Contents

[1 Introduction 5](#_Toc112354182)

[1.1. Purpose 5](#_Toc112354183)

[1.2 Game Goals 5](#_Toc112354184)

[1.3 Scope of the Project 6](#_Toc112354185)

[2 Process Model 6](#_Toc112354186)

[3 Use Cases 7](#_Toc112354187)

[3.1 Use Case 1: Grades K-2 7](#_Toc112354188)

[3.2 Use Case 2: Grades 3-5 7](#_Toc112354189)

[3.3 Use Case 3: Grades 6-8 8](#_Toc112354190)

[4 Use Case Diagram: 8](#_Toc112354191)

[4.1 Deployment Diagram 9](#_Toc112354192)

[4.2 Class Diagram: 10](#_Toc112354193)

[4.3 State Diagram 11](#_Toc112354194)

[4.4 Activity Diagram 11](#_Toc112354195)

[5 Personas 12](#_Toc112354196)

[6 GUI Mock-up 13](#_Toc112354197)

[6.1 Home Screen – Grade Selection 13](#_Toc112354198)

[6.2 GradeK2 – Initiate the Grid Size 13](#_Toc112354199)

[6.3 GradK2 – Game window 15](#_Toc112354200)

[6.4 GradeK2 – Two players meet 15](#_Toc112354201)

[6.5 Grade3-5 – Initiate the Grid Size 16](#_Toc112354202)

[6.6 Grad3-5 – Game window 16](#_Toc112354203)

[6.7 Grade3-5 – All players meet 17](#_Toc112354204)

[6.8 Grade 6-8 – Initiate the Grid Size 18](#_Toc112354205)

[6.9 Grad 6-8 – Game window 19](#_Toc112354206)

[6.10 Grade 6-8 – All players meet 19](#_Toc112354207)

[7 Testing Strategy 20](#_Toc112354208)

[7.1 Integration Testing 21](#_Toc112354209)

[7.2 Performance Testing 21](#_Toc112354210)

[7.3 Usability Testing 21](#_Toc112354211)

[7.4 Acceptance Test 22](#_Toc112354212)

[References 22](#_Toc112354213)

# Wandering in the Woods Game

# Introduction

## Purpose

This document details the creation of the kid-friendly game "Wandering in the Woods Game." This report details the software development of the "Wandering in the Woods Game" with simulation for three levels for grades K–2, 3-5, and 6–8, with complexity ranging from simple to slightly complex to complex.

Java script was used in the game's design. It is intended to replicate the schoolchildren's virtual atrium and provide an age-appropriate recreation of a walk through the woods. The creators of online games will find this document to be helpful. The system's characteristics and visual specifications are analyzed to kick off the project. For improved learning, it is essential to comprehend the guidelines for each simulation.

## Game Goals

The game's simulation is made to be played by two people in front of a screen. It is intended to symbolize the residents of the blocks. The K–2 simulation should be square and start from the corners that are diagonally opposed. The rectangular blocks are used to symbolize the forests. The first game rule states that players cannot see or hear each other until they are inside the same grid block. The default theme, "Wander in the World," must be played until the players start wandering. For the K-2 simulation, a screen with cheery graphics is provided along with the statistics.

For grades 3 through 5, the simulation's complexity rises. The game supports 2, 3, and four players. And let them choose where on the grid to begin. The system must show the number of shortest runs, the number of runs without a meeting, and the average score at the conclusion of the game. The pupils are given the opportunity to alter their grid size and investigate the best method for wandering to discover the shortest path in the grade 6 to 8 wandering simulation (Cardoso & Romão, 2021).

The system is built to support the creation of the gaming environment using Java Slick 2D. There are various benefits to adopting this JAVA library, including cost savings and the fact that it doesn't need any specialized hardware to be developed environments. The goal of this idea is to create a game. The pre-design of the visual images and the theme music for the game's wandering in the woods should be created. Given that this is a 2D gaming environment, Slick 2D can be the ideal tool for creating a kid-friendly game.

