**Wandering in the Woods: A Simulation Game for K-2 & 3-5 Education**

**Software Engineering A Practitioner’s Approach 9th Edition Supplement**

[**Project Group 2**](https://lewisuniversity.blackboard.com/ultra/courses/_203825_1/groups/enrollments/group-space?groupId=_123344_1&groupSpaceView=content)

**SU24-CPSC-60500-001**

**Fadi Wedyan**

**Table of Contents**

1.0 Introduction

1.1 Purpose

1.2 The Wandering in the Woods Game

2.0 Process Model

2.1 Iterative Development

2.2 Agile Principles

2.3 Continuous Feedback and Improvement

2.4 Risk Management

3.0 Use Cases

3.1 Use Case 1: Navigate the Grid (Grades K-2)

3.2 Use Case 2: Customize the Grid (Grades 3-5)

3.3 Use Case 3: Group Movement (Grades 3-5)

4.0 UML Model

4.1 Use Case Diagram

4.2 Class Diagram

5.0 UI Mock-up

6.0 Testing Strategy

6.1 Unit Testing

6.2 Integration Testing

6.3 Usability Testing

6.4 Validation Testing

7.0 How it Works  
     7.1 Step 1: Running the Launcher  
     7.2 Step 2: Home Page Navigation  
     7.3 Step 3: K-2 Grade Game Functionality  
     7.4 Step 4: Game Over and Messages  
     7.5 Step 5: Back, Reset, and Home Buttons  
     7.6 Step 6: Total Moves Display  
     7.7 Step 7: 3-5 Grade Game Complexity  
     7.8 Step 8: Final Output Messages

8.0 Code Explanation  
    8.1 WanderingInWoodsK2 (K-2 Version)  
     8.2 WanderingInWoods35 (Grades 3-5 Version)  
    8.3 WanderingInWoodsLauncher

9.0 Conclusion

**1.0 Introduction**

**1.1 Purpose**

The "Wandering in the Woods" simulation game serves as an educational tool to introduce K-2 students to fundamental concepts in computation, computational thinking, mathematics, and computer science. Through a playful and immersive environment, students guide characters through a virtual forest on a grid, engaging with entertaining scenarios that intuitively teach algorithmic thinking, problem-solving, and data representation. This game designed to make learning as well as these essential skills both fun & accessible, ensuring students could easily relate to the content and feel confident as they advance through different grade levels.

Developed in Java using software NetBeans IDE, where the project emphasizes creating robust, scalable, & maintainable software. We choose Java for its platform independence and suitability for educational software, while NetBeans offers comprehensive tools for coding, debugging, and testing. Following an iterative and agile development model, the project is meticulously planned to meet both educational and technical standards. This approach ensures the game provides a smooth and engaging user experience, making it a valuable long-term resource for teaching computational and mathematical concepts.

**1.2 The Wandering in the Woods Game**

"Wandering in the Woods" is an educational simulation game designed to help students navigate characters through a dense forest represented by a rectangular grid, providing a unique way to learn essential computational and mathematical concepts. The game is structured into two distinct stages, each tailored to different grade levels—K-2 and 3-5—offering increasing levels of complexity as students progress. In the early stage for grades K-2, the game introduces basic concepts in a simple, engaging format where students guide characters through a square grid, focusing on foundational ideas like counting and spatial awareness. As students advance to grades 3-5, the game evolves, allowing them to customize the grid's size and shape, and manage multiple characters, thereby introducing more complex challenges that involve strategy, problem-solving, and data interpretation. The consistent visual and interaction design across both stages ensures that students experience a seamless transition as they advance, making it easier for them to grasp increasingly complex concepts without being overwhelmed. This thoughtful progression not only reinforces their learning but also builds their confidence in applying computational and mathematical principles, making "Wandering in the Woods" a powerful educational tool that grows with the learner.

**2.0 Process Model**

In the "Wandering in the Woods" project, we employed **iterative and also an agile process model** to guide the development. This approach will be chosen for its flexibility & adaptability, which are crucial for a project aimed at creating an educational tool that meets both technical and pedagogical requirements.

**Iterative and Agile Process Model**

**2.1 Iterative Development**

The iterative aspect of the process model involves breaking down the project into smaller, manageable increments, or iterations, each representing a complete cycle of development. In each iteration, a portion of the game's features is designed, developed, and tested, ensuring that the project evolves gradually and continuously improves based on feedback. This method allows for early detection of issues, quick incorporation of new ideas, and refinement of existing features, which is particularly important in educational software development where user feedback (from both students and educators) can significantly influence the final product.

