**Wandering in the Woods Game**

Table of Contents

[1. Introduction 3](#_Toc121746977)

[1.1. Purpose 3](#_Toc121746978)

[1.2. Wandering in the woods game 3](#_Toc121746979)

[2. Process Model 3](#_Toc121746980)

[3. Use Cases 4](#_Toc121746981)

[4. UML Model 6](#_Toc121746982)

[5. Personas 10](#_Toc121746983)

[6. GUI Mock-up 10](#_Toc121746984)

[7. Testing Strategy 14](#_Toc121746985)

[7.1. Unit Testing 14](#_Toc121746993)

[7.2. Integration Testing 14](#_Toc121746995)

[7.3. Usability test 14](#_Toc121746996)

[8. Appendix – User manual 15](#_Toc121746997)

[References 20](#_Toc121746998)

# Introduction

## Purpose

This document describes the software design of the wandering in the woods game. This game or simulation consists of three stages they are K-2, 3-5, and 6-8, the complexity increases for each stage. The main objective of this game is to help the students to learn about math concepts, computation, computer science, and computational thinking. This document is intended to provide a complete description of how the system is expected to perform. The developers or anyone interested in developing online games will find this document useful.

## Wandering in the woods game

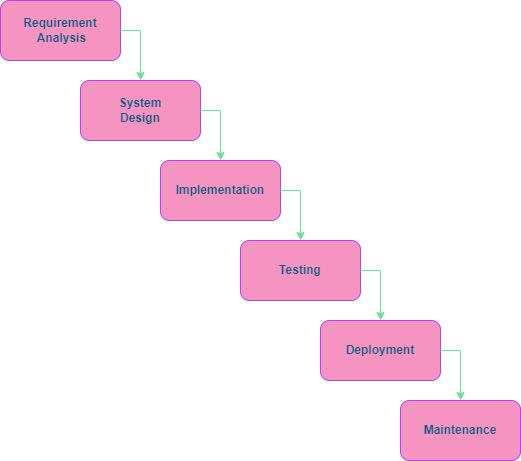
A pair of students participating in front of the same screen while using the simulation is required. In addition, tests to determine how well the students are handling the challenges, the simulation should contain audio suggestions and prompts.In this game, the rectangular grids are used to represent the woods. The main concept of this game is the people are lost in the woods and the people cannot see or hear one another until they are in the same grid cell because the woods are dense. The grids are always square, there are always two players, and they begin at the diagonally opposite corners of the grid in stage K-2. It is a simple version of the game. Each move is tracked using a counter for each participant as they move about at random. Cartoon figures are pacing around the woods as music is playing. The results from the wandering are displayed and audibly announced, when the people in the same grid cell, along with the cheerful graphical display. Following that, the students can restart the game when it has been reset.

The students can determine the size of the grid for stage 3-5 and the shape of the grid might be rectangular. Students can place as many as four persons on their grid and there might be two, three, or four of them. It is possible to play and repeat the game several times after the statistics are displayed. It is a slightly more complicated version of the game. If three or four people are playing, two of them will move together if they come across one another until they come across the third person. If the third person is found then the three players move together until the fourth person is discovered.

When the students are in stage 6-8 then they have full control over stage 3-5, they will challenge for conducting experiments to find out how the typical run changes depending on the size and shape of the grids. It is a more complex stage in this simulation game. To reduce the time it takes to meet up, they will be able to compare several wandering procedures and determine which one is best. Swing and AWT are used to develop window-based Graphical user interface (GUI) in java.

# Process Model

The software development team will use the waterfall model as shown in the figure below. The reason behind choosing this model is, the waterfall model is a sequential development method that moves like a waterfall through all the phases, with each phase finishing up entirely before the next phase is start. The phases of this model are the designing phase, analysis phase, implementation phase, testing phase, deployment phase, and testing phase (Dave Swersky, 2022).



**Figure 1: Process flow Diagram:**

# Use Cases

* 1. **Use Case 1: For Grades K-2**

**Primary Actor:** Student in Grades K-2

**Preconditions:** This stage is created for students in Grades K-2, and this stage is created for two players.

**Description:** The grids are always square, there are always two players, and they begin at the diagonally opposite corners of the grid in stage K-2. It is a simple version of the game. Each move should track using a counter for each participant as move about at random. Cartoon figures are pacing around the woods as music is playing. The results from the wandering are displayed and audibly announced, when the people in the same grid cell, along with the cheerful graphical display. Following that, the students can restart the game when it has been reset.

**Acceptance Criteria:** I can create a game for two people for specified conditions

* 1. **Use Case 2: For Grades 3-5**

**Primary Actor:** Student in Grades 3-5

**Preconditions:** This stage is created for students in Grades 3-5, and this stage is created for two /three /four players.

**Description:** The students can determine the size of the grid, and the shape of the grid might be rectangular. Students can place as many as four persons on their grid and there might be two, three, or four of them. It is possible to play and repeat the game several times after the statistics are displayed. It is a slightly more complicated version of the game. If three or four people are playing, two of them will move together if they come across one another until they come across the third person. If the third person is found then the three players move together until the fourth person is discovered.

**Acceptance Criteria:** I can create a game for a maximum of four people for specified conditions

* 1. **Use Case 3: For Grades 6-8**

**Primary Actor:** Student in Grades 6-8

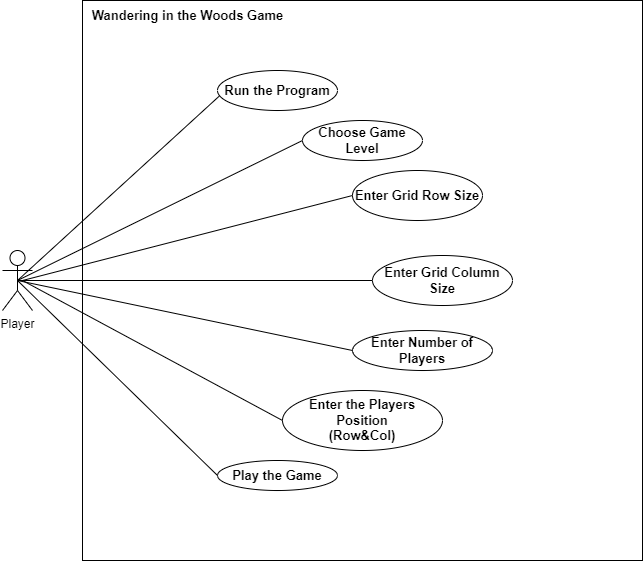
**Preconditions:** This stage is created for students in Grades 6-8, and this stage is created for a maximum of eight players.

**Description:**  In this stage, the player has full control over stage 3-5, they will be challenged for conducting experiments to find out how the typical run changes depending on the size and shape of the grids. It is a more complex stage in this simulation game. To reduce the time it takes to meet up, they will be able to compare several wandering procedures and determine which one is best.

**Acceptance Criteria:** I can create a game for a maximum of eight people for specified conditions

# UML Model

* 1. **Use Case Diagram:**

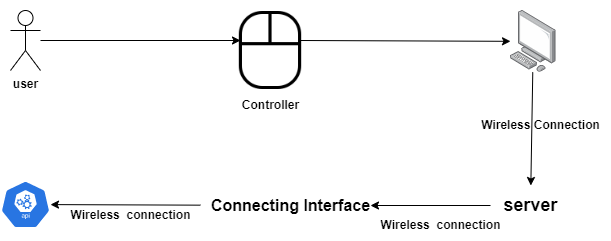


**Figure 2: Use Case Diagram for the Wandering in the Woods Game**

The different uses and potential scenarios for a system are represented visually in use-case diagrams. It should show the actions and interactions a user would have with a specific system, such as a website or an application (LETÍCIA FONSECA, 2022). An only actor is a person who interacts with the wandering in the woods game GUI. The person is able to start the program. After that, the GUI will display. In the GUI, the person is able to select the game level, based on the game level the person is allowed to enter the Grid row and column size, the number of players, and the position of those players. After entering all the details, the person is able to play the game.

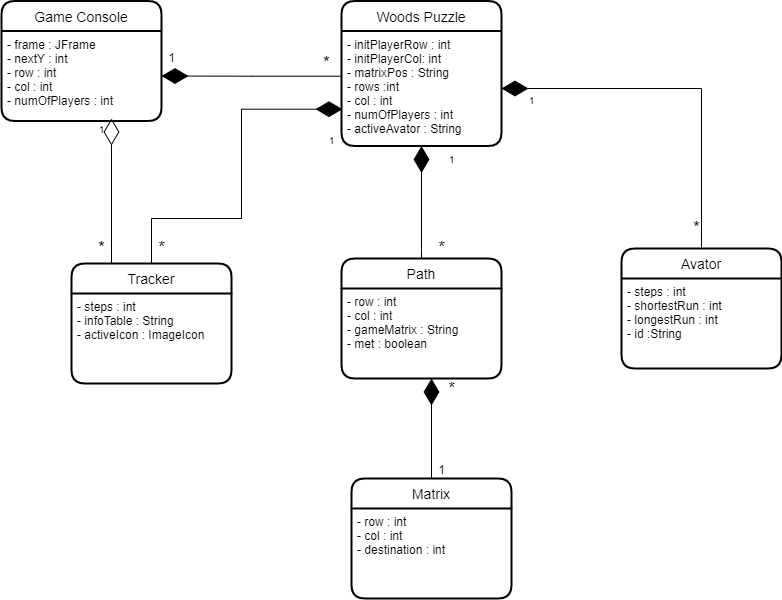
* 1. **Deployment Diagram:**

The deployment diagram is used to display the execution architecture of a system. The user controls the system using the handheld controller. The system is connected to a server in a wireless connection. The connecting interface connects the server and the application with a wireless connection network.



**Figure 3: Deployment Diagram**

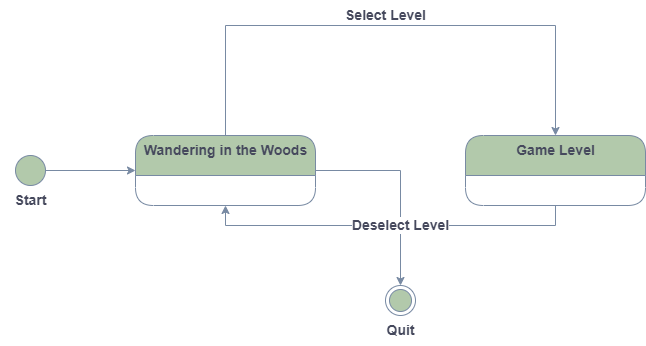
* 1. **Class Diagram:**

****

**Figure 4: Class Diagram**

The class diagram is used to describe the structure of the system by showing its classes, attributes (private, public, protected), and methods. The above-mentioned class diagram shows the classes that exist in this wandering in the woods game and the attributes of each class. The matrix class gets the row, col, and destination details from the user inputs. The tracker class tracks each player’s moves. The avatar class provides the avatar for each player. The wood puzzle class combines the avatar class, tracker class, and path class methods to create a wood puzzle simulation. The game console class is used to create a GUI for the players and get the input from the user and based on the user input create a simulation.