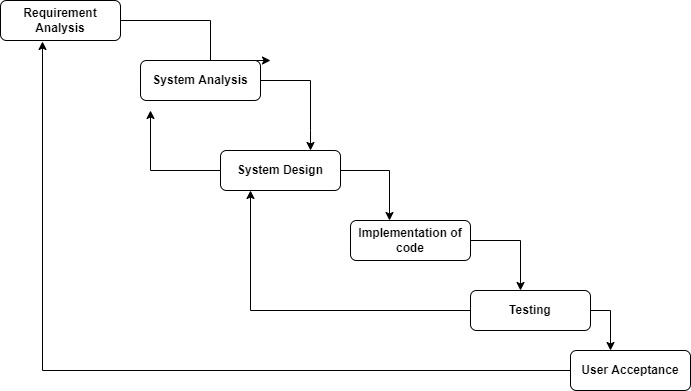
## Scope of the Project

The goal of the project is to create a 2D gaming environment that makes use of the EventKey library to let users press keys to move around grids. For kindergarten and lower grade kids, the purpose of this game is to have fun while experiencing an adventure in a grid-controlled environment. For students in grades 6 through 8, the game has evolved to effectively teach computation, scientific reasoning, arithmetic principles, and computer expertise. The scope of the game is listed below

* 2, 3, and 4 players
* PC based
* 2D animation game
* Source code Witten using JavaScript
* 2D platform for GUI interface
* Testing with null errors in the game environment

# Process Model

The waterfall model is employed for the creation of the Wandering in the Woods Game, as seen in Figure 1. The waterfall model was chosen because it has a straightforward design environment and does not require a continuous iteration process. This process flow diagram shows the list of tasks that have been sequentially structured in order to complete the project. This will aid in coming up with ideas for the project's early stages of development (Czauderna & Guardiola, 2019).



**Figure 1: Process flow diagram for the development of the Wandering in the Woods Game.**

# Use Cases

## Use Case 1: Grades K-2

**Primary Actor:** Player

**Preconditions:** The game has been created for two peoples

**Description:** As a student, run the code and then selects the game type Grade K-2. Then give the number of columns and rows of the grids. After assigning row and column click the ok button then display the game. In grades K-2 the grids are square with two people and start in diagonally opposite corners of the grid. When starting the game each move is counted for each person. When the people meet each other displayed the happy graphics and display each people’s count then the game is restarted again with two people.

**Acceptance Criteria:** create a game for two people with specified conditions.

## Use Case 2: Grades 3-5

**Primary Actor:** Player

**Preconditions:** The game has been created for maximum 4 people

**Description:** As a student, run the code and selects the game type grades 3-5. After selecting the game type set the number of rows, columns and players then assign the player's position on the grids. The student can place the player wherever on their grid and then click the ok button to display the game. Once start the game can be played multiple times. If the players meet each other display happy graphics otherwise each step is counted for each player. The end of the game displays the statistics with the longest run without a meeting, shortest run, and average run.

**Acceptance Criteria:** create a game for a maximum of 4 people with specified conditions.

## Use Case 3: Grades 6-8

**Primary Actor:** Player

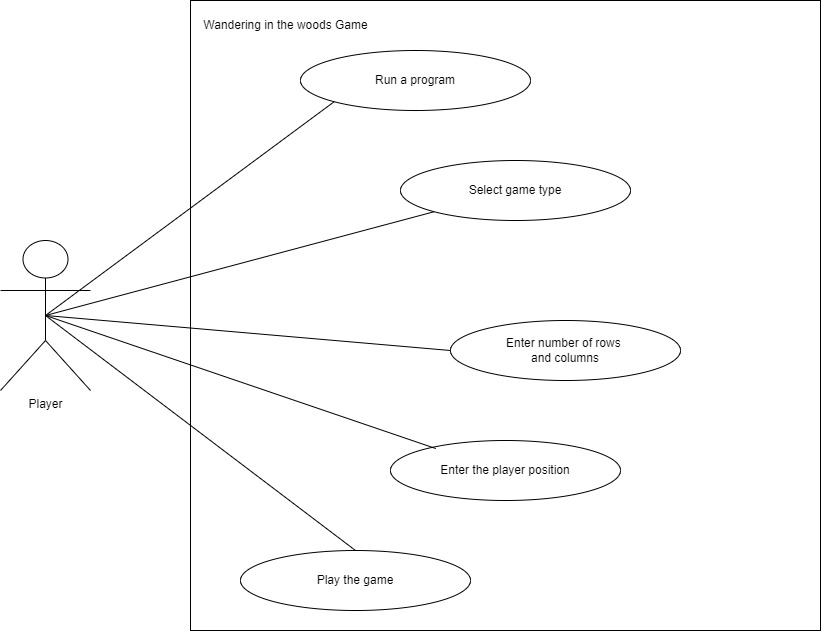
**Preconditions:** The game has been created for a maximum of 8 people