**2.2 Agile Principles**

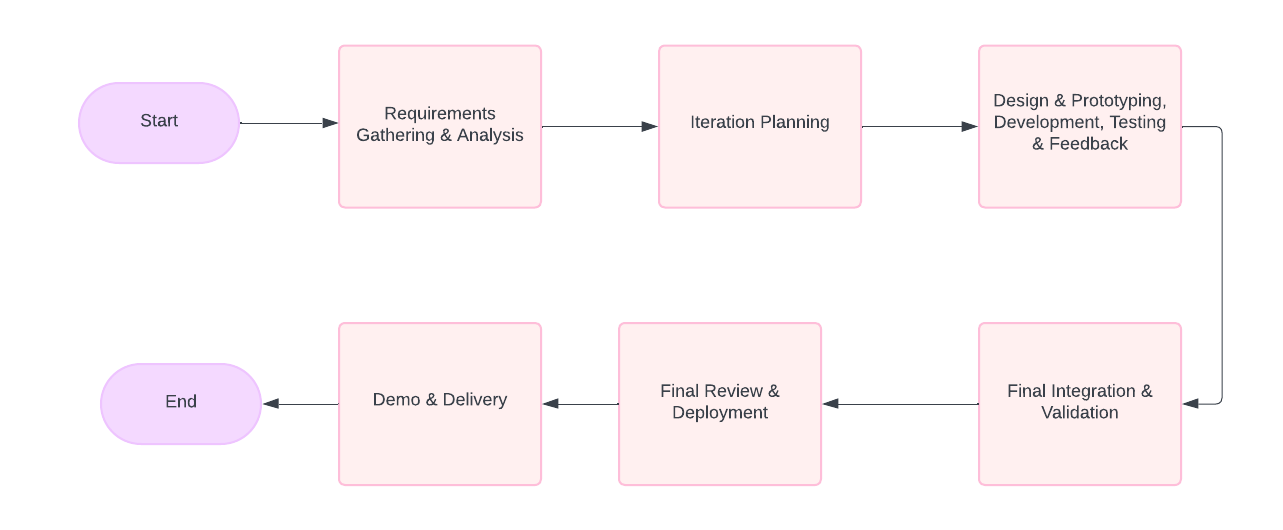
The agile component of the process model emphasizes flexibility, collaboration, and customer (or end-user) involvement. Agile principles prioritize working software, frequent delivery of small, functional segments of the product, and responsiveness to change. In context of our project, agile methods enabled the development team to adapt to new insights or requirements that emerged during the development process. Example, based on the early testing and feedback from educators, adjustments could be made to the user interface, the complexity of tasks, or the instructional design elements embedded in the game.

**2.3 Continuous Feedback and Improvement**

The combination of iterative and agile practices ensures that the project is continuously evolving in response to user needs and technical challenges. Each iteration is followed by testing and evaluation, with feedback loops allowing for ongoing improvements. This approach aligns well with the educational goals of the project, ensuring that the final product is not only technically sound but also effective as a teaching tool.

**2.4 Risk Management**

The iterative and agile process model also supports risk management by allowing the team to identify and address potential issues early in the development cycle. By developing the project in small, incremental steps, the team can avoid the pitfalls of overcommitting to a particular direction without sufficient testing or validation.



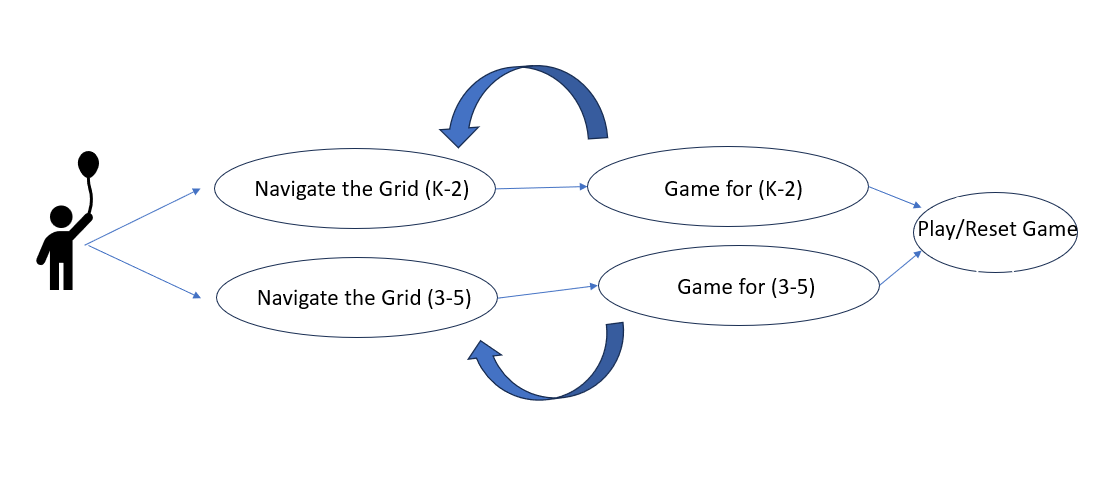
**3.0 Use Cases**

The following use cases define the core functionality of the "Wandering in the Woods" game:

