Java Game Suite Phase 3

Revision 9

CMSC 495 6382

October 3, 2021

Group Charlie

Sherry Funches, Oyewole Sanusi, Janee’ Jones, Wayne Mack, Jeffrey McGlinn

## **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date** | **Description** | **Name** |
| 1 | 9/22 | Creation of Doc and Outline | Oyewole |
| 2 | 9/23 | Overview and Milestone | Oyewole |
| 3 | 9/23 | Phase 3 created/ Modified | Oyewole |
| 4 | 10/3 | Updated Phase 4 Goals | Sherry |
| 5 | 10/3 | Phase 3 Documenation Snakes | Oyewole |
| 6 | 10/3 | Word Search and Main Menu Phase 3 Writeup | Sherry |
| 7 | 10/3 | Slider Puzzle Phase 3 Description | Jeff |
| 8 | 10/3 | Phase 3 Sudoku | Janee’ |
| 9 | 10/3 | Combine and format files. Add Table of Contents | Sherry |
|  |  |  |  |

**Table of Contents**

I. Overview………………………………………………………………………………………..4

1.1 Overview of Project…………………………………………………………………………...4

II. Milestones………………………………………………………………………….…………..4

2.1 Projected Team Milestones……………………………………………………………………4

2.2 Current Status on Phase 3 …………………………………………………………………….5

A. Maze………………………………………………………………………………...….6

B. Snakes…………………………………………………………………………………..6

C. Sudoku …………………………………………………………………………………7

D. Word Search……………………………………………………………………………9

E. Slider Puzzle…………………………………………………………………………..11

F. Main Menu…...……………………..…………………………………………………14

## **I. Overview**

**1.1 Overview of project**

The Java Game Suite program is a simple GUI-based application that provides entertainment to users by allowing them to play several simple single player computer games. The user will be presented with an interface that indicates the available games: a maze, snakes, Sudoku, word search, and a slider puzzle. The user will be able to select a game from the suite and launch it from that interface.

## **II. Milestones**

* 1. **Milestone**

The following table shows our completed and pending goals. The timeline of certain goals have changed based on changes made to the implementation process.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Projected Date of  Accomplished Goal | Name of the Goal | Was the Goal  Accomplished |
| Phase 1 | 9/13 | Design what the system does and how the different pieces fit together. | Yes |
| 9/20 | Start GUI coding | Yes |
| 9/20 | Start other required classes coding | Yes |
| Phase 2 | 9/26 | Continue GUI and other required classes coding | Yes |
| 9/26 | Finish and share prototypes of working games with group | Yes |
| 9/26 | Start Testing | Yes |
| Phase 3 | 10/1 | **Integrate all games and main menu into package** | Yes |
| 10/1 | **Testing** | Yes |
| 10/3 | **Revise code and implement additional features** | Yes |
| Phase 4 | Final Sprint and Delivery | | |
| 10/6 | Testing | Pending |
| 10/6 | Finishing Touches | Pending |
| 10/6 | Update all documents with any changes | Pending |
| 10/8 | Group Revision | Pending |

**Current Status on Phases**

**Phase 3:**

**A. Maze**

**B. Snakes**

Phase 3 is still pending. However, I had selected an ideal speed for and created an out of bounds feature that will trigger an automatic game over when crossed by players. Refer to figure 1.2.

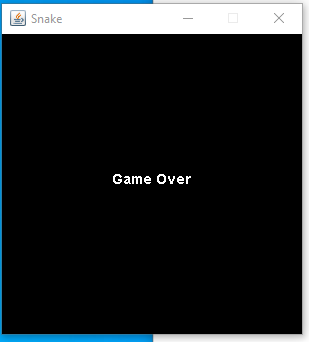


Figure1.2

The Grid size from phase 2 has also been coded into the snake game. The goal for phase 4 is to polish my code, fix any remaining bugs and test all games with my team members.

**C. Sudoku**

For Phase III, the Sudoku puzzle was tested throughout the week for continuity purposes. Mentioned in phase II, I was going to attempt to incorporate a timer. As I was doing this the code was having issues, so I decided not to continue with adding the timer, being that I had a working project/prototype to begin with. The code for the sudoku puzzle went through testing well and did not produce any errors. Once the user chooses Sudoku from the main menu, the initial game will be displayed to the user. The user will still have the option to proceed with that game or choose a new game by clicking “New” to be shown a different board. As the user continues to enter the numbers in the appropriate box, the “Check” option can still be used while playing the game so the user can see the progression of the game. To choose a number, the user simply selects any number they would like to start with. Once the user finishes inputting the numbers in the boxes the numbers will grey out and can no longer be clicked. This acknowledges to the user that the number chosen is completed and to proceed to the next number. When the user has completed the puzzle, they will be able to use the “Check” option to make sure that the numbers were entered correctly. Lastly, once the user has completed the game, they have the option to “Exit” game to go back to Main Menu and select another game or “New” game to start a new game.