**Description:** As a student, run the code and selects the game type grades 6-8. After selecting the game type set the number of rows, columns and players then assign the player's position on the grids. The student can place the player wherever on their grid and then click the ok button to display the game. Once start the game can be played multiple times. If the players meet each other display happy graphics otherwise each step is counted for each player. The end of the game displays the statistics based on the protocols.

**Acceptance Criteria:**  create a game for a maximum of 8 people with specified conditions

# Use Case Diagram:

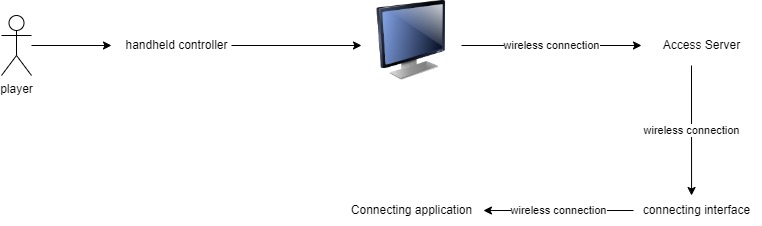
The only player interfaces with the game. The player is able to run the program. After running the program player is able to select the game type. Based on game type enter the number of columns and rows and player positions. After entering all values able to play the game.



**Figure 2:Use case diagram**

## Deployment Diagram

The deployment of the wandering in the woods game system software handheld controls the system. The system is with wireless connection to the server. The access server connected to the connecting interface then should be connected with the application. the wandering in the wood game software runs on an access server for the system which we view as running on a player’s personal home computer or laptop.



**Figure 3: Deployment diagram**

## Class Diagram:

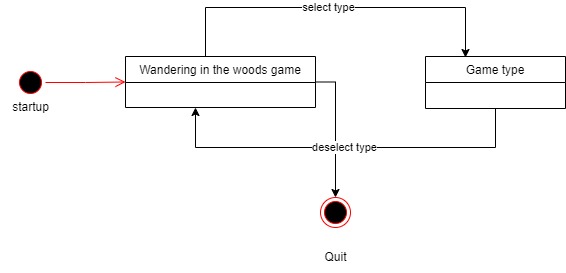
The class diagram has the number of entities. The wandering in the woods game has a player as a string. The game type has different types of games like grades K-2, grades3-4, and grades6-8. The grades K-2 have columns, rows, and positions for the players. Grades 3-4 have the same grades K-2 as well as grades 6-8. Finally have a statistic for displaying the longer time, shorter time, and average time for individual players.

## class.jpg

**Figure 4:Class Diagram**

## State Diagram

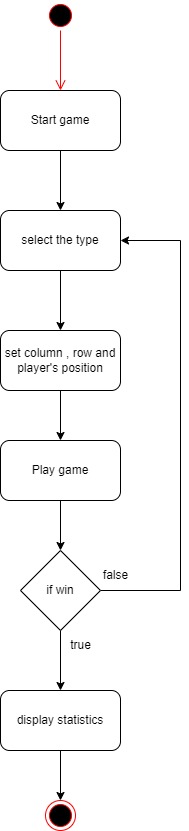
The system starts in the initial **startup state** where a player selects a game type. This system starts from startup to the wandering in the woods game. The state of wandering in the woods game display the game type selected by the player. The Game type state is executed based on the player's selected game type. The players can able to edit the number of rows, columns, and the player’s position of the game type. The player deselects the game type they are editing returning to the wandering in the woods game. Finally, the players quit the wandering in the woods game.