**3.1 Use Case 1: Navigate the Grid (Grades K-2)**

**Primary Actor:**

Student  
**Preconditions:** The game has started, and characters are placed on diagonally opposite corners of a square grid.   
**Description:** Students will observe characters wandering randomly within the grid. Music plays as the characters move, and when they meet, a celebratory graphic and statistics are displayed.  
**Acceptance Criteria:** The characters wander randomly, meet, and the game resets after displaying statistics.



**3.2 Use Case 2: Customize the Grid (Grades 3-5)**

**Primary Actor:**

Student  
**Preconditions:** The game has started, and students can access customization options.  
**Description:** Students can customize the grid's size, shape (square or rectangular), and the number of characters (2, 3, or 4). They can place the characters at any grid location. The game then runs, tracking various statistics.   
**Acceptance Criteria:** The game correctly implements the customized grid and character settings, and displays the appropriate statistics.

**3.3 Use Case 3: Group Movement (Grades 3-5)**

**Primary Actor:**

Student  
**Preconditions:** The game is running with 3 or 4 characters on the grid.   
**Description:** When two characters meet, they move together until they find the next character. This continues until all characters are together.   
**Acceptance Criteria:** The characters move as a group after meeting, and the game tracks this behavior correctly.

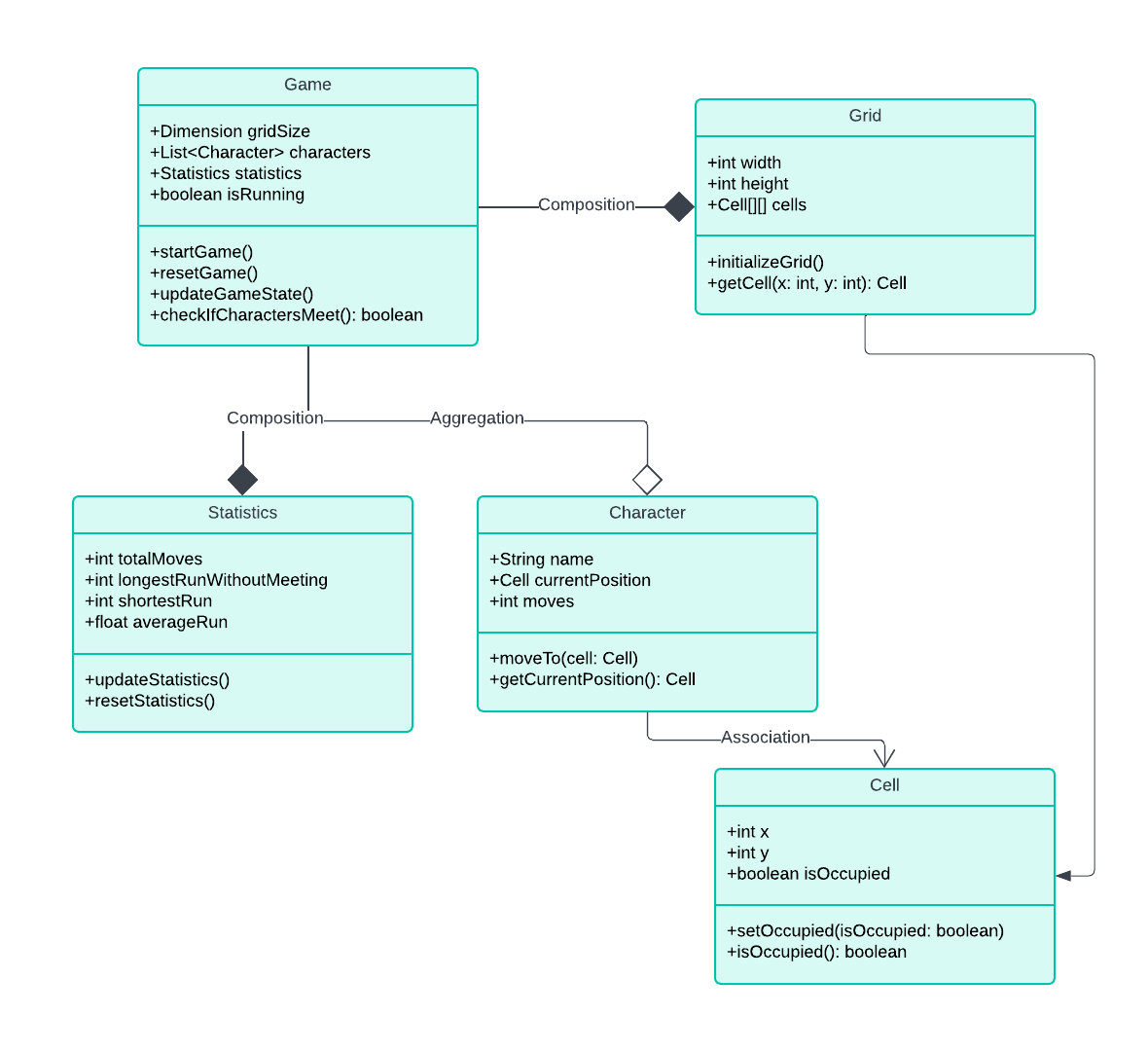
**4.0 UML Model**

**4.1 Use Case Diagram**

In the "Wandering in the Woods" project, the use case diagram represents the interactions between the student (as the primary actor) and the game system, illustrating the progression from simple to more complex activities as students move from grade levels K-2 to 3-5. For students in grades K-2, the game focuses on basic interactions where students guide two characters through a square grid, with the characters moving randomly until they meet, reinforcing foundational concepts like counting and spatial awareness. As students progress to grades 3-5, the complexity of interactions increases; they can now customize the grid's size and shape and manage multiple characters. This allows them to engage in more strategic decision-making and introduces them to basic problem-solving and data analysis concepts. The use case diagram highlights how the game transitions from simple wandering and meeting tasks to more sophisticated scenarios where students make choices that directly affect outcomes, all within a consistent and intuitive interface that supports their learning journey. This progression ensures that as students grow, they build on their previous knowledge, gaining confidence and deepening their understanding of key computational and mathematical principles.

**4.2 Class Diagram**

The class diagram provides a detailed view of the system’s architecture, including classes for the Grid, Characters, and Statistics, and their relationships.