* 1. **State Diagram:**

****

**Figure 5: State Diagram**

The state diagram in figure 5 shows the major states of the system. The system starts in the initial startup state where a user runs the application. From the GUI user can select the game level, grid rows, and columns, Number of players, and their position in the grid. The system moves to the game state when the user selects the start button. If the user selects the reset game, the system moves to the wandering in the woods state. The system moves to the quit state when the user selects the exit option.

* 1. **Activity Diagram:**

****

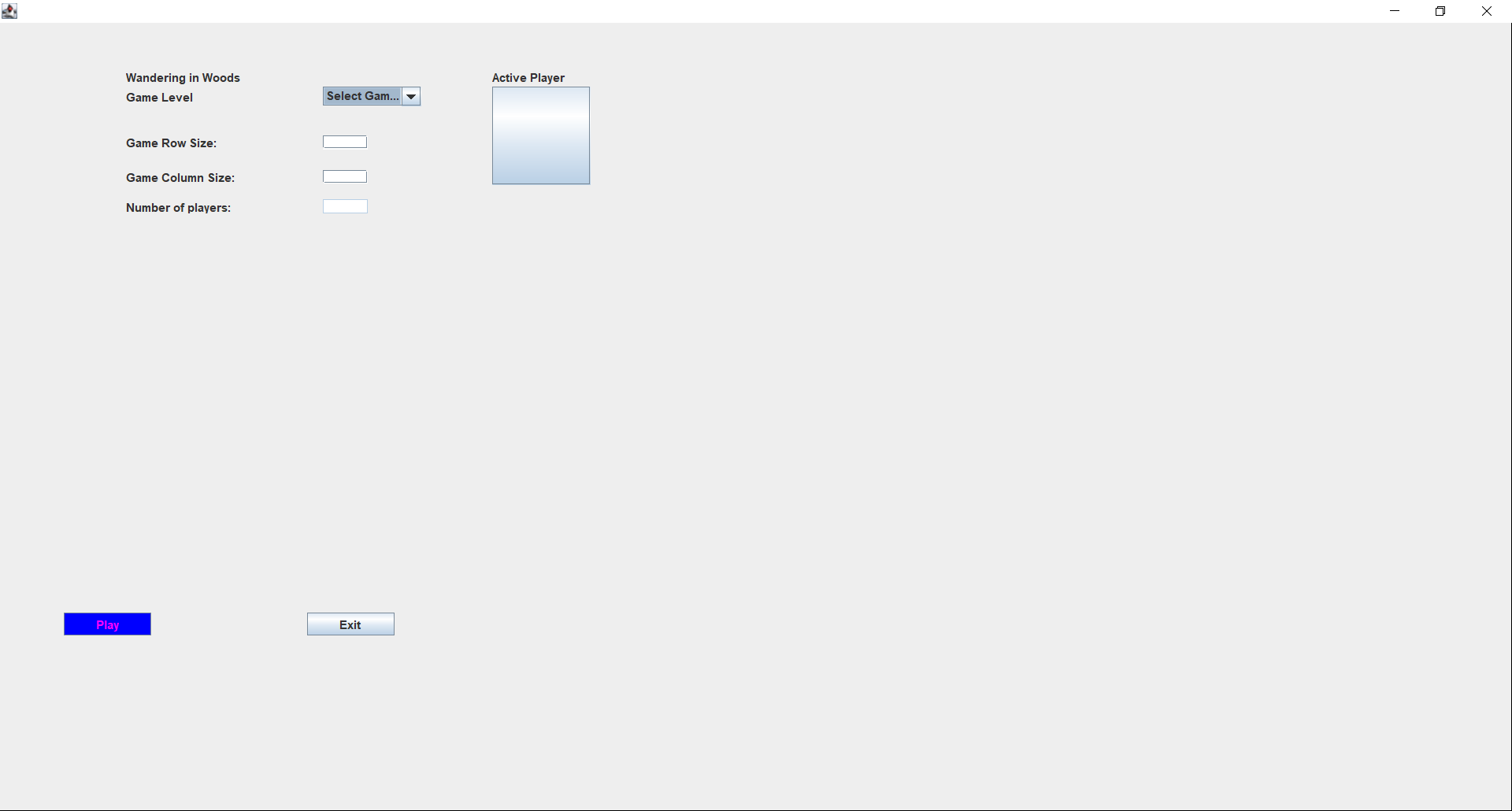
# Personas

In this game, the rectangular grids are used to represent the woods. The main concept of this game is the people are lost in the woods and the people cannot see or hear one another until they are in the same grid cell because the woods are dense. This simulation consists of three levels. In level K-2, the grid shape is always square. The main goal of this simulation game is for all the players in the same grid cell. To determine how well the students are handling the challenges, the simulation should contain audio suggestions and prompts also. In level 3-5, a maximum of four players can be allowed. In level 6-8, students have full control over level 3-5, they will challenge for conducting experiments to find out how the typical run changes depending on the size and shape of the grids. It is a more complex stage in this simulation game. To reduce the time it takes to meet up, they will be able to compare several wandering procedures and determine which one is best.

# GUI Mock-up

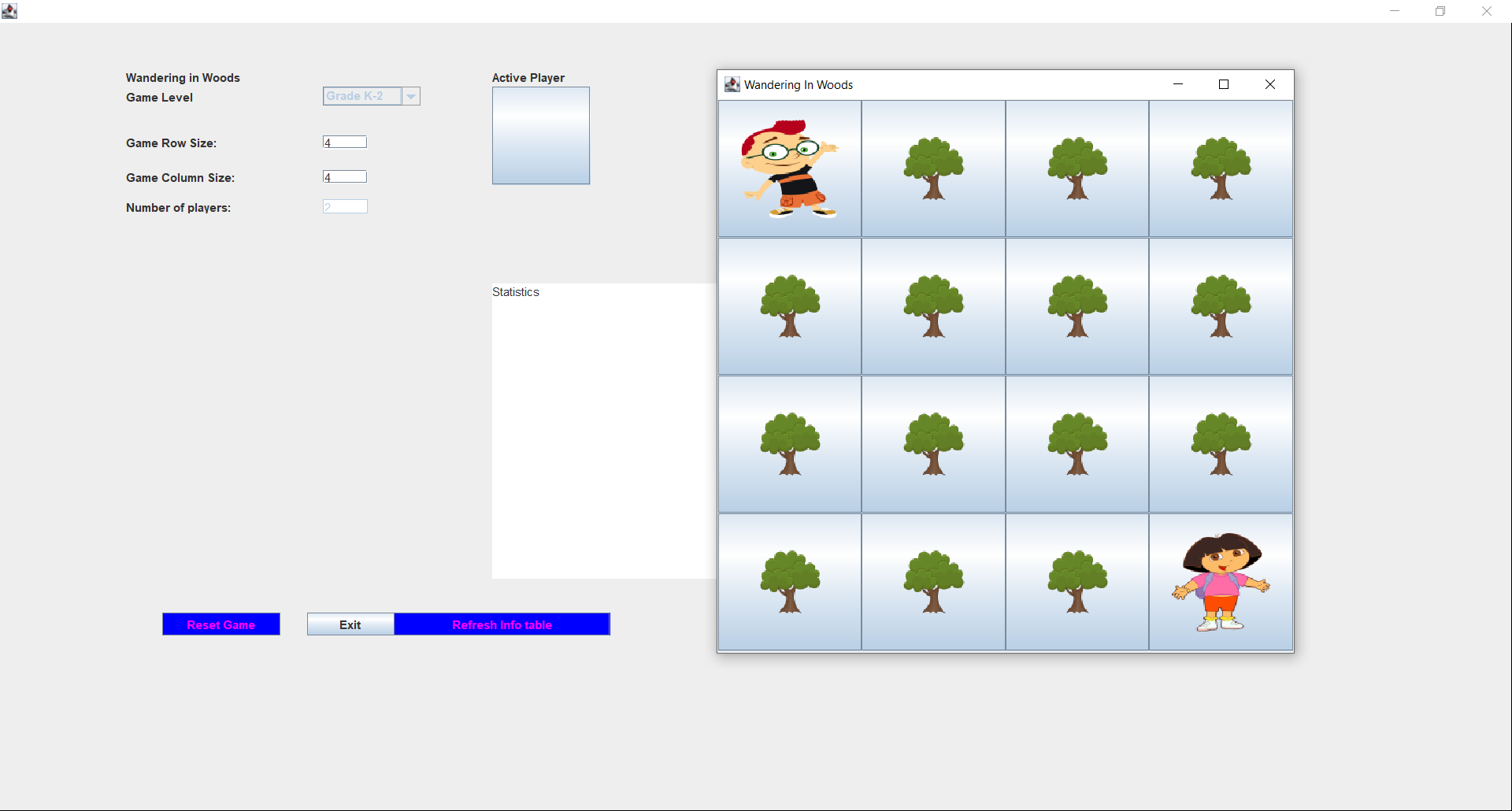
**Game Setup:**

This interface is used to select the game level, grid column, and row sizes.