**Figure 5: State Diagram**

## Activity Diagram

The activity diagram presents a description of the wandering in the woods game. The players start the game. When the player starts the game select the game type then enter the number of rows, columns, and position of players for the selected game type. After entering all values the players should play the game. If the player win shows the statistics of the player’s running time the quit the program otherwise start the game from the beginning.



**Figure 6: Activity Diagram**

# Personas

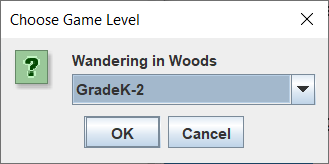
Three levels are assigned to each player throughout the game. For kids in kindergarten through grade 2, level 1 has a two-player scenario. The grid blocks might serve as a representation of the user. The statistics roaming and interactive audio announcement of the display will be displayed on the output screen during this implementation phase. Level 2 is for students in grades 3 to 5, and it allows for the participation of 2, 3, and 4 players. It will result in the game's shortest path taken and longest run in a free-roaming environment. In Level 3, users are given the option to alter the size of their grid and learn how to determine the shortest and longest paths in a roaming environment. The user environment of the game is depicted in the following figure to help with implementation knowledge.

# GUI Mock-up

The user interface created for the gaming environment is depicted in the accompanying figure. The two players of the game "wandering in the woods" are visible on the screen. It shows the menu for adjusting the volume. For each of the three tiers, a different user interface screen will be used. It is straightforward for level 1 and offers the ability to start, pause, and display the statics. For level 2, it might show the screen and provide options to show statistics and determine the average. The grid size can be changed for level 3 as an added function.

## Home Screen – Grade Selection

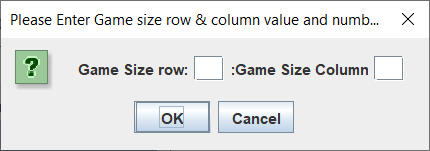
The user interface will be opened to select the Grade level to initiate the game. In the next window, the grid size can be initiated by giving the row and column values.



**Figure 7: Grade K2 - Home Screen**

## GradeK2 – Initiate the Grid Size

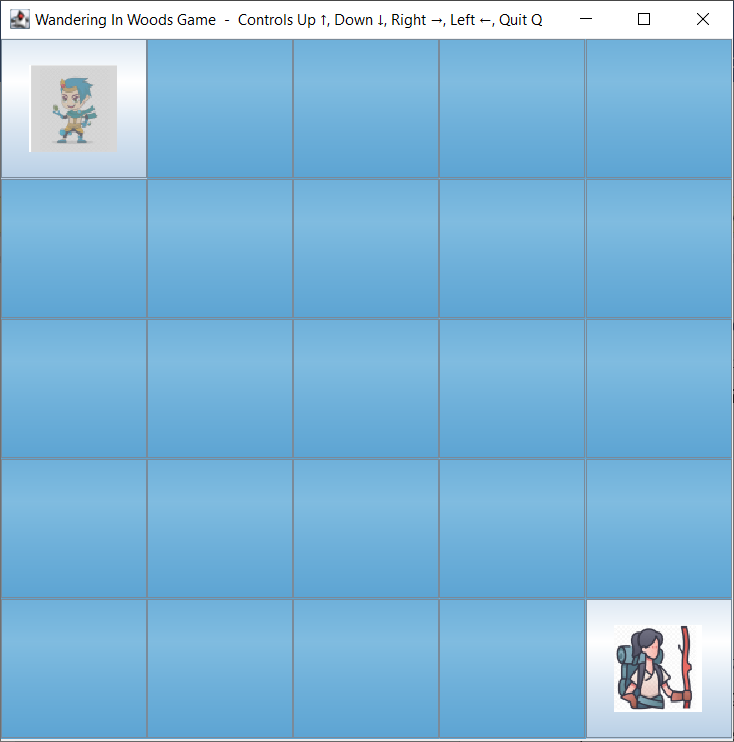
The grid can be initiated with the row and column values given in this window. The GradeK2 can be given as a square box. Once the size has been, the initial game window can be opened.