**5.0 UI Mock-up**

The user interface (UI) mock-up displays a simple, intuitive design suitable for K-8 students. The design includes large, colorful buttons, clear text, and engaging visual elements to guide students through the game.

**6.0 Testing Strategy**

**6.1 Unit Testing**

Unit testing focused on verifying the functionality of individual components within the game's core classes. For the K-2 version, we tested the methods responsible for character movement, ensuring that characters moved randomly within the grid without crossing boundaries. Each character's movement was tracked, and the move count was updated correctly. The methods responsible for resetting the game, such as resetGame() and moveCharacter(), were also tested to confirm they returned the characters to their starting positions and reset the move count to zero​(WanderingInWoodsK2).

In the 3-5 version, unit tests ensured that multiple characters could move independently and within the constraints of a customizable grid. We also tested the loading of character images and the proper updating of move counts for each character. Special attention was given to testing the logic that checks if all characters have met on the grid, a key feature in this version​(WanderingInWoods35).

**6.2 Integration Testing**

Integration testing was conducted incrementally, allowing us to verify that new features worked seamlessly with existing components. For instance, after adding features such as the back, reset, and home buttons, we tested their functionality within the broader game environment. The integration tests confirmed that these buttons correctly interacted with the main game loop and UI, allowing players to return to the main menu, reset the game, or navigate home without disrupting the game state​(WanderingInWoodsK2)​(WanderingInWoods35).

The game’s timer and character update mechanisms were tested to ensure that they synchronized correctly, especially in the 3-5 version where multiple characters move simultaneously. This testing phase ensured that all components, including the graphical interface and game logic, worked together smoothly​(WanderingInWoodsLauncher).

**6.3 Usability Testing**

Usability testing was a crucial part of our strategy, focusing on how well young students could interact with the game. We used personas representing typical K-8 students to test the game's ease of use, including the clarity of the instructions, the intuitiveness of the UI, and the overall engagement level. The tests revealed that the large, colorful buttons and clear labels were effective in guiding students through the game. Feedback from these tests led to minor adjustments in the layout and positioning of buttons to enhance the user experience​(WanderingInWoodsLauncher).

For instance, in the K-2 version, the placement and size of the back, reset, and home buttons were tested to ensure they were easily accessible to young children without interfering with the gameplay. Similarly, in the 3-5 version, the customization options for the grid and character management were tested for ease of use, ensuring that students could easily understand and apply these features​(WanderingInWoodsK2)​(WanderingInWoods35).