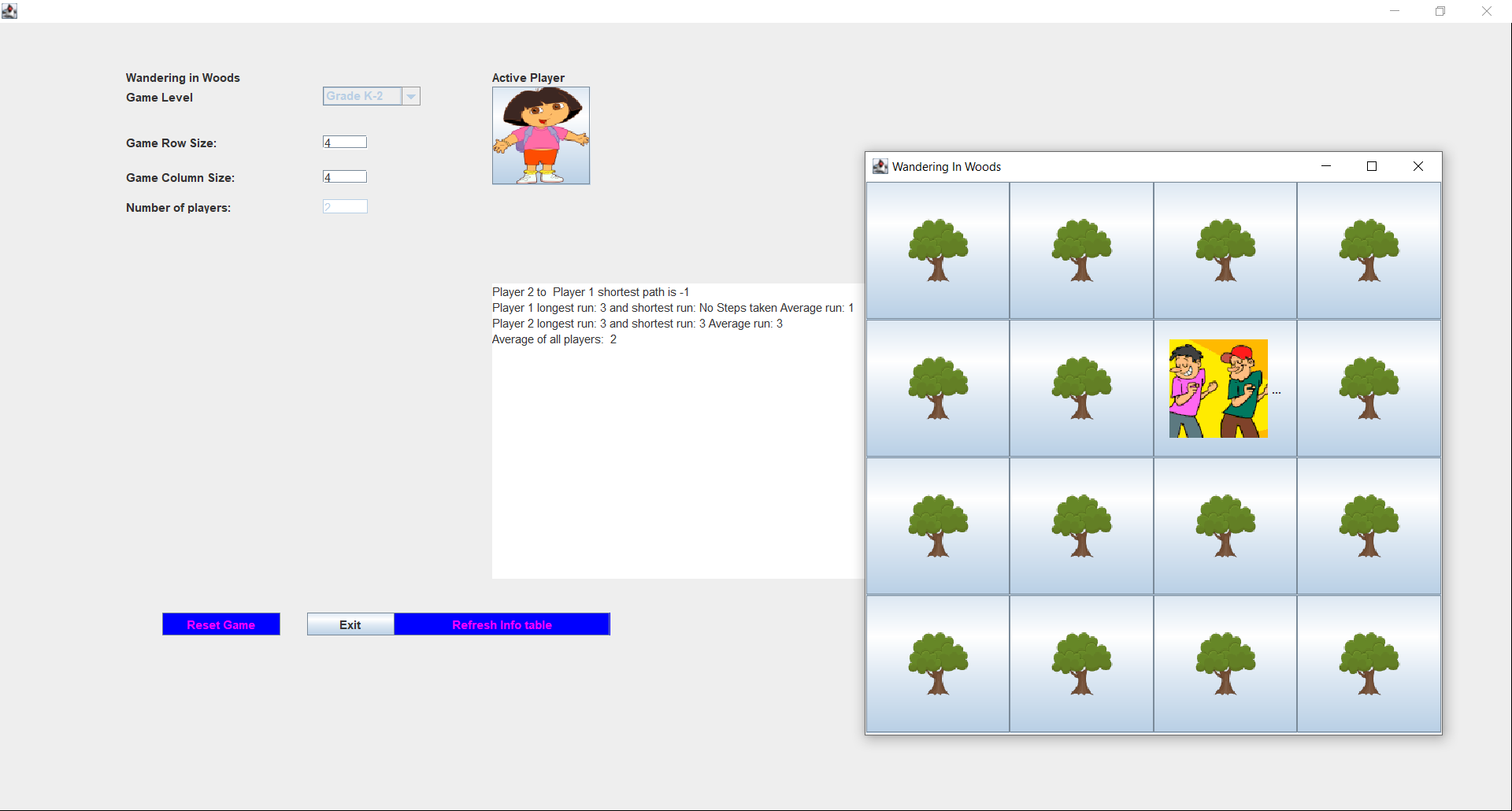
**Setup Interface for Grade K-2:**

In this interface, the student can select the game level, set the grid row size and column size, and the number of players is two as default.



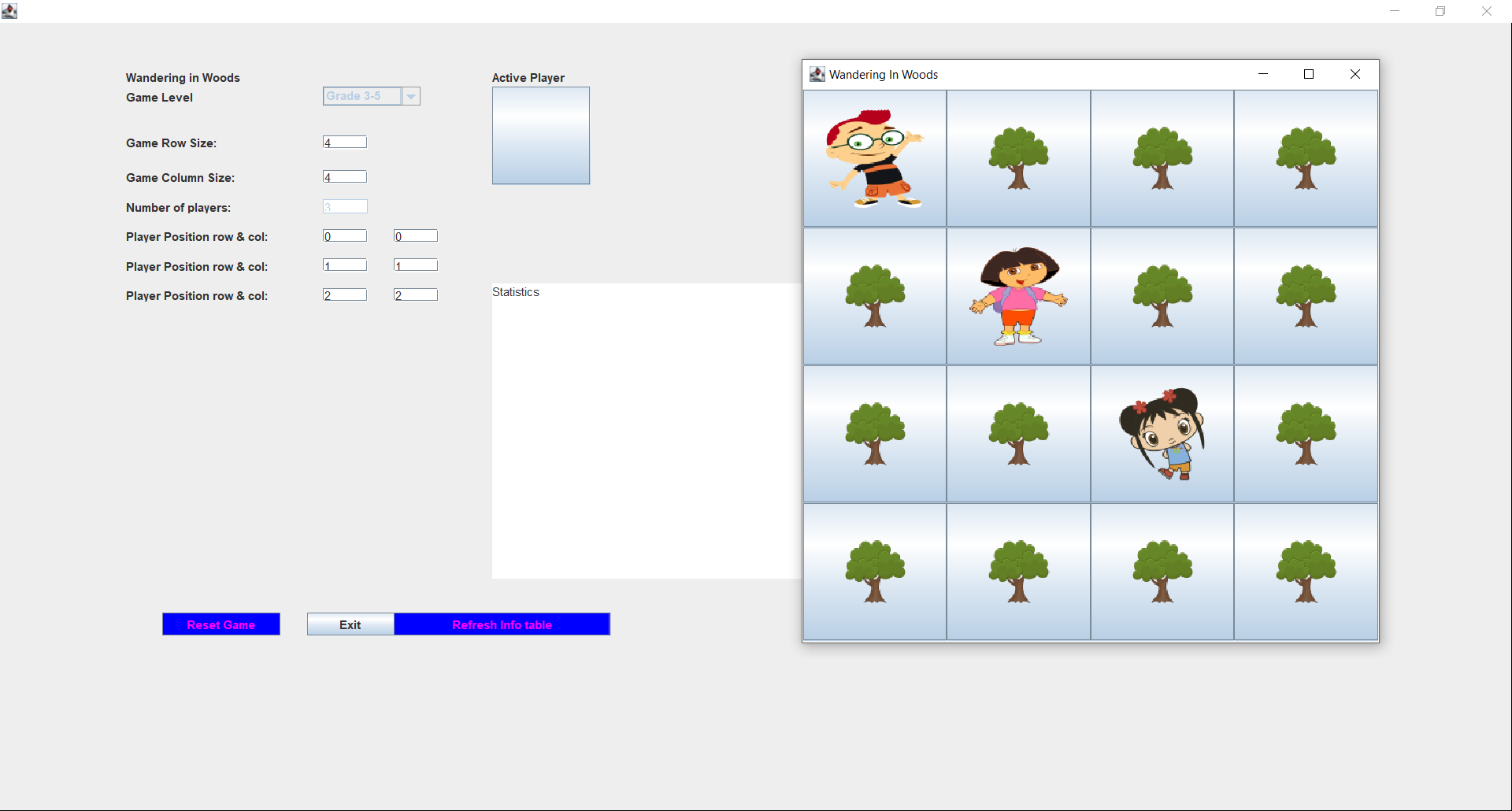
**Grade K-2: If the players meet:**

If the two players meet each other, the celebration avatar is displayed. Then click Refresh info table to view the game statistics.



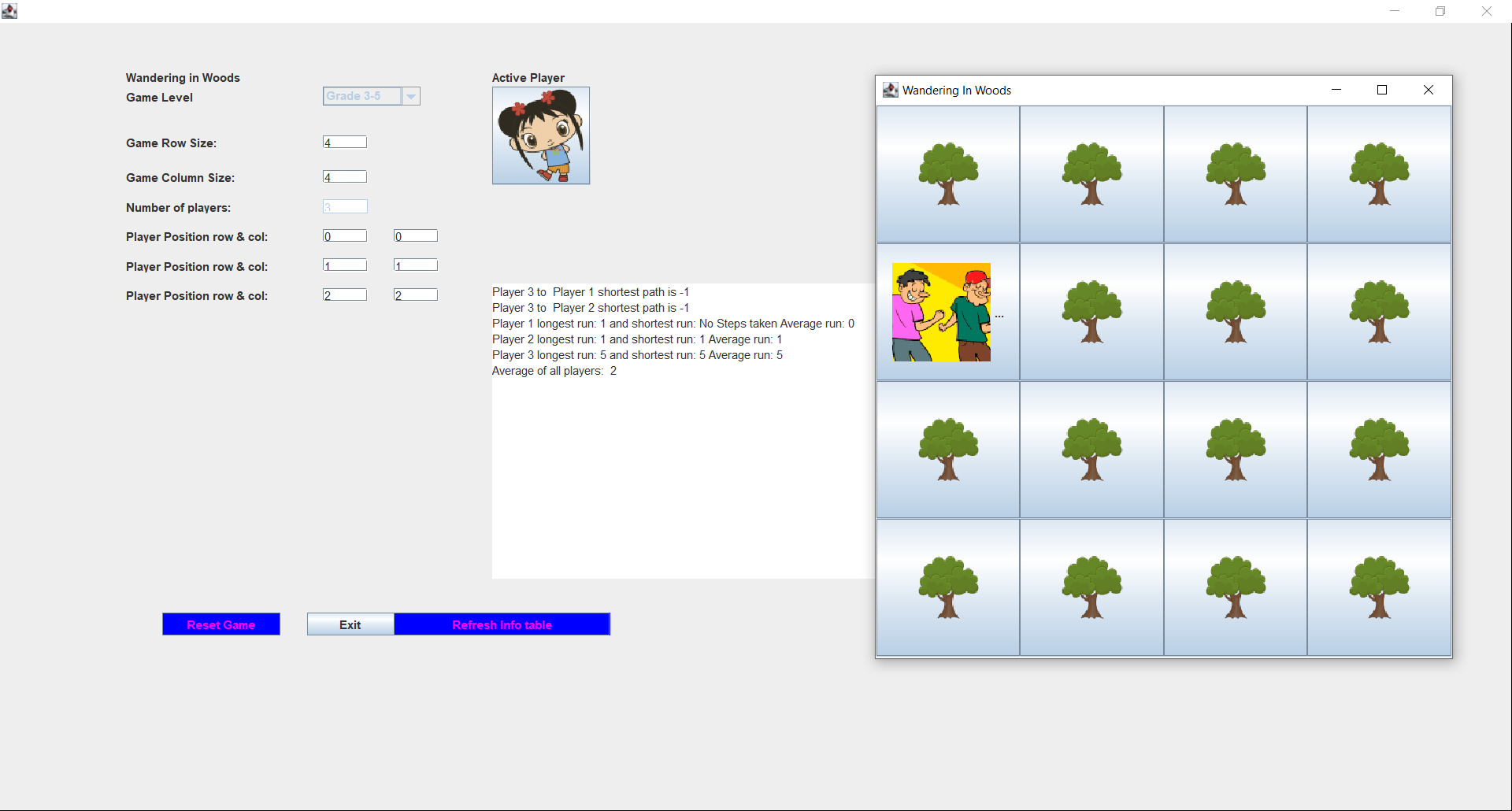
**Setup Interface for Grade 3-5:**

In this interface, the student can select the game level 3-5, set the grid row size and column size, and enter the number of players, and then set the position for those players in the grid. A maximum of four players can be allowed at this level.