**Figure 8: Grade K2 - Initiate the Grid Size**

## GradK2 – Game window

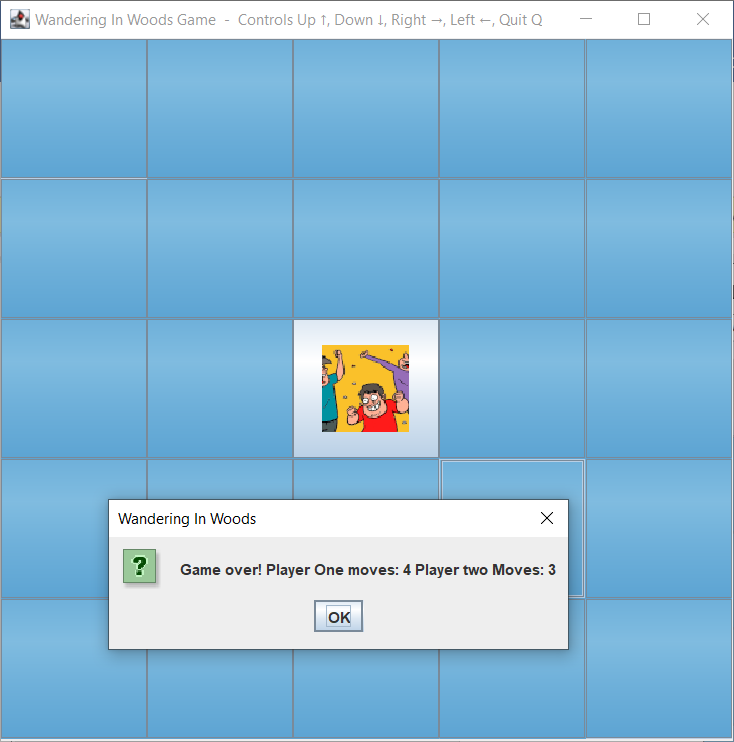
The GradK2 level can be a square window. At this level, there will two players and their positions will be assigned to the diagonal corner of the window. The user can use the keyboard keys to move the players into the grid windows.



**Figure 9: Grade K2 - Game Window**

## GradeK2 – Two players meet

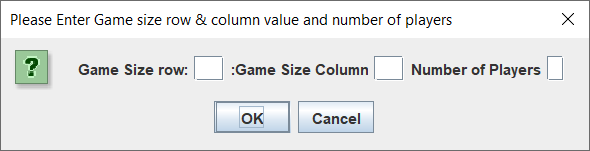
The user can move the players upward, downward, left, and right by using the keyboard. When two players meet in a grid, then the smiley will be displayed and music will be started.



**Figure 10: GradeK2 – Two players meet**

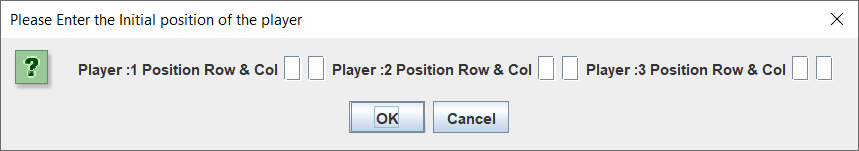
## Grade3-5 – Initiate the Grid Size

The grid can be initiated with the row and column values given in this window. The Grade3-5 can be given as a rectangle box.



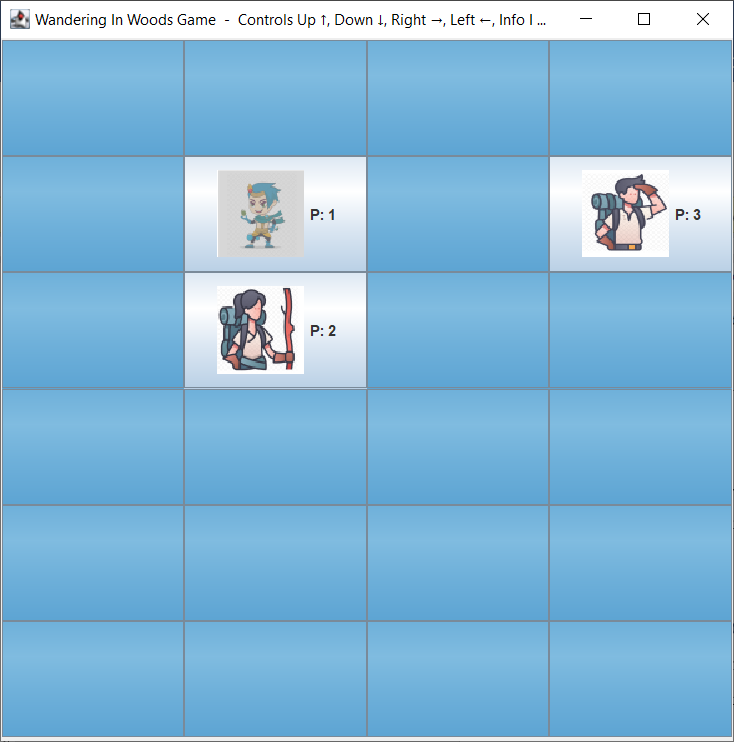
**Figure 11: Grade3-5 – Initiate the Grid Size**