**6.4 Validation Testing**

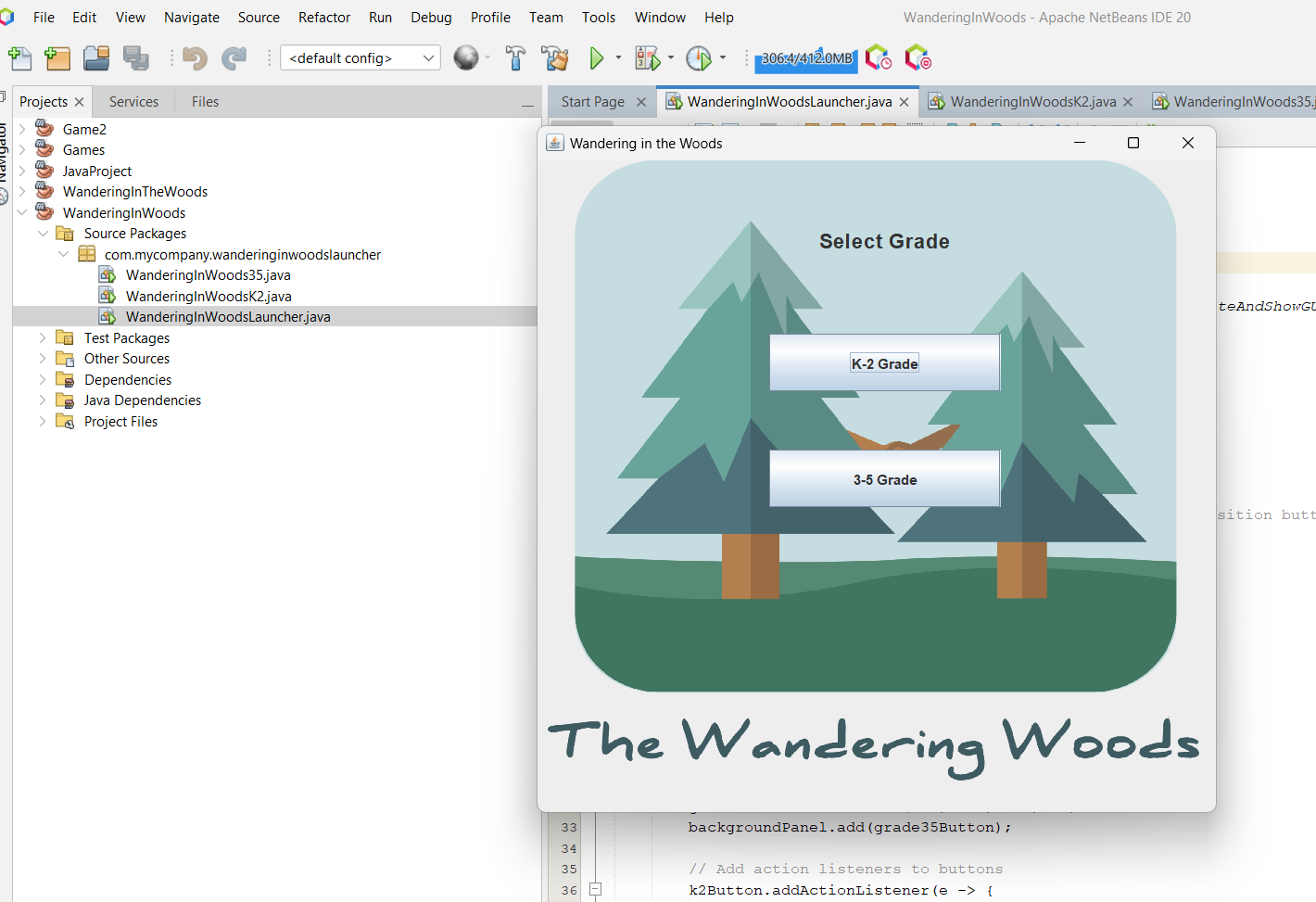
Validation testing was conducted to ensure that the final product met all the specified functional and non-functional requirements. This phase involved verifying that the game’s educational objectives were achieved and that the software performed reliably across different scenarios. For example, we validated that the K-2 version effectively introduced basic computational concepts like counting and spatial awareness through simple, engaging gameplay. In contrast, the 3-5 version was validated to ensure it provided more complex challenges, such as grid customization and strategic movement, reinforcing higher-level computational thinking​(WanderingInWoodsK2)​(WanderingInWoods35).

Furthermore, the game's overall stability was tested across various platforms and screen resolutions to confirm that it provided a consistent experience regardless of the environment. The validation tests confirmed that the game not only met its educational goals but also provided a robust and user-friendly experience for both students and educators.