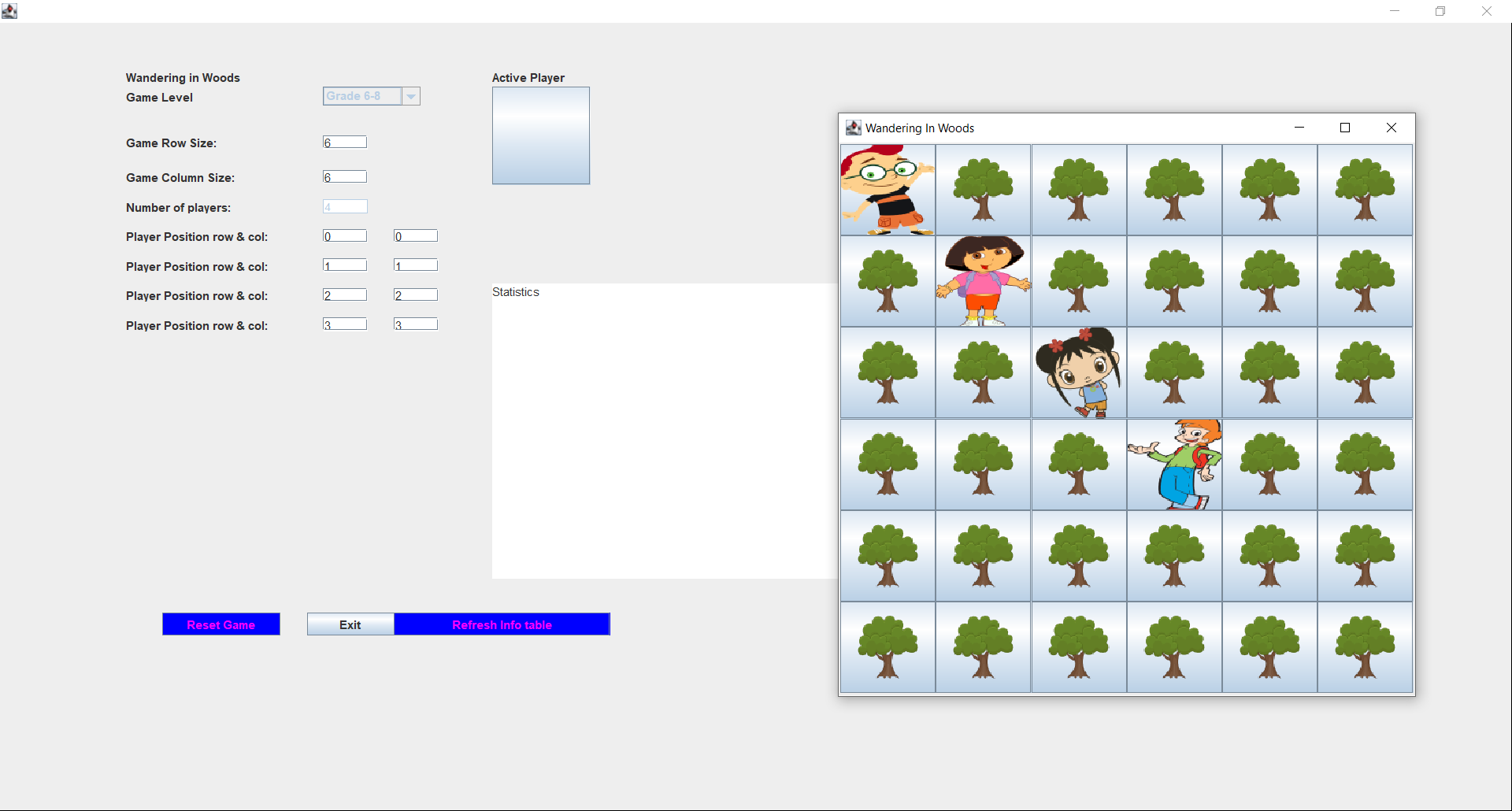
**Grade 3-5: If the players meet:**

If the two players meet each other, then those two move together to find other people. If all the people are met, then the celebration avatar is displayed. Then click Refresh info table to view the game statistics.



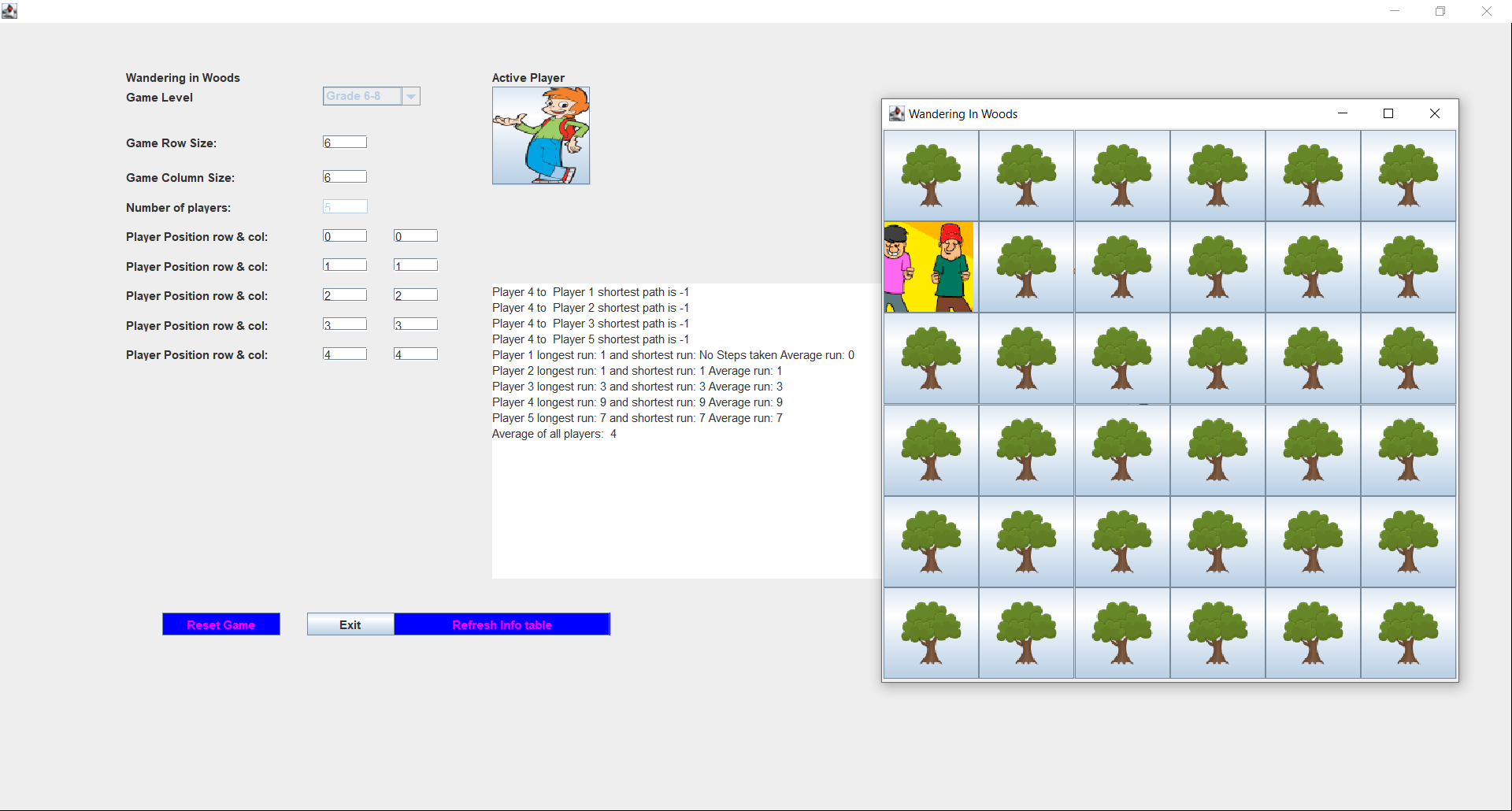
**Setup Interface for Grade 6-8:**

In this interface, the student can select the game level 6-8, set the grid row size and column size, and enter the number of players, and then set the position for those players in the grid. A maximum of eight players can be allowed at this level.



**Grade 6-8: If the players meet:**

If the two players meet each other, then those two move together to find other people. If all the people are met, then the celebration avatar is displayed. Then click Refresh info table to view the game statistics.



# Testing Strategy

Software testing is a crucial part of every software project that aims to ensure the long-term dependability and the proper operation of the system under development. The same amount of time, money, infrastructure, and experience is needed for software testing and for development. Systems with safety-critical software require more money and effort. For any organization in which the cost of software development is significant, having a solid testing plan is crucial ( Thota, Shajin, & Rajesh, 2020).



## Unit Testing

As a complement to integration and system testing, unit testing examines individual components. Because the developers are familiar with a component’s underlying implementation, unit testing of a component is meant to be done by those who created them. The functionality of every component should be covered by the unit test cases (Buchgeher & Pichler, 2020).