Graphical user interface

Description automatically generated

Figure 1: Once application is running the Main Menu is displayed, user chooses Sudoku to play.

Calendar

Description automatically generated

Figure 2: User is presented initial game once Sudoku is chosen from the main menu.

A screenshot of a computer

Description automatically generated with low confidence

Figure 3: User will select a number i.e. “1”. All the locations where a 1 can go will be displayed in blue. The user will be able to click on the box that they want the 1 to go. The blue display is annotated by the “Help on” check box. User will be able to turn this on or off depending on if they want the help. User will proceed with each number with the same concept.

Table

Description automatically generated with medium confidence

Figure 4: User checks progress, Green annotates that number is entered in correct block, Red annotates that number is incorrectly entered in block.

Table

Description automatically generated

Figure 5: User has completed the game. User can select New Game to play again (new game board generated) or Exit Game to be directed to the main menu.

**D. Word search**

I successfully completed the phase 3 goals for the word search program, which was to add any additional features and perform initial testing and debugging. The main feature that I added before testing was to enable the user’s highlight to appear dynamically as they dragged the cursor over the letters rather than only appearing after they released the mouse.

I performed two types of initial testing and debugging: using the test plan and free play by myself and other users. First I successfully passed the tests in the test plan, which I will update in the final documentation after the team also tests the game. The program displayed the available puzzles, the test puzzle loaded and displayed as expected, and the hidden words were displayed on the list panel. A strikethrough successfully appeared or was removed through each word on the list when clicked. The user was able to drag their cursor over the letters in the grid to highlight words. Clicking submit presented the expected unfound words to appear in red and the correct number of correctly identified words to be displayed as seen in Figure A. The program rejected invalid highlights. The undo button removed the last highlight and the clear button cleared the entire board.

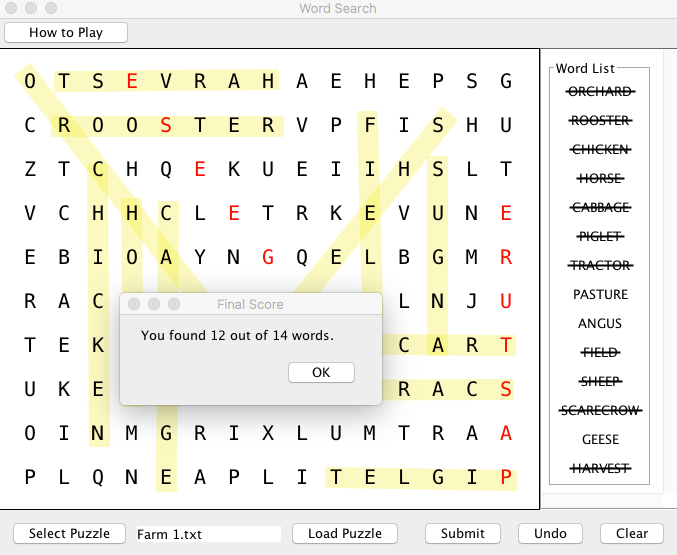


Figure A. Actual display of unfound words and appropriate score.

However, I did not just rely on the test plan and unexpected bugs were caught while I was programming and playing the game. For example, I noticed that the red letters from the unfound words remained when clearing a puzzle in order to replay it. I will update this in the test plan. I also fixed an issue in which the user has to drag the JFrame after loading the first puzzle in order to see it.

The second round of testing involved observing and informally interviewing a couple of test users. This was very useful because those outside of development process have different perspectives and are not influenced by their own knowledge of how the game functions. Through this process I discovered several additional features that I needed to add or think about including in the future. First, I discovered that a user could inadvertently clear the board with the clear button, so I implemented a dialog box that checks if the user wants to clear everything (Figure B). Secondly, I discovered that while buttons were labeled in what I thought was an intuitive manner, users might prefer an option to briefly read the game’s instructions and they may not want to refer to the user guide. I thus implemented a button and dialog box to display short instructions on the game’s functionality seen in Figure C. Third, I discovered that not all users have the dexterity to draw a single stroke across the center of the words. This program was intended for the general user so I did not make any changes. However, I may want to consider a more accessible version of the program in the future. Doing this would change the underlying logic of how I detect words. Lastly, I discovered that users may try to keep highlighting and submitting puzzles even after the solution was displayed. This is not necessarily a problem, but not behavior that I anticipated and I changed the code to prevent this from happening.