**7.0 How it Works:**

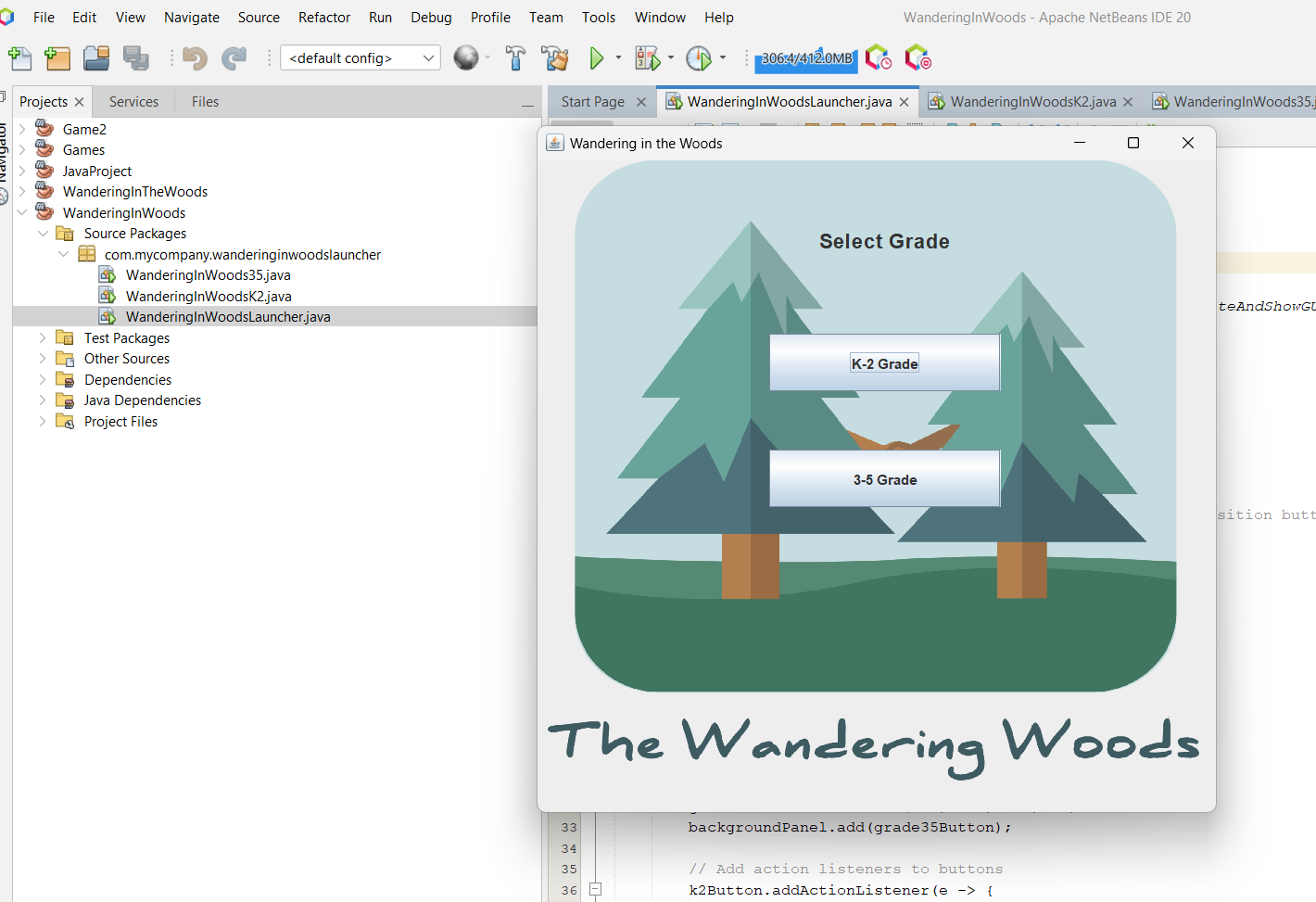
**7.1 Step 1:**

First, we will run the main class of WanderingInWoodsLauncher.java in Netbeans.