## Integration Testing

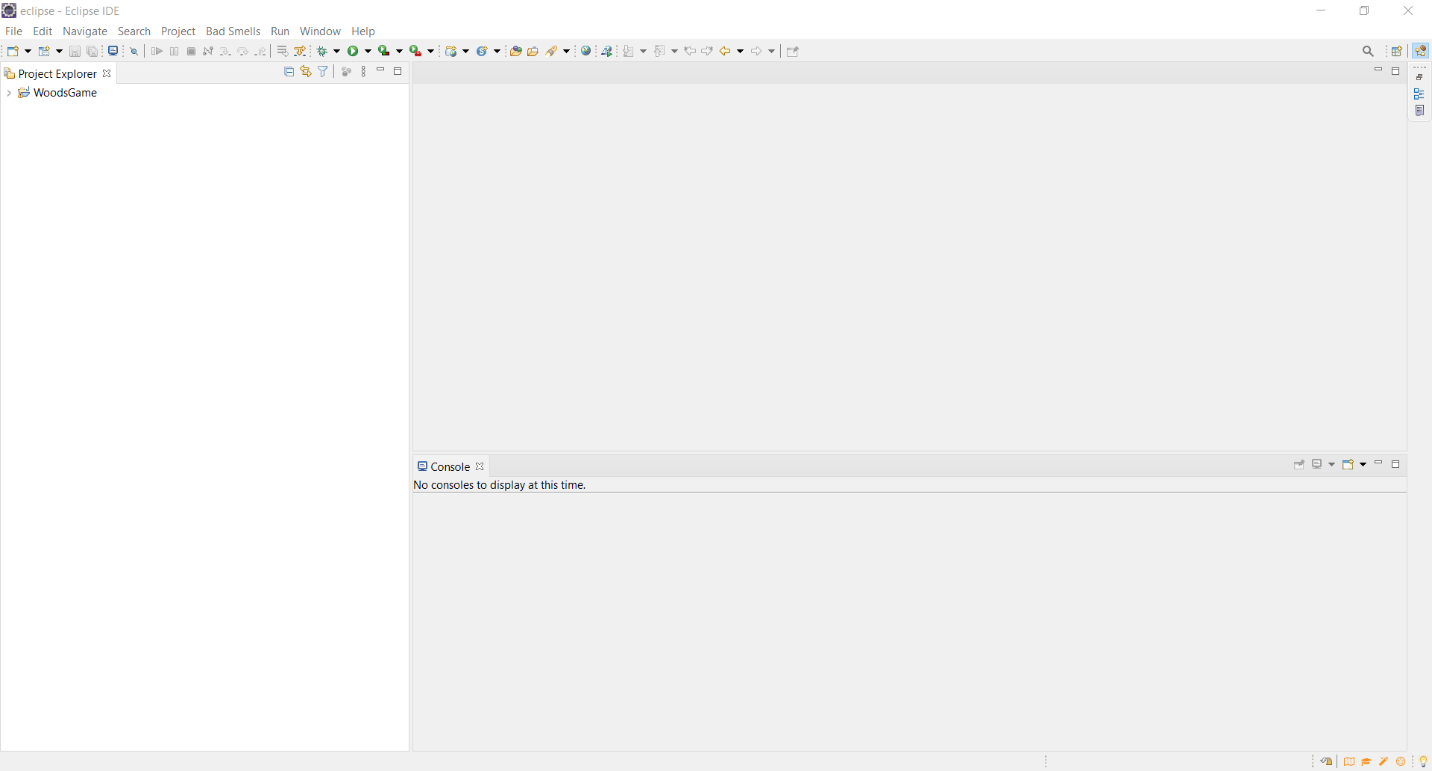
The next level of software testing is known as integration testing. It involves combining and testing various modules and components together. This type of testing is done to find the problems with how integrated units interact with one another. To help with integration testing, test drivers and test stubs are employed. The integrating testing main objective is to examine any flaws that could appear when these components are integrated and have to communicate with one another. It also aims to evaluate the interfaces between the modules. After the unit testing, which examines whether the smallest possible units of code are functionally accurate, integration testing is carried out. In the software, every unit is capable of being logically separated. Unit testing can disclose more detailed information about a smaller unit (A. B. Weikle, O. Lam, & S. Kirkpatrick, 2019).

## Usability test

The main objective of this testing is to identify how the Graphical user interfaces interact with users. This test is used to find the issues when working with the software application. During this test all the user interfaces are tested and find the interfaces are user-friendly or not.

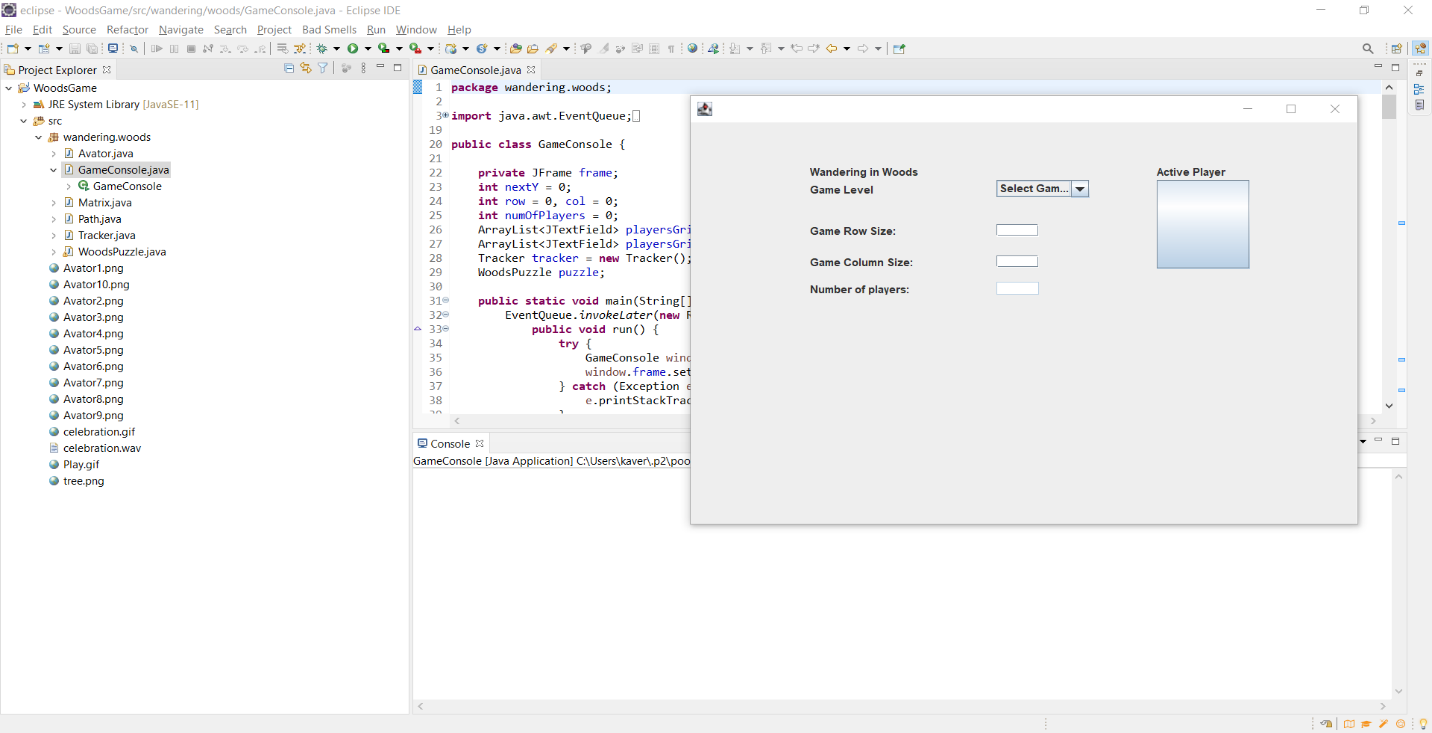
# Appendix – User manual

**Import the Project File:**

Open the Eclipse IDE; select the File🡪Open Projects from File system 🡪Import Source🡪select the project folder and then click Finish.

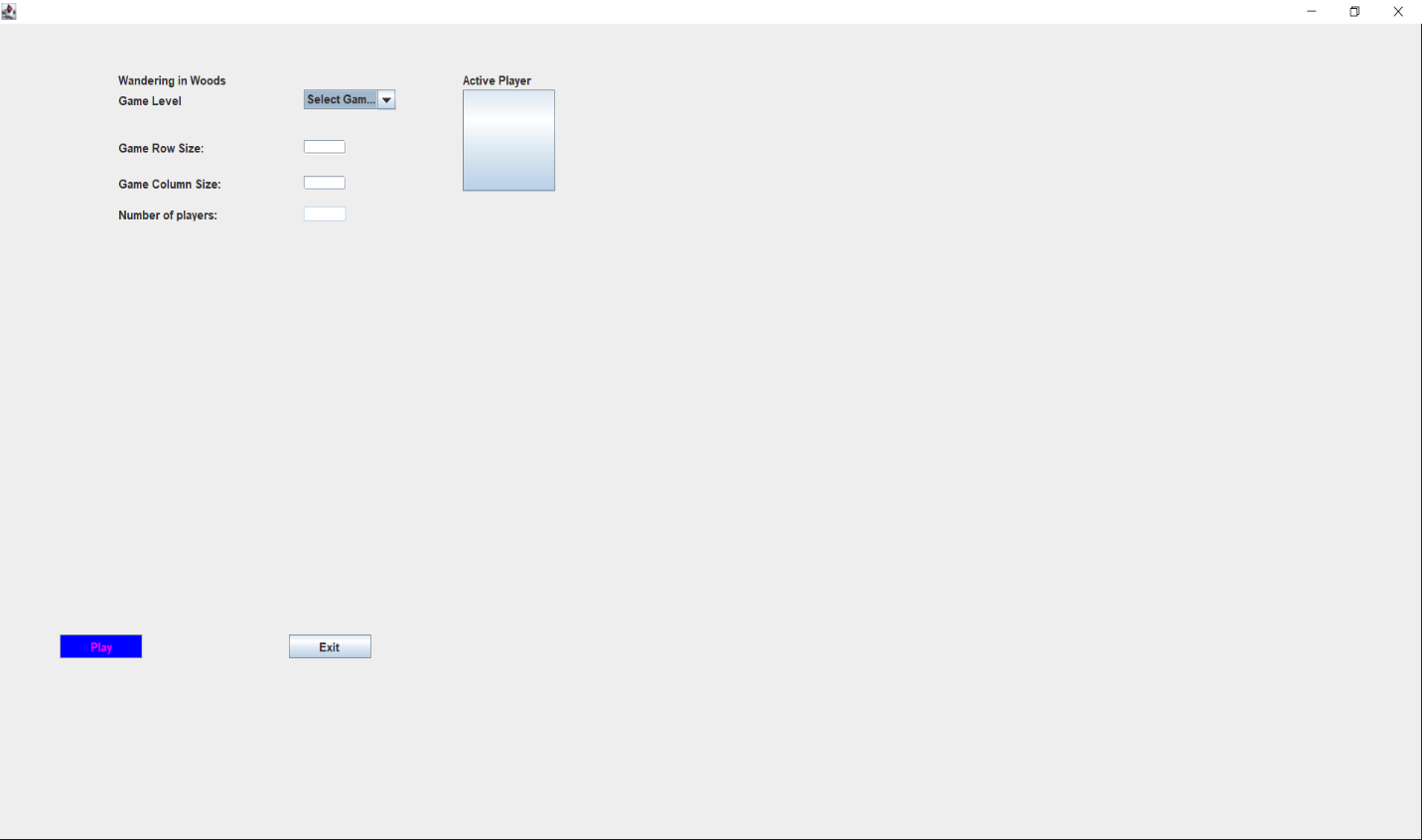
**Run the Program:**

Select the file GameConsole.java in the wandering woods package. Right Click on GameConsole.java file, select Run as 🡪 Java Application.