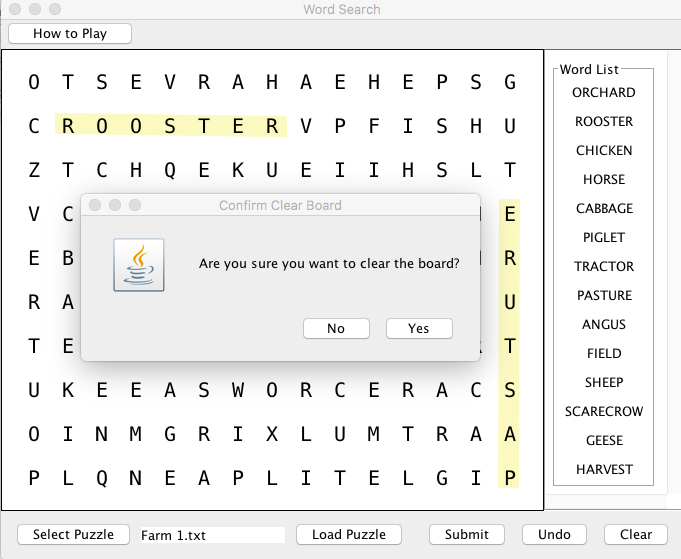


Figure B. Clear board confirmation dialog box.

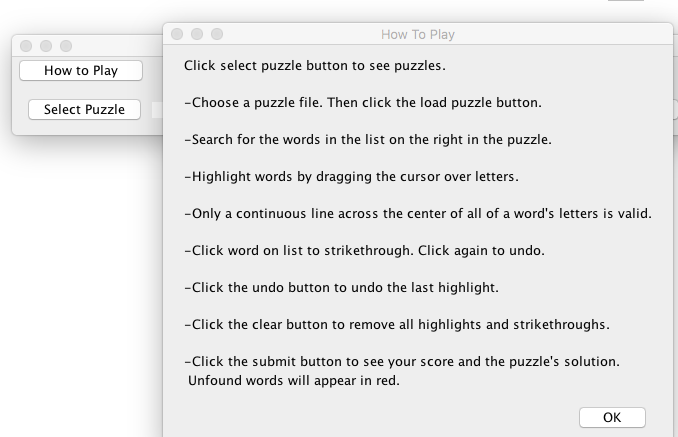


Figure C: Image of game play instructions.

**E. Slider Puzzle**

The goal for phase 3 was to integrate the slider puzzle into the main menu and start testing this portion of the Java Game Suite. After loading the main menu of the program, I selected the slider puzzle option. This successfully displayed the slider puzzle GUI. This is shown in the screen capture below:

Graphical user interface, website

Description automatically generated

Next, I tested to make sure that the user can load a new puzzle. Once a new puzzle Jbutton was selected, the program loaded the correct image files:

Background pattern

Description automatically generated

After the puzzle is rearranged correctly, a message is displayed informing the user that the puzzle has been completed:

Fireworks in the sky

Description automatically generated with medium confidence

The next thing that needed to be tested is the program’s error handling. If an image file is not found, a message was successfully displayed informing the user that an image file was not found:

A picture containing text

Description automatically generatedGraphical user interface, text, application, email, website

Description automatically generated

Lastly, I tested to make sure that the program does not exit the main menu once the slider puzzle is closed. This ensures that multiple games can be launched at once.

**F. Main Menu/ Game Integration**

Combining all programs and testing that they could all launch from the main menu was another goal for phase three that was successfully accomplished. Initially, when the individual programs were combined, different team members were either not able to launch all the games or received compilation errors. The solution was to place the files for each game into separate packages. Another issue that was discovered and resolved was that closing the window on everyone’s individual programs closed the entire program, which was not the behavior specified in the project’s requirements. In Phase 4 we will continue testing the main and individual and package the games in an executable file.

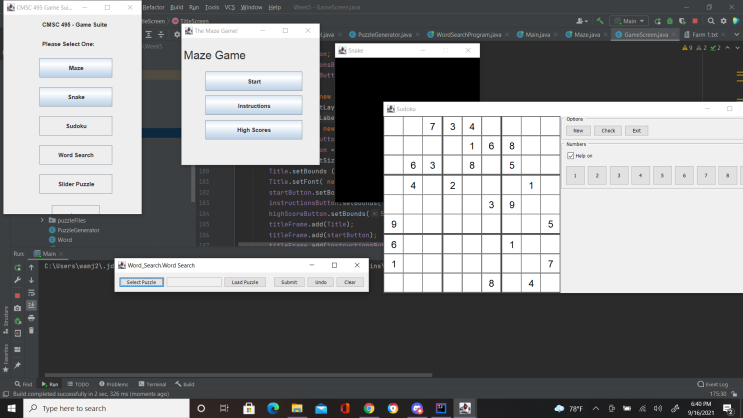


Figure 1. Image of games launched from main menu.