**7.2 Step 2:**

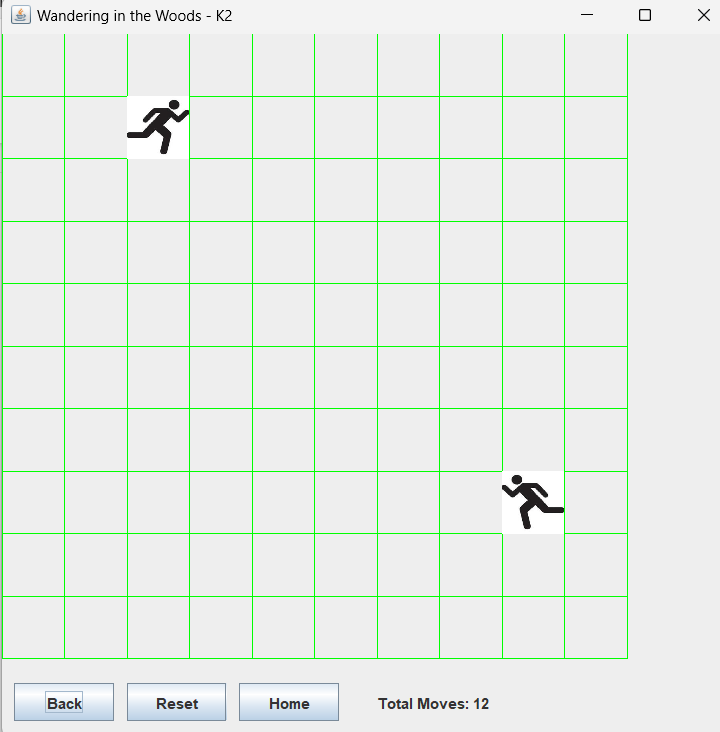
Home page of the project. Here we integrated two buttons K-2 Grade & #-5 Grade along with a background and a Header on the top.



**7.3 Step 3:**

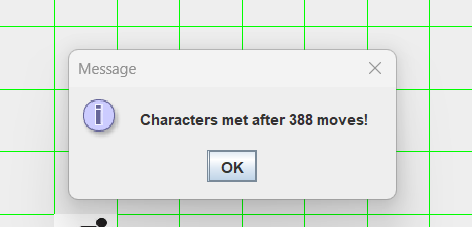
If we select K-2 Grade the game will start running here the characters will move forward, up and down when the characters collate then the game will get end.

Here on the bottom, we added three buttons: Back, Reset, Home each having the working functionality executed in the code.



**7.4 Step 4:**

When the game over then it will prompt a message.



**7.5 Step 5:**

When selected Back it will back to back Page.

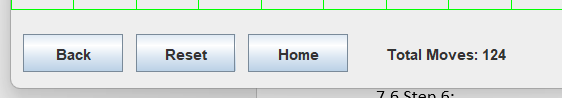
When selected reset the game will reset.

When selected Home it will go to home page.



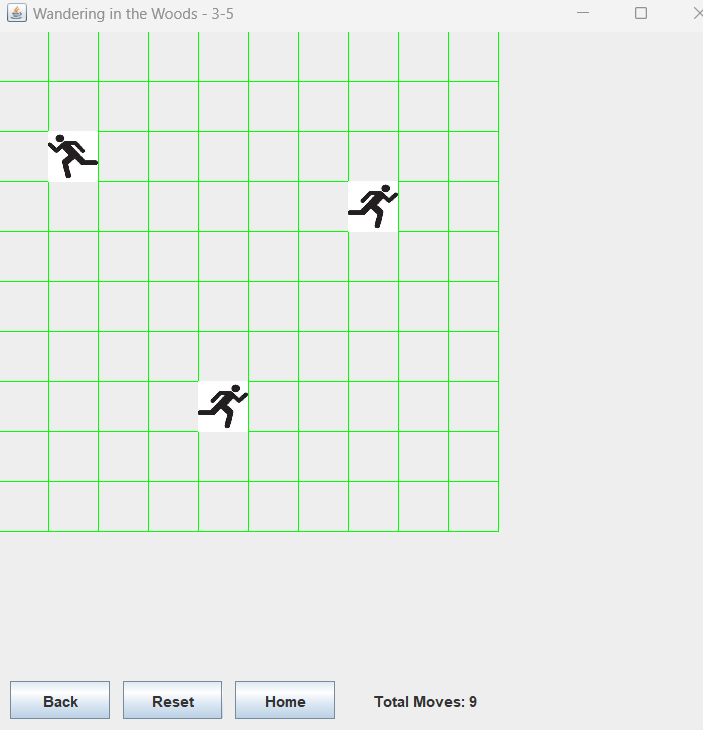
**7.6 Step 6:**

There we integrated a Total moves column which shows how many moves the characters moved.



**7.7 Step 7:**

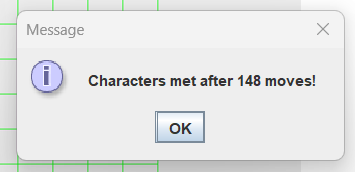
Now when entered into 3-5 Grade the game becomes more complex as here, we used three characters to meet.



