**PROJECT- WANDERING IN THE WOODS**

**By-** Vamshi Kamal Eligeti

Manideep Reddy Karnati

Jasmine Kaur

Dhruv Harshadkumar Nayee

Jasmine Shaik

**Introduction**

Mathematical concepts are an unavoidable part of the education system and it is also considered among the difficult subjects for students. Lack of knowledge is considered one of the vital causes behind the failure of students in mathematics. However, the computational weakness is not much evaluated because of its fundamental nature and less discussed among scholars. Computation weakness is that key reason that drags down a student’s performance in mathematical conception. Computation appears to be simple calculation, but it gets complex with its growing level. Likewise, it needs capacity within a student to make calculations that cannot be achieved overnight. The purpose of this study is to provide a platform to the students to practice mathematical calculation through simulation by game. Fulfillment of purpose can be done by integrating an offline interface of a game in the process of mathematical simulation by students. Once a digital platform is created for any purpose, it provides the flexibility to develop features as per the real time requirement.

**Overview of Project**

The project shed lights on the current development of games where various features of mathematical calculation will be incorporated. The project is going on with an intention to present a **blueprint** of an actual offline application that can give real time assistance in the student learning process. “*Wandering in the Woods*” is the name of this project that sheds light on its features that I am going to incorporate in the form of a game. A game cannot be developed practically without proper evaluation. Therefore, I have created multiple figures of application on the basis of different concepts in order to estimate appropriate software architecture. Hence, these forms of figures are Use case diagram, deployment diagram, class diagram, state diagram and activity diagram I have already developed.

Creation of computer simulation is happening by dividing three sections on this interface of the game. Computer simulation refers to a game played by the user to simulate mathematical concepts. In other words, the simulation makes a student of K-8 (Grade 8) expertise in various concepts of mathematics and computer science. However, the aforementioned facts of three sections outline the division of students (players) on the basis of grades. I have considered it Grade 1, Grade 2 and Grade 3 where Grade 1 represents the simplest form of mathematical simulation. The visualization of Grade 1 interface has been kept very simple in its visualization so that students of Grade 1 -2 can handle it. Likewise, I will give a comparatively little complex version by adding up additional challenges to the second piece. Second piece of the project is made for the students of Grade 3-5 so that one can make the choices while playing this game. The third piece of the project will be the most complicated one where the challenge would be actually difficult to overcome by the students. I have estimated this complex section of the project appropriate for the students belonging to Grade 6 to Grade 8.

**Process Model**

Process model can be considered as a means of visualization that outlines the internal process of business. The model is used for better management of processes and exploiting a method to optimize it. The nature of the model sheds light on the constant development of process and therefore, it can be considered as constant development activities. An evolutionary process framework has been applied to the design for the game. Hence, an iterative approach is adopted to integrate the evolutionary model with the whole software development lifecycle.

However, I have considered the incremental factor of the feature of the game while applying the evolutionary model. Consequently, the approach I have adopted for the product has declined the scope of risk to an optimum level by matching up demand of target customers. In addition, the incorporation of the model has reduced the cost of the project abruptly. After the use of this model, an incremental enhancement has been achieved very fast. As I have described the project into three pieces, the evolutionary method seems to be very suitable to accomplish the building of this product.

Each piece of the project has a level of simulation whose range starts from simple ways of simulation to very complex methods of mathematical simulation. Adoption of an evolutionary model allowed me to rank each piece or section of a project in the respective simple, moderate and complex levels. The modular pattern of the product allows the student to choose the level of simulation according to one’s preference and need of learning. Subsequently, students with multiple competencies of mathematical knowledge can interact with the interface of game and simulate on it.

**Initial Tentative Requirement**

**Specification**

**Creation**

**Validation**

**Starting Version**

**Intermediate Version**

**Final Version**

**Use Case**

Use cases have been applied with an intention to outline the pattern of functions performed by the user on a system. I tried to explore the viewpoint of a user during the interactions with the product on a system that further defines the overall behavior of a student. I considered the main objective of a student as the initial point for creating a use case. Consequently, an accomplishment of the aforementioned objective is considered as the end point of the use case. I am applying the concept of starting point and end point described in the prior statement to every use case.

