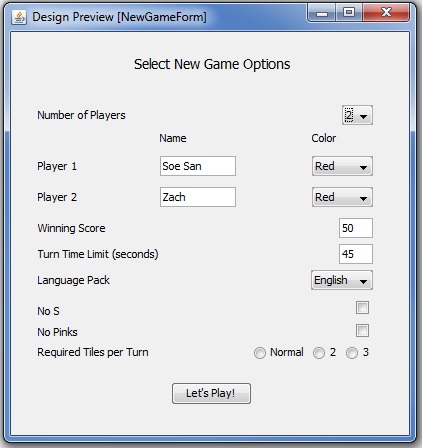


Figure 4 - New Game Form

Number of players:

* PLAYERs select the number of PLAYERs that will be in the game.  The available options are 2, 3, and 4.
* After the number of PLAYERs is selected, the appropriate number of PLAYER information fields appears. The remaining text in the window is shifted downward to provide room for the PLAYER fields.

Player options:

* The names of each of the PLAYERs are entered into the PLAYER name fields.
* The color for each PLAYER is selected from each of the drop down menus.  The available options are red, orange, blue, green, yellow, and purple.
* PLAYERs must choose unique names and unique colors.

Game preferences:

* PLAYERs select the language pack for the game from the drop down menu.
* PLAYERs select the minimum number of tiles by using the radio buttons.  These buttons are mutually exclusive.

Game modifications:

* PLAYERs use check boxes to select what game variations they want to use.  The options are “No S”, “No Pinks,” and “Two/Three Tile Rule.”
* The two/three tile variation is enabled by selecting either the 2 or 3 option. Normal allows the players to place any number of tiles per turn.

“Let’s Play” Button:

* When selected the “Let’s Play” button closes the form and starts a game with the entered information.
* If the “Let’s Play” button is pressed and there are PLAYERs with similar names or similar colors then an error box will appear and the form will not be submitted.

Keyboard-only play:

* Tab moves between all editable fields in the form.
* Selection of fields is indicated by a highlighted boarder around their respective UI elements.
* Enter changes the state of check boxes and radio buttons when selected.
* Enter opens drop down menus when selected.
* Enter activates button when selected.
* Arrow keys move between options in a drop down box and enter selects one of the options.