And here we have integrated all the three buttons and a total move column.

**7.8 Step 8;**

Finally we also integrated the message which shows the output.



**8.0 Code Explanation:**

**8.1 WanderingInWoodsK2 (K-2 Version):**

This WanderingInWoodsK2 class Java program which we designed as an educational game for kindergarten to 2nd-grade students. This class extends the JPanel and implements ActionListener to manage the game logic and user interface. The game we designed a simple 10x10 grid where two characters start at opposite corners of the game. The characters move randomly within the grid, and the goal is for them to meet. The game includes a Timer that triggers the movement of the characters at set intervals, and a paintComponent method is overridden to handle the drawing of the grid and characters. UI includes buttons for "Back," "Reset," & "Home," which allowing users to navigate between screens or reset the game. The movesLabel tracks&displays the number of moves taken by the characters until they meet​(WanderingInWoodsK2).

In terms of functionality, the game primarily teaches the spatial awareness & basic counting as students observe the characters’ movements. The logic for character movement ensures that characters stay within the grid boundaries. When the characters meet, a sound can be played (though the code for sound is left as a placeholder), and a dialog box informs the player of the number of moves taken. The game then resets, encouraging repeated play to reinforce learning. This class effectively uses Java Swing components to create an interactive and visually engaging educational experience for young learners​(WanderingInWoodsK2).

**8.2 WanderingInWoods35 (Grades 3-5 Version):**

The WanderingInWoods35 class is designed for older students in grades 3-5 and introduces more complexity compared to the K-2 version. This class also extends JPanel and is responsible for managing a more customizable game environment where the grid size and the number of characters can vary. The grid dimensions are specified by gridSizeX and gridSizeY, and the number of characters is passed to the constructor. Each character is represented by a Point object, and a Timer controls the movement of all characters on the grid. The program handles character movement with additional logic to ensure they stay within the grid boundaries, and it checks if all characters have met, in which case the game displays a congratulatory message and resets​(WanderingInWoods35).

This version includes a more complex interaction with the user interface, allowing students to customize their experience by choosing different grid sizes and character counts. The game enhances strategic thinking and problem-solving as students observe and predict the characters’ paths. The movement of multiple characters and the requirement for all to meet introduces a more challenging aspect, suitable for the cognitive level of older students. The game also features basic error handling for loading character images, ensuring that the game can continue even if an image fails to load, which adds to its robustness and usability​(WanderingInWoods35).

**8.3** **WanderingInWoodsLauncher:**

The WanderingInWoodsLauncher class serves as the main entry point for the game application, providing a simple launcher interface that allows users to choose between the K-2 version and the 3-5 version of the game. This class uses Java Swing to create a user-friendly window with buttons for launching either version of the game. When a user selects a version, the appropriate game window opens, and the launcher window closes. The use of SwingUtilities.invokeLater() ensures that the GUI operations are performed on the Event Dispatch Thread, maintaining thread safety for the user interface​(WanderingInWoodsLauncher).

This launcher acts as a central hub for the application, offering a straightforward way for teachers or students to select the appropriate game version based on the student's grade level. By separating the game launcher from the actual game logic, the code remains modular and easier to maintain or extend. The launcher enhances the user experience by providing an organized starting point, which is particularly important in an educational setting where ease of use can significantly impact engagement and learning outcomes​(WanderingInWoodsLauncher)002E

**9.0 Conclusion**

Finally, the project *"Wandering in the Woods"* effectively will harness the principles of software engineering which is to create an engaging & educational game that introduces Kindergarden-5 students to essential computational concepts. With this through a combination of playful interaction & thoughtfully designed challenges, the game will succeed in making complex ideas like algorithmic thinking also a problem-solving accessible to kids. The iterative & agile development process allowed the team to refine the game continuously, which will ensuring that it meets both the technical and the educational standards.

Here, testing is main in ensuring the game's functionality. Rigorous unit testing confirmed the functionality of individual components, while the integration testing ensured seamless interaction between the different features. Usability testing provided valuable insights into how students engage with the game, leading to enhancements that improved the user experience. Finally, validation testing confirmed that the game met all functional requirements and educational objectives, making it a reliable tool for teaching computational thinking.

*"Wandering in the Woods"* stands as a testament to the power of well-executed software development and its potential to enhance educational experiences. By providing a robust and scalable learning tool, the project not only achieves its immediate goals but also lays the groundwork for future educational innovations. The game is well-positioned to be a long-term resource that grows with students, reinforcing their understanding of computational and mathematical principles as they progress through their education.