Once the size and number of players have been given, then the user can give the position of each player. The number of players can be selected from the previous window.



## Grad3-5 – Game window

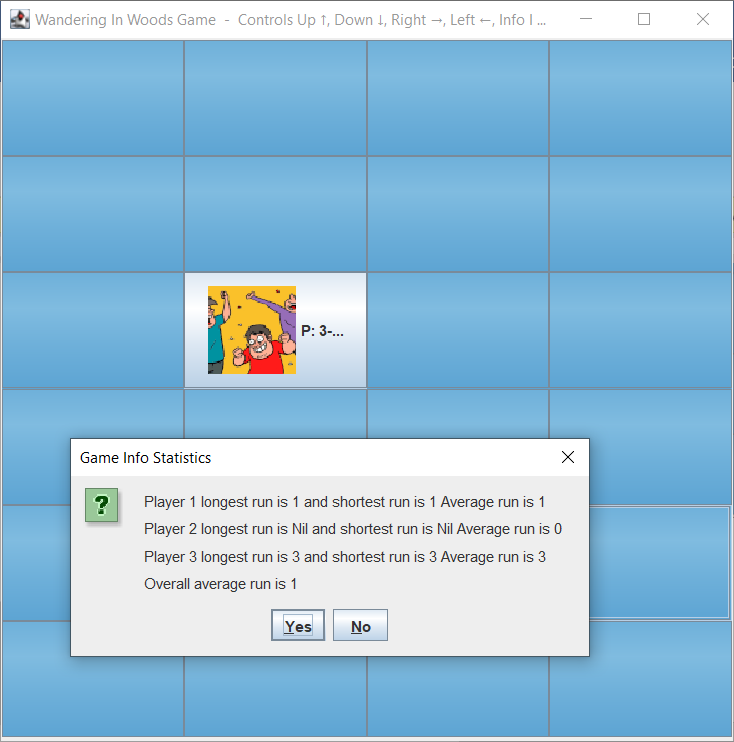
The Grade3-5 level can be a rectangle window. At this level, the players will be placed in the given position. The players can be moved using the keyboard buttons.



**Figure 12: Grad3-5 – Game window**

## Grade3-5 – All players meet

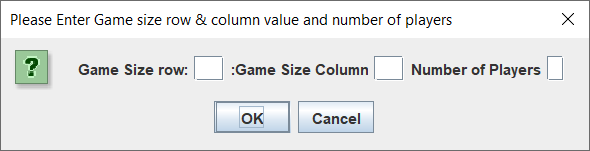
The user can move the players upward, downward, left, and right by using the keyboard. When all the players meet in a grid, then the smiley will be displayed and music will be started and the statistics can be displayed.



**Figure 13:Grade3-5 – All players meet**

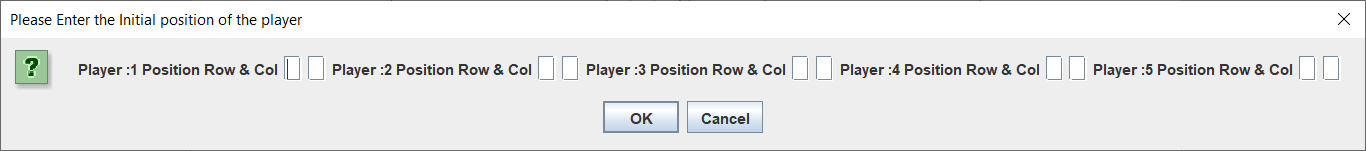
## Grade 6-8 – Initiate the Grid Size

The grid can be initiated with the row and column values given in this window. The Grade 6-8 can be given as a rectangle box.