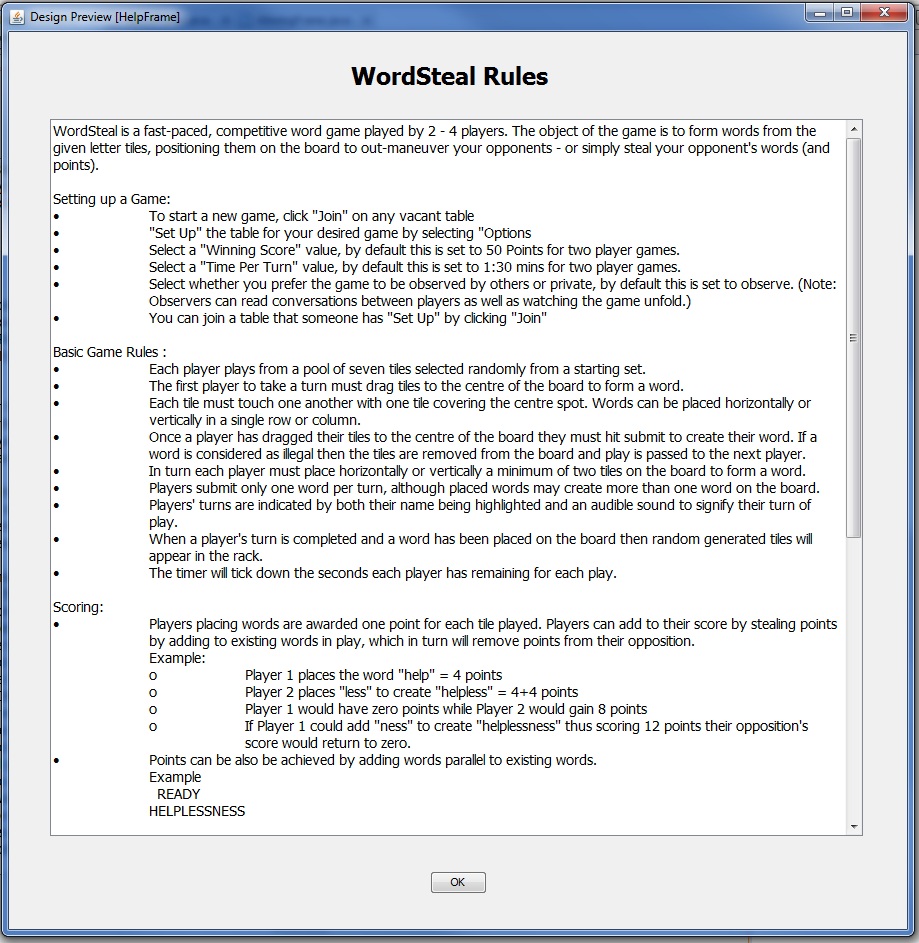


Figure 5 - Help Frame

Rules Window:

* All of the game rules are outlined in a non-editable field in the form.
* The scroll bar can be used in standard fashion to view all of the rules information stored in the field.

“OK” Button:

* When selected, this button closes the rules form.  The game returns to the state it was in when the form was opened.

Keyboard-only play:

* When the form is opened the “OK” button is automatically selected.
* Pressing enter key selects the “OK” button and thus closes the form.

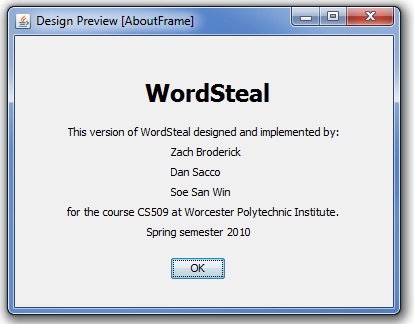


Figure 6 - About Frame

“OK” Button:

* When selected, this button closes the rules form.  The game returns to the state it was in when the form was opened.

Keyboard-only play:

* When the form is opened the “OK” button is automatically selected.
* Pressing enter key selects the “OK” button and thus closes the form.



Figure 7 - Winning Frame

“OK” Button:

* When selected, this button closes the form.

Keyboard-only play:

* When the form is opened the “OK” button is automatically selected.
* Pressing enter key selects the “OK” button and thus closes the form.

# Change History

As this document changes, it will be made available with Microsoft Track Changes turned “on” so you will be able to see all the updates that have occurred.

|  |  |
| --- | --- |
| **Version** | **Description** |
| 20100128.1421 | Initial Version released to the class |
|  |  |

# References

Java Printing Resources [http://articles.techrepublic.com.com/5100-10878\_11-6150617.html]

# Appendix: WordSteal Rules (<http://wordsteal.mymesis.com/altpages/wordstealrules.htm>)

Wordox is a fast-paced, competitive word game played by 2 - 4 players. The object of the game is to form words from the given letter tiles, positioning them on the board to out-manoeuvre your opponents - or simply steal your opponent's words (and points).

## Setting up a Game

* To start a new game, click "Join" on any vacant table
* "Set Up" the table for your desired game by selecting "Options
* Select a "Winning Score" value, by default this is set to 50 Points for two player games.
* Select a "Time per Turn" value, by default this is set to 1:30 mins for two player games.
* Select whether you prefer the game to be observed by others or private, by default this is set to observe. (Note: Observers can read conversations between players as well as watching the game unfold.)
* You can join a table that someone has "Set Up" by clicking "Join"