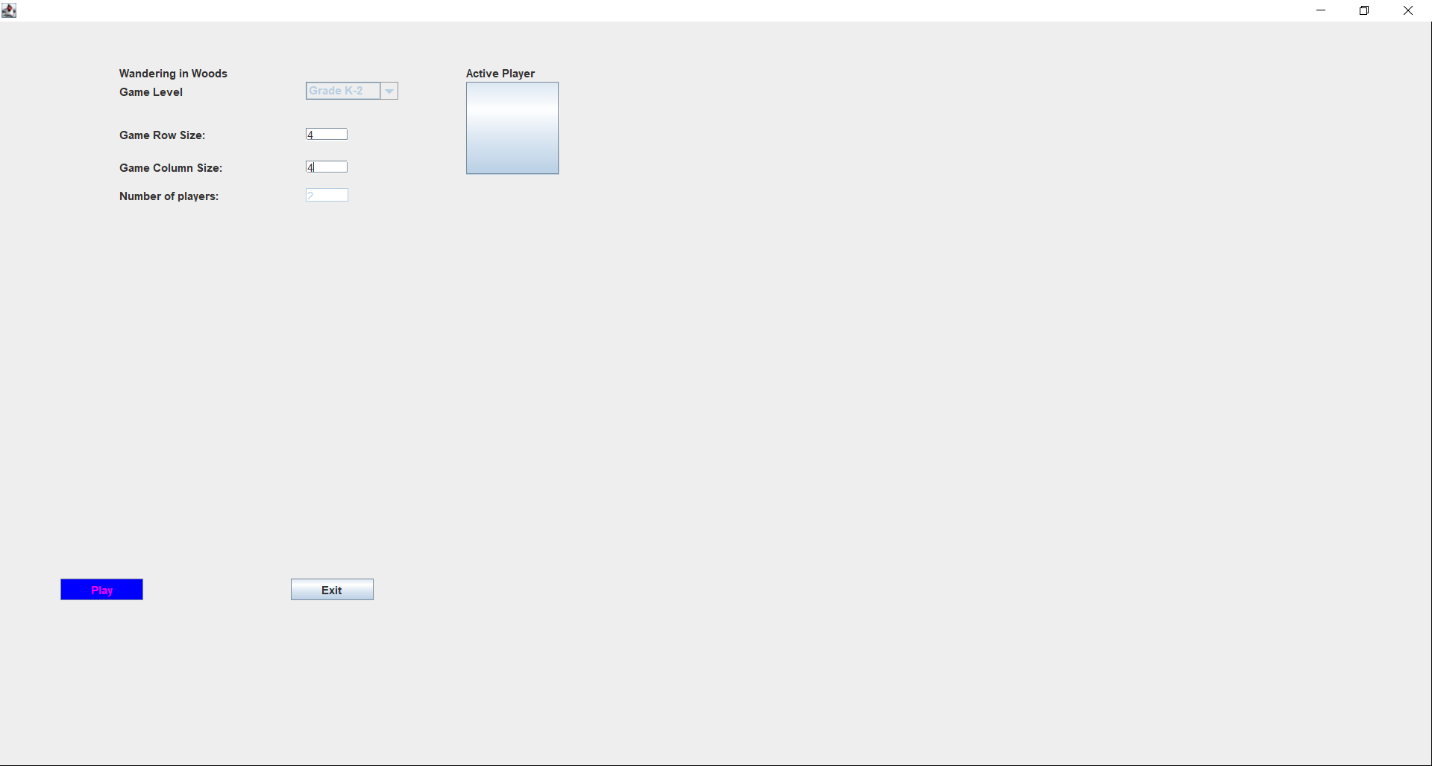
**Game GUI:**

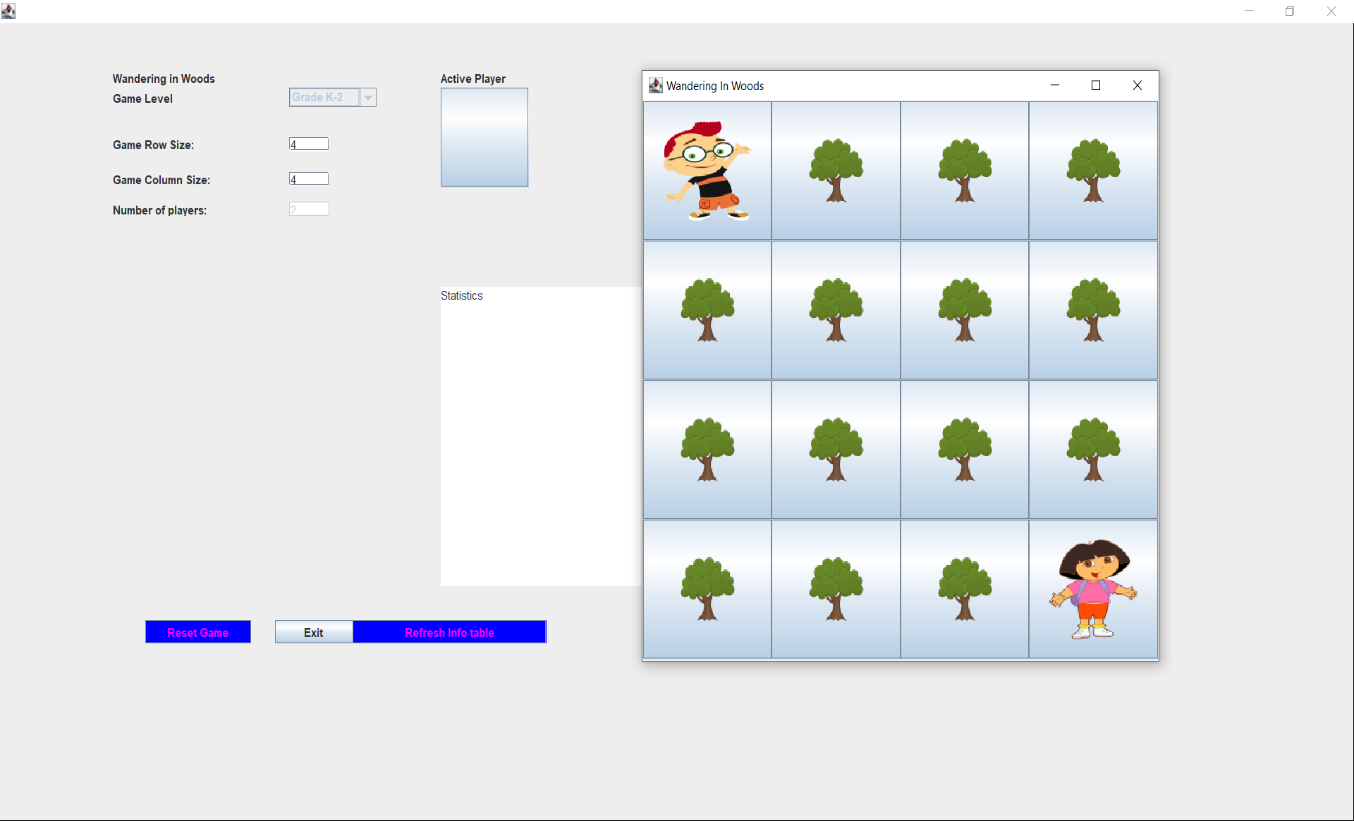
After run the program, maximize the GUI. The Game GUI show in the below figure.