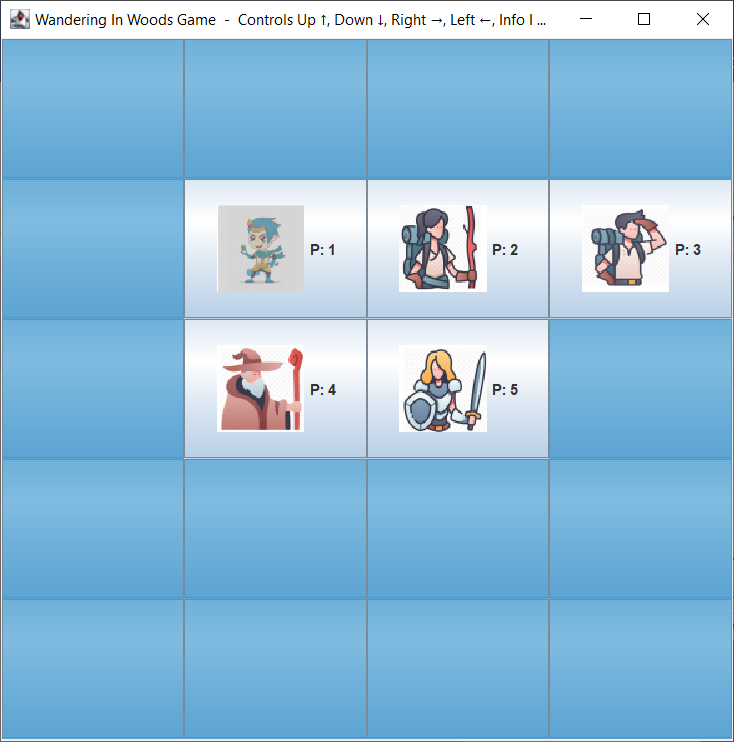
**Figure 14: Grade 6-8 – Initiate the Grid Size**

Once the size and number of players have been given, then the user can give the position of each player. The number of players can be selected from the previous window.



## Grad 6-8 – Game window

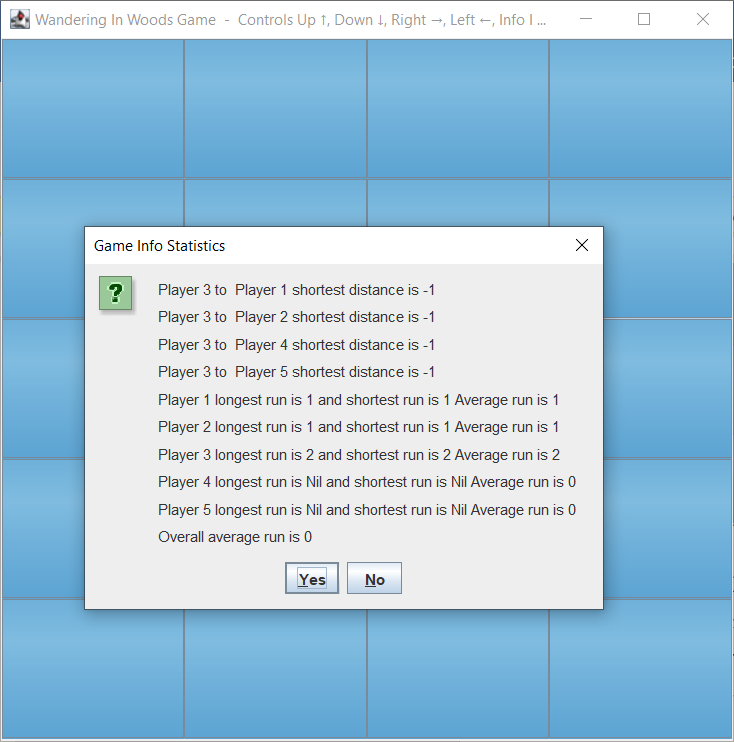
The Grad 6-8 level can be a rectangle window. At this level, the players will be placed in the given position. The players can be moved using the keyboard buttons.