In the meantime, I realized that brainstorming to forecast the behavior of students is possible after portraying the use case because it expresses the situation clearly and effectively. Use case facilitated me with multiple objectives enlisted for calculating the expenditure on the development of this game.

Likewise, the list of objectives retrieved from the use case also introduces the level of complexity the system will have. It has been observed that process boundaries play a crucial role when a programmer needs to explore the functions from use cases. An additional actor can be added to the outside of boundaries that is interpreted as supporting actor in the use case diagram. Hence, the functions can be negotiated during the development of the application.

**Use Case 1: Initiating Game**

|  |  |
| --- | --- |
| **Identification Number** | 1 |
| **Evaluation** | Students would be responsible for uploading the game and it will be started by the user. |
| **Actor** | Student |
| **Before Condition** | Launching the game by student will be starting after its installation. |
| **Event Flows** | 1. Click on the screen to initiate the game by students 2. Choose the game option 3. Game will initiate |
| **After Condition** | Once the game is loaded in the memory, the system starts executing it. |

**Use Case 2: Play Game**

|  |  |
| --- | --- |
| **Identification Number** | 2 |
| **Evaluation** | In order to make the movements of the user on a grid, the student plays the game for navigating to the grid. |
| **Actor** | Student |
| **Before Condition** | Student chooses a play options from the menu |
| **Event Flows** | 1. Students click the game present on the screen. 2. Game started by a student. 3. Students would click the option of play. 4. Game screen has been exhibited on a system with a display of a grid. 5. Students will initiate play. |
| **After Condition** | A grid page is exhibited on the system to the student. |
| **Addition of Use Case** | Select the stage |

**Use Case 3: Select Stage**

|  |  |
| --- | --- |
| **Identification Number** | **3** |
| **Evaluation** | Selection of stages will take place by student in this use case because of availability of different stages. |
| **Actor** | Student |
| **Before Condition** | The selection of stages is mandatory for students. |
| **Event Flow** | 1. Students click the game reflected on the screen of the system. 2. Student starts the game. 3. Students choose options from different stages. 4. Presses play option 5. Exhibition of grid on the game screen in the system. 6. Initiation of game. |
| **After Condition** | Exhibition of grid page to the student by a system. |
| **Add up Use Case** | Select Stage |

**Use Case 4: Visualize Statistics**

|  |  |
| --- | --- |
| **Identification Number** | 4 |
| **Evaluation** | In the use case, a statistics is visualized by a student which is generated due to game played. |
| **Actor** | Student |
| **Before Condition** | Students mandatorily play games. |
| **Event Flow** | 1. Student clicks game reflects on screen. 2. Game will start. 3. Options from stages are selected by students. 4. Pressing play options 5. Statistics will be displayed on the system. |
| **After Condition** | Exhibition of statistics is needed on a frequent basis. |

**Use Case 5: Replay Game**

|  |  |
| --- | --- |
| **Identification Number** | 5 |
| **Evaluation** | Game is played in a repeated manner multiple times by students. |
| **Actor** | Student |
| **Before Condition** | Games should be played by the student. |
| **Events Flow** | Student will   1. Click game available on screen 2. Initiate the game 3. Student chooses the stage option. 4. Press option of play 5. Exhibition of grid on game screen in the system 6. Initiate the play 7. Exhibit the statistics 8. Replay the game |
| **After Condition** | System should repeat the game several times on the request of the student. |

**Use Case 6: Exit Game**

|  |  |
| --- | --- |
| **Identification Number** | 6 |
| **Evaluation** | An exit will be made by the student to finish the game in the use case. |
| **Actor** | Student |
| **Before Condition** | Students should exist within the screen of the game. |
| **Flow of Event** | 1. Click the game screen 2. Initiate the game 3. Exit from the game by student |
| **After Condition** | System should exit the game that sequentially closes the game windows. |