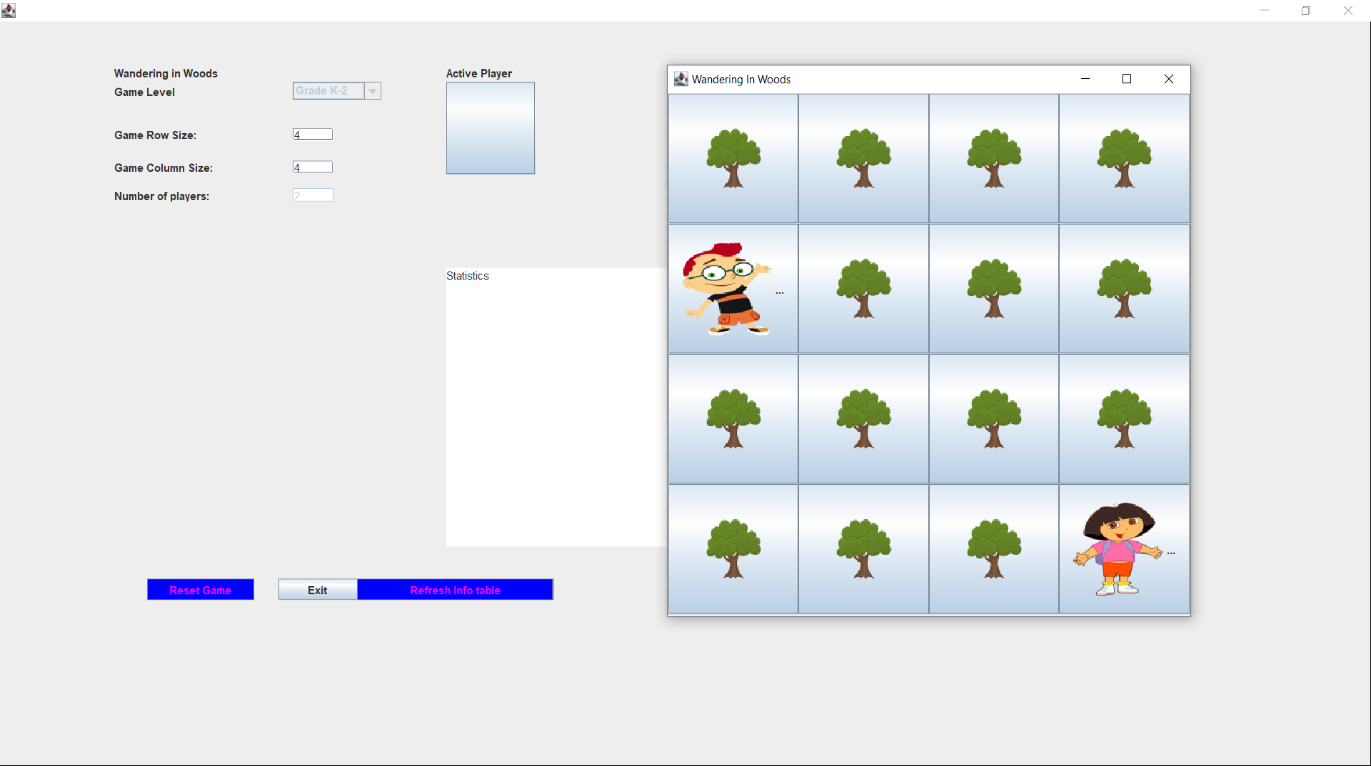


**For Grade K-2:**

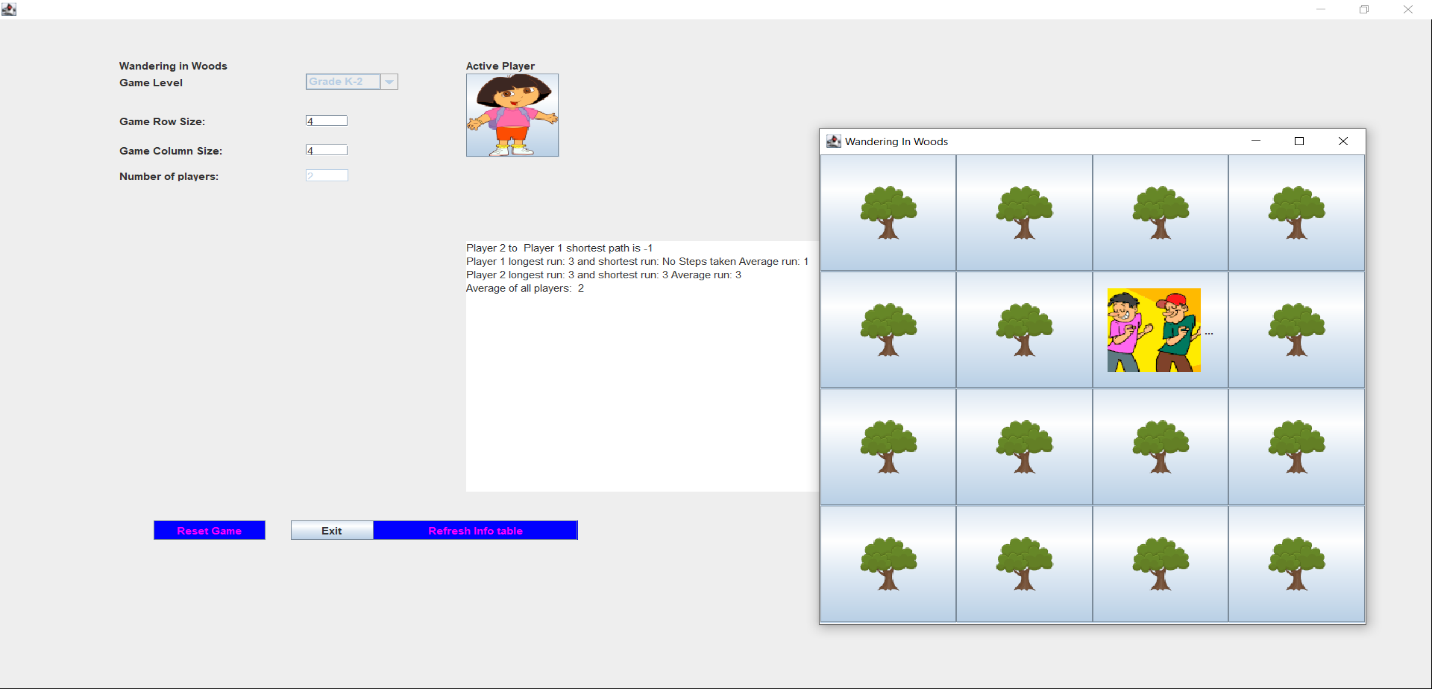
* For Grade K-2 students, select Grade K-2 in the Game Level and then enter the Game Grid row and Column size.



* Then click play button in the GUI. The Game Interface look like the below figure. 
* Use the arrow keys in keyboard to play the game.



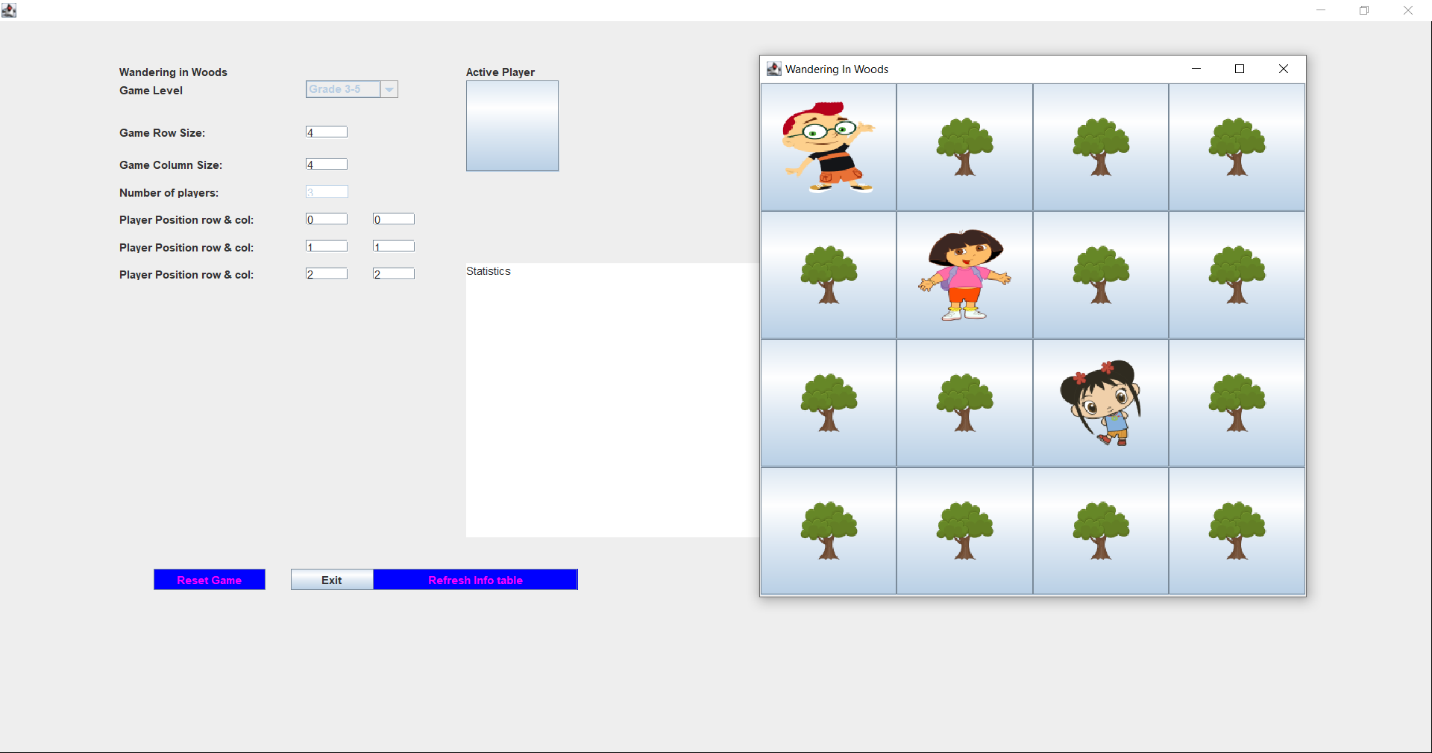
* If the players meet each other, then the application display the celebration avatar and play the music.
* Select “Refresh Info Table” button in the game setup GUI to display the Statistics.



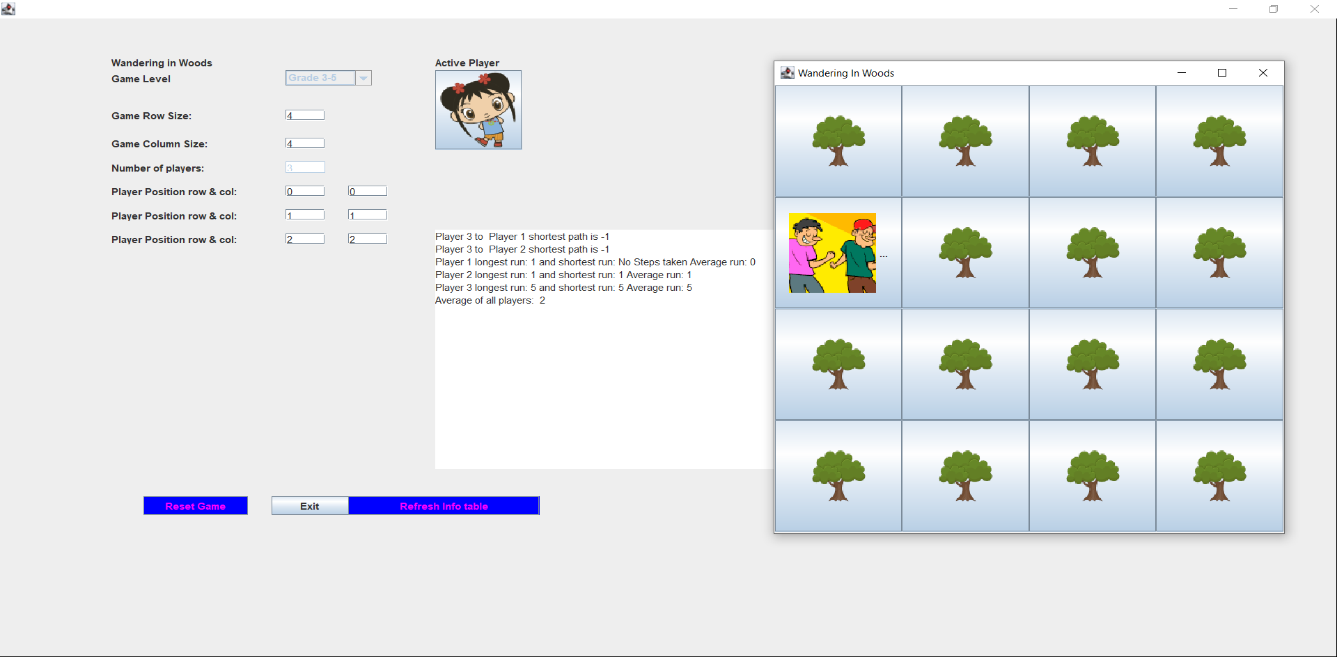
* The “Reset Game” button reset the position in the Game GUI.
* Click button “Exit” to exit from the application.

**For Grade 3-5:**

* For Grade 3-5 students, select Grade3-5 in the Game Level and then enter the Game Grid row and Column size and enter the number of players, and then set the position for those players in the grid.
* A maximum of four players allowed at this level.
* Then click play button in the GUI. The Game Interface look like the below figure.



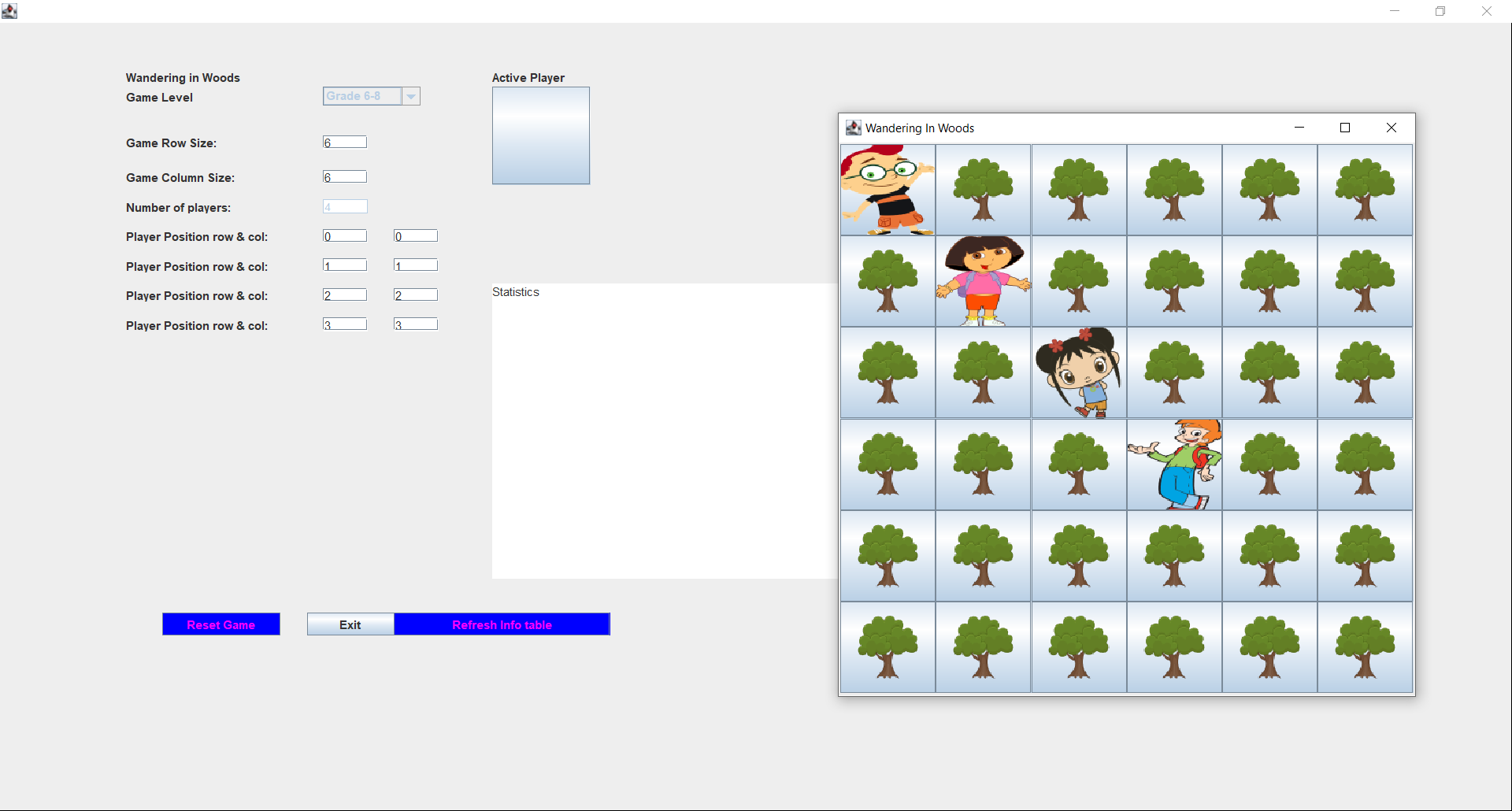
* Use the arrow keys in keyboard to play the game.
* If the two players meet each other, then those two move together to find other people. If all the people are met, then the celebration avatar is displayed.
* Then click “Refresh info table” to view the game statistics.



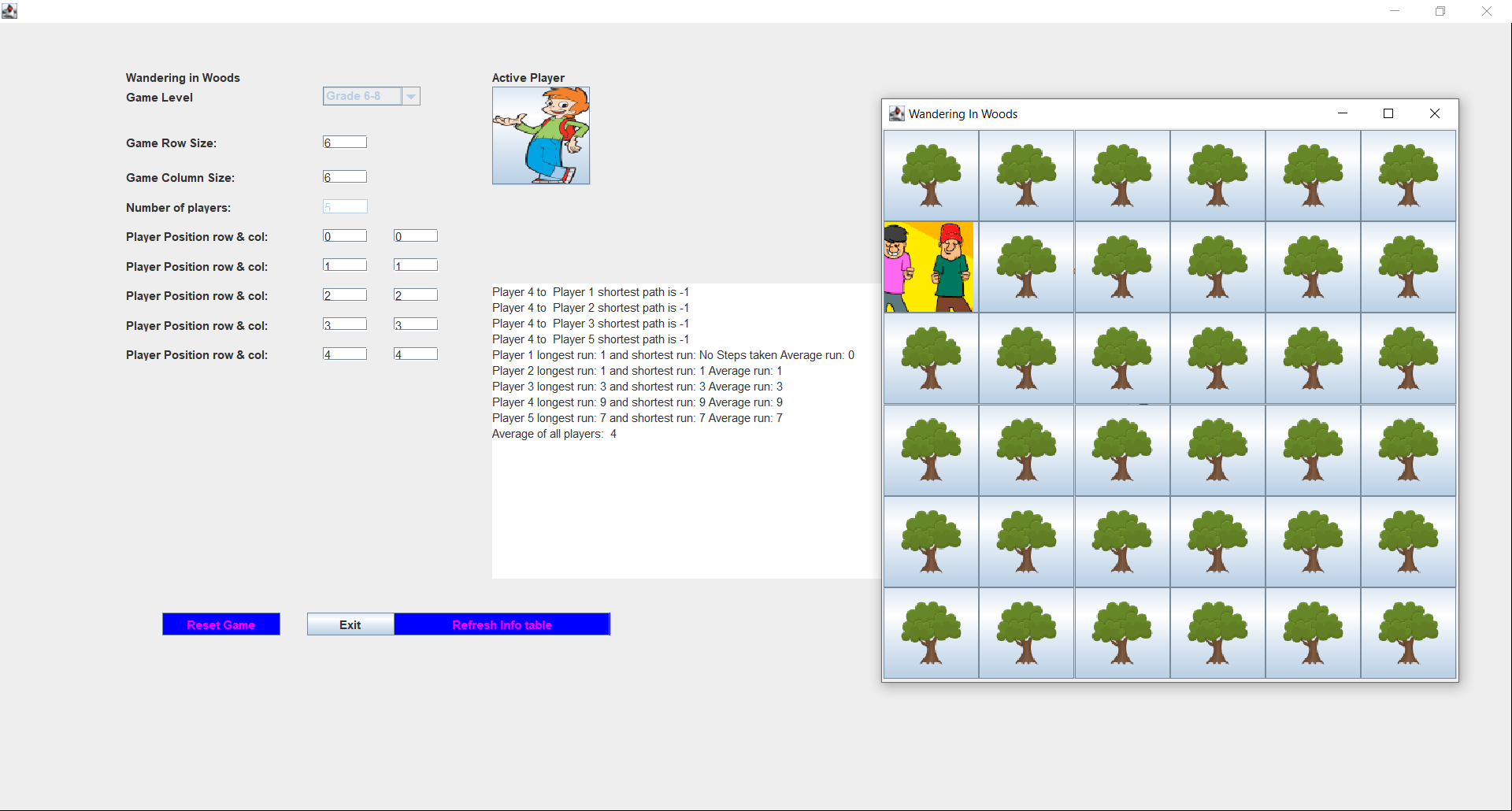
* The “Reset Game” button reset the position in the Game GUI.
* Click button “Exit” to exit from the application.

**For Grade 6-8:**

* For Grade 6-8 students, select Grade 6-8 in the Game Level and then enter the Game Grid row and Column size and enter the number of players, and then set the position for those players in the grid.
* A maximum of eight players allowed at this level.
* Then click play button in the GUI. The Game Interface look like the below figure.



* Use the arrow keys in keyboard to play the game.
* If the two players meet each other, then those two move together to find other people. If all the people are met, then the celebration avatar is displayed. Then click “Refresh info table” to view the game statistics.



* The “Reset Game” button reset the position in the Game GUI.
* Click button “Exit” to exit from the application.

# References

Thota, M. K., Shajin, F., & Rajesh. (2020). Survey on software defect prediction techniques. *International Journal of Applied Science and Engineering*, 331-334.

A. B. Weikle, D., O. Lam, M., & S. Kirkpatrick, M. (2019). Automating Systems Course Unit and Integration Testing. *Paper Session: Testing 2*, 556-570.

Buchgeher, G., & Pichler, J. (2020). An Early Investigation of Unit Testing Practices of Component-Based Software Systems. *IEEE Xplore* (pp. 12-15). London, ON, Canada: IEEE.

Dave Swersky. (2022, June 1). *The SDLC: phases, popular models, benefits & more*. Retrieved from RAYGUN: https://raygun.com/blog/software-development-life-cycle/

LETÍCIA FONSECA. (2022, February 15). *What is a use case diagram?* Retrieved from VENNGAGE: https://venngage.com/blog/use-case-diagram-example/