## Basic Game Rules

* Each player plays from a pool of seven tiles selected randomly [from a starting set](http://wordsteal.mymesis.com/altpages/englishtiles.htm).
* The first player to take a turn must drag tiles to the centre of the board to form a word.
* Each tile must touch one another with one tile covering the centre spot. Words can be placed horizontally or vertically in a single row or column.
* Once a player has dragged their tiles to the centre of the board they must hit submit to create their word. If a word is considered as illegal then the tiles are removed from the board and play is passed to the next player.
* In turn each player must place horizontally or vertically a minimum of two tiles on the board to form a word.
* Players submit only one word per turn, although placed words may create more than one word on the board.
* Players' turns are indicated by both their name being highlighted and an audible sound to signify their turn of play.
* When a player's turn is completed and a word has been placed on the board then random generated tiles will appear in the rack.
* The timer will tick down the seconds each player has remaining for each play.

## Scoring

* Players placing words are awarded one point for each tile played. Players can add to their score by stealing points by adding to existing words in play, which in turn will remove points from their opposition.

*Example*

* + Player 1 places the word "help" = 4 points
  + Player 2 places "less" to create "helpless" = 4+4 points
  + Player 1 would have zero points while Player 2 would gain 8 points
  + If Player 1 could add "ness" to create "helplessness" thus scoring 12 points their opposition's score would return to zero.
* Points can be also be achieved by adding words parallel to existing words.

*Example*

READY

HELPLESSNESS

* + Player 2 would lose 3 points while player 1 would gain 8 points. 5 points for the word they placed "ready" and 3 for the additional 2 tile words they created, "re","es" & "as"
* Other points can be gained by placing tiles over orange coloured squares. These additional points are indicated with an icon below players' names and can only be won if you place a tile over a pink coloured square, which clears the board. If someone else clears the board then all bonus points you gained by covering orange squares are lost.

## Miscellaneous

One situation might cause confusion, so is worth clarifying:

* It is legal (and sometimes beneficial) to deliberately skip your turn even if you have a valid move available. Doing this is not considered bad sportsmanship. *Rationale: A 'No Skip' rule would not work, because it would be totally unenforcable, and would likely lead to arguments about whether someone skipping was deliberate or not.*

## Additional Rules

Originally, the game was played without any extra rules, but several variations have developed over time. These additional rules only take effect if the captain of the game selects them in the game Options before the game is started, or for one variation, announces it before the game starts. The captain can choose any combination of one or more of these rules. Players that do not agree with the rules the captain chooses or announces must quit either before the game starts or before they play their first move (ratings are not deducted). If the captain does not choose or mention a rule, it should be assumed it is not in effect.

* **No S**: Also called NS, this is a popular variation of play. No player may extend a word (not even their own word) by adding a single S to the end of the word. Note that it is OK to extend 'CATCH' to 'CATCHES', or 'TOP' to 'STOP', or 'PIN' to 'SPINS', but not OK to extend 'SPIN' to 'SPINS'. Extending 'THESE' to 'THESES' is also disallowed even though the two words are unrelated. However, extending single letters such as 'A' to 'AS' is OK. Any player who breaks this rule automatically loses the turn. Is is enforced by the server.
* **No Pinks**: Also called NP. It is not legal to clear the board. The pink clearing squares have a line in them to indicate they are not valid plays. Any player who plays a tile on a pink square automatically loses the turn. This is enforced by the server.
* **No S, No Pinks**: This combination is very popular. Some players play nothing else.
* **Two Tile/Three Tile**: A few players enjoy this a lot. The captain must announce it ahead of time, and all players must agree to the rule before the game starts. A player must play exactly 2 (or 3) of the 7 tiles on each move (or else skip). Any player who plays more tiles automatically loses the game, or some other penalty announced by the captain before the game begins (but this is currently not enforced by the server).

http://counter.dreamhost.com/cgi-bin/Count.cgi?df=wordstealrules1.dat&pad=F&ft=0&dd=E&istrip=Thttp://counter.dreamhost.com/cgi-bin/Count.cgi?df=wordstealrules1.dat&pad=F&ft=0&dd=E&istrip=T&display=date