

An Interactive E-learning Approach to Wind Energy Concepts Using Low Cost Access Devices

Archana Iyer, Qaish Kanchwala, Tanisha Wagh, Dr. Abhijit Joshi

Abstract — In an age where technology is rapidly evolving, new and improved methods of teaching children through e-learning are being developed as well. Furthermore, using low-cost technology will allow the masses to access such educational resources. The application, 'Wind Energy' consists of two applications – the first imparts knowledge to students through a 3D interactive session about conversion of wind energy to electricity while the second teaches students about windmills and their internal working. Blender 3D for modeling, BlenderPlayer and LibGDX for deploying the application have proved to be the best frameworks for the creation of a user-friendly, interactive, free and open source application. By deploying this application on Low Cost Access Devices (LCAD) such as the Aakash tablet, students will be able to learn as well as test their knowledge in the area of Wind Energy. The application complements 2D videos and traditional blackboard teaching by incorporating life-like 3D interaction that makes learning a more interesting process.

Index Terms — Open Source, Android, LibGDX, E-learning, BlenderPlayer, 3D Model, LCAD, Blender

1 INTRODUCTION

The aim was to create an e-learning application to teach concepts about wind energy. This application helps students in learning fundamental concepts of wind energy in an interesting manner. By developing this application, students in the Central Board of Secondary Education (CBSE) schools and engineering colleges will benefit. As this topic is very universal and fundamental in any student's education, 3D models and simulations have been used to portray wind energy concepts that are practically very difficult to show.

The application is divided into two major parts. The first part covers the portion of the CBSE syllabus which teaches students about wind energy. The second part teaches students the anatomy of the windmill, which is useful for engineering students as well. The teaching model of the system uses animations and simulations to teach various concepts and impart theoretical knowledge about wind energy concepts. The assessment model tests the students' understanding about what they have learnt.

The rest of the paper is organized as follows. Section 2 covers the literature review of the various frameworks that have been used to develop the application and the existing systems available to learn the wind energy concepts. Section 3 covers the approach that has been used to develop the application. Section 4 covers the evaluation and results

obtained. The paper ends with the conclusion and future scope of the application.

2 LITERATURE REVIEW

The latest technology entrant to the arena of education is mobility, as a combination of computing and communication technologies. With the help of tablets and mobile devices, the teaching learning process becomes easier, cheaper, faster and more accessible. This section discusses the current teaching methodologies for the topic wind energy concepts and various Game Engines available to develop today's e-learning system to teach wind energy concepts. The section ends with the analysis of existing systems.

2.1 Review of Game Engines

A variety of different Game Engines have been reviewed to develop the application. They include Unity 3D, Shiva 3D, OgreKit, Blender Game Engine, LibGDX, Project Anarchy and Panda 3D. Unity 3D has not been chosen as it is not free and open source. Panda 3D, Project Anarchy and OgreKit have been found unsuitable as they are still in the native stages of development. Shiva 3D was not used as the deployment method for Android was under developed. Let us see the reasons for selection of various Game Engines that are useful in building an effective e-learning environment for learning wind energy concepts. The section also discusses their role in terms of opportunities and challenges in the development of application. In addition to these game engines, various deployment tools have also been reviewed.

2.1.1. Blender

Blender is an Open Source modeling tool. There is a wide toolset available that defines a complete pipeline (from modeling to sequence editing) controlled by a flexible and consistent user interface. Blender's animations system

• Dr. Abhijit Joshi is currently the Vice Principal (Academic) and Head of Department, information technology at Dwarkadas J. Sanghvi College of Engineering, Mumbai. E-mail ID: abhijit.joshi@djsce.ac.in

• Archana Iyer, Qaish Kanchwala and Tanisha Wagh are currently pursuing their Bachelors degree in Information Technology at Dwarkadas j. Sanghvi College of Engineering, Mumbai E-mail IDs: archanaiyer92@gmail.com, qaish.kanchwala@gmail.com, tanishawagh@gmail.com

supports a variety of techniques and tasks, allowing the creation of complex animations. It works on a wide variety of platforms such as Windows XP, Vista, Mac OS X (PPC and Intel), Linux and FreeBSD. Blender works on 32 and 64 bit systems.

Blender files (extension of .blend) require tools to deploy them on various platforms. BlenderPlayer is used to run Blender animations and games on Android and Burster Plugin is required to embed the .blend files in a HTML page. Blender is used in this application to make professional looking models of the anatomy of a windmill with a good degree of interactivity using the Game Engine.

2.1.1.1 BlenderPlayer for Android

BlenderPlayer for Android is a free and open source .apk file used for porting the Blender Game Engine (.blend file) to the Android platform. It supports most of the Python libraries supported by Blender on the desktop. There are many types of interactivities and certain properties of materials such as subsurface modifiers that are still not supported by BlenderPlayer as it is currently under research and development. It can be used to show Blender models, run animations and have simple games on Android devices. A .blend file cannot directly be run on an Android device and requires BlenderPlayer to execute it.

It has been used to run the first part of the application on Android.

2.1.1.2 Burster

Burster 1.5.5 is a lightweight program that can be installed on a wide variety of platforms and is supported by multiple browsers. It is also a free and open source plugin used to integrate the .blend files in HTML pages and create 3D interactivity.

It has been used to run the first part of the application on the Web.

2.1.2 Android SDK

Android SDK has been selected mainly because it is Open Source. Another factor is that the Aakash tablet, which is provided by the Indian Government, runs on Android OS. An Android APK is created using the Android SDK. The APK acts as a user interface for the application. The APK file calls the BlenderPlayer, which runs the Blender models stored on the device.

2.1.3. LibGDX

LibGDX is a Java game development framework that provides a unified API that works across all supported platforms such as Android, Desktop, iOS, Blackberry, HTML. The framework provides an environment for rapid prototyping and fast iterations [12]. Instead of deploying to Android/iOS/Javascript after each code change, it can be used to run and debug the game on the desktop, natively. LibGDX can also use Blender models to create games. It gives direct access to file systems, input devices, audio devices and OpenGL.

It has been used to incorporate the interactivity of swiping the screen to rotate, pinching the screen to zoom in and out