**Figure 15: Grade 6-8 – Game window**

## Grade 6-8 – All players meet

The user can move the players upward, downward, left, and right by using the keyboard. When all the players meet in a grid, then the smiley will be displayed and music will be started and the statistics can be displayed.



**Figure 16: Grade 6-8 – All players meet**

# Testing Strategy

The crucial step in the software development process is testing. By examining the system requirements and changes made during the development stage, applying the precise testing models will enable the identification of flaws in the intended system. A well-thought-out testing approach will shorten software development life cycles and enable the delivery of products on schedule.

Software testing is the process of assessing the generated product to make sure it complies with all specifications and follows the suggested structure to get the desired results. The capacity of the programmed to achieve the desired results and detect errors or a failure to adhere to the specifications is its efficacy. The process of assessing and managing the software's quality is known as a test strategy (Khan, Sharma, & Srivastava, 2010). A test plan is created with the testing purpose in mind. For the development of the Wandering the wood game we have selected four test cases such as Integration Testing, Performance Testing, Usability Testing, User Acceptance Testing.

## Integration Testing

The process of verifying the interaction between the various components is known as integration testing. Any individual module can function well while it is designed, but when it is combined with another module to execute the finished programme, it may not function well. There is also a potential that the software won't act as intended and won't produce the desired results. The software's capacity to operate faultlessly will be ensured by the integration testing. The intricate flaws in the integrated software components can be identified with the aid of integration testing (Weikle, Lam, & Kirkpatrick, 2019). Compared to running an end-to-end test, this test is quick and effective. Integrity testing will lower software failure rates.

Software integration testing methods include big bang integration, gradual integration, top-down integration, bottom-up integration, and sandwich integration. We conducted Top-down integration testing for this game development module. Every testing step in this method is started from the architecture's topmost module.

## Performance Testing

The effectiveness of the software that has been produced is evaluated through performance testing. It entails carrying out steps such system analysis, testing method selection, test execution, result analysis, and tweaking the incorrect result to find the issue. The outcome is examined to determine the software's efficiency and performance factor. Performance in our game of "wandering in the woods" is evaluated using factors like accuracy and speed. The produced software's performance was evaluated for accuracy and speed.

## Usability Testing

Usability testing is a technique used to evaluate a product by analyzing the user interface and user-centered interactions. The purpose of this testing is to pinpoint usability issues and gather both qualitative and quantitative information about the product to gauge customer satisfaction. In this testing, participants are given a specific task to do, and the results are examined to gauge how user-friendly the system is (Sergeev & Kaklauskas, 2018). This evaluation will show what modifications to the system are required to increase its effectiveness. To make sure the system's usability goals are met, the system's performance is examined. By assuming the role of a user, the user interface of the gaming environment is tested on all three levels.

## Acceptance Test

During the development process, an acceptance test is carried out to determine whether the created system satisfies all user requirements. It is a step in the software development process' quality assurance procedures. Before the software is deployed, this test must be performed. The product's performance under the simulation will be examined during the acceptance test. It guarantees the software product's stability. The following steps are used to perform it: planning, testing, obtaining the results, comparing the data, and releasing it. The user acceptability test, an end user test that evaluates how the programmed is utilized by the target audience, is used to test the software in the game Wandering in the Woods.

# References

Cardoso, B. M., & Romão, T. (2021). 2D Game Editor On-line A Cloud Computing. *Perspective on Game Edition* .

Czauderna, A., & Guardiola, E. (2019). The gameplay loop methodology as a tool for educational game design. *Electronic Journal of e-Learning* *, 17* (3), 207-221.

Khan, R., Sharma, P., & Srivastava, A. (2010). Improving the Quality of Gaming Apps After Testing using Genetic Algorithm.

Sergeev, S., & Kaklauskas, A. (2018). Cybersport: Stages and content of usability testing of gaming interfaces and environments. *MATEC Web of Conferences.* *245*, p. 04017. EDP Sciences.

Weikle, D. A., Lam, M. O., & Kirkpatrick, M. S. (2019). Automating systems course unit and integration testing: Experience report. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* , (pp. 565-570).