**Use Case 7: Diagonal Movement**

|  |  |
| --- | --- |
| **Identification Number** | 8 |
| **Evaluation** | The movement will be made by the student diagonally in the opposite direction. |
| **Actor** | K-2 Student |
| **Before Condition** | Students need to maintain the group of K-2 players before initiating the game. |
| **Flow of Event** | 1. Click game reflects on screen. 2. Starting the game 3. Pressing play option 4. Exhibition of grid along with game screen. 5. A movement will be made toward the opposite direction by student 6. A happy graphic will be displayed when two players are together. 7. Reset the game by student. |
| **After Condition** | Calculation of movements and its presentation in a statistical form outlines graphics and music is played. |

**Use Case 8: Random Movement**

|  |  |
| --- | --- |
| **Identification Number** | 8 |
| **Evaluation** | The movement made in this use case will be random by student in terms of directions and placement of character happens anywhere across the grid. |
| **Actor** | K3-5 Grade students; K6-8 Grade student |
| **Before Condition** | Either K3-5 Grade students will be selected for the game or K6-8 students and further play the games. |
| **Flow of Event** | 1. Click on screen 2. Initiate the game 3. Pressing the play option 4. Grid will be displayed 5. Changing the size of grid 6. Movement happens in any direction by student 7. Exhibition of happy graphic 8. Reset of game |
| **After Condition** | Number of moves have been counted by the system and shown the outcome in statistical form. |

**Use Case 9: Changing the Grid**

|  |  |
| --- | --- |
| **Identification Number** | 9 |
| **Evaluation** | The grid size will be changed by a student in this use case. |
| **Actor** | Grade K3-5 Students, Grade K 6-8 student |
| **Before Condition** | Students belonging to Grade K3-5 and K6-8 will initiate the game. |
| **Flow of Event** | 1. Clicking the game on screen 2. Initiate the game by student 3. Pressing the play options by student 4. Exhibition of grid of game screen within a system 5. Changing the grid size by student 6. Student will take random direction 7. Happy graphic will be exhibited when two students meets with each other 8. Game reset by student |
| **After Condition** | Movements are counted by system and presented in statistical form. The size of the grid will be changed with the input of the student. |

**Use Case 10: Play Challenges**

|  |  |
| --- | --- |
| **Identification Number** | 10 |
| **Evaluation** | Challenges are given to the student from the system in this use case. |
| **Actor** | Grade K 6-8 Student |
| **Before Condition** | The students must belong to the level Grade K 6-8 and initiate the game. |
| **Event Flow** | 1. Click game by student 2. Initiate the game by student 3. Press the play option 4. Systems exhibit the grid of game screens within a system. 5. System facilitates new challenges to student 6. Movement will happen in any direction 7. With the meeting of two students, a graphic is displayed. 8. Reset the game |
| **After Condition** | Counting of movement should happen in a system and can be presented in statistical form. |

**Use Case 11: Testing the variation of Wandering Methods**

|  |  |
| --- | --- |
| **Identification Number** | 11 |
| **Evaluation** | Students will apply the various wandering methods for a short span of time. |
| **Actor** | Grade K 6-8 Students |
| **Before Condition** | The student should be selected from the Grade K 6-8 student group and initiate the game. |
| **Event Flow** | 1. Student clicks the game 2. Student will initiate the game 3. Pressing play options by student 4. Exhibited the grid of the game screen in the system. 5. Test wandering methods in a system 6. Random movement will make by student 7. With the meeting of two students, the happy graphic is displayed 8. System reset the game again |
| **After Condition** | Movement will be counted in the system and presented in the statistical form. Therefore, the system should represent wandering ways to student. |

**4. UML Diagrams**

**4.1 Use Case Diagram**

The main purpose of incorporating Use-case diagram is to showcase the behavior of end user with the concerned developed system in the premier stage. Subsequently, it helps the developer to estimate the activities of customer with the interface. A similar objective is accomplished in this study when I have drawn the user-case diagram for computer simulation. The diagrammatic representation of Use case diagram for Wandering in Woods is given below:

Student

Grade 2 Group Student

Grade 3-5 Group Students

Grade 6-8 Group Students

**4.2 Deployment Diagram**

Deployment diagram signifies the hardware and software component exists within a system that is used for interaction with any interface. Subsequently, the physical architecture of “*Wandering in Woods*” system has been drawn below with the support of deployment diagram:

Web Server

Apache

Browser

**HTTPS**

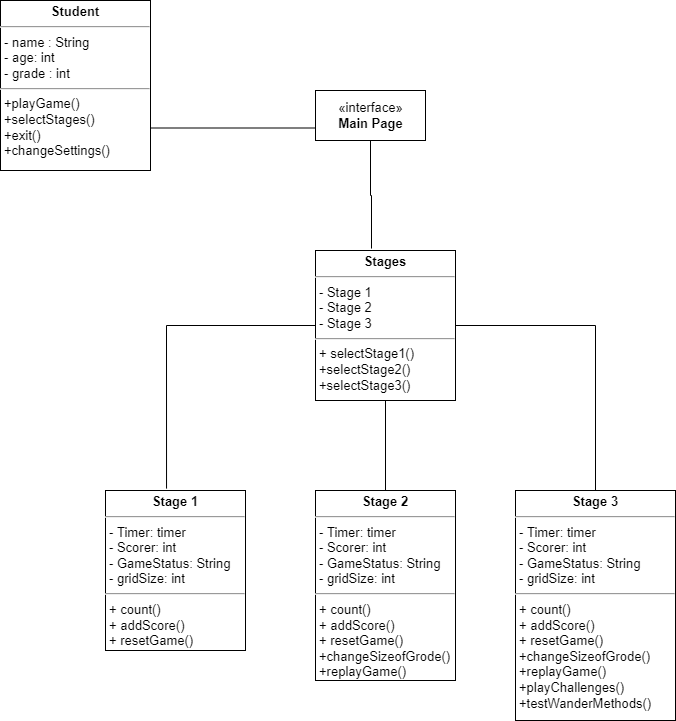
Application Server

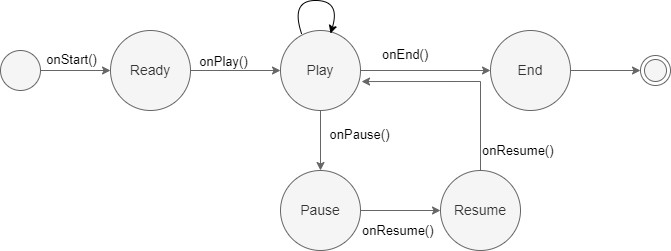
Server Application

SQL

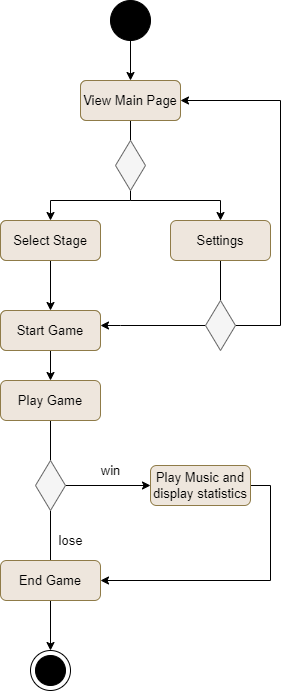
Database Server

4.3 Class Diagram



**4.4 State Diagram**

The above diagram showcases the different states of the game “Wandering in Woods” and it justifies that the most fundamental concept has been reflected.

**4.5 Activity Diagram**

The above diagram showcases the activities of player while interacting with game interface in a system.

**Conclusion**

From the above evaluation, it has been concluded that “Wandering in Woods” is a game developing with the expectation of mathematical simulation would be accomplished soon. The project facilitates the end-user with a platform that gives an entertainment with the knowledge of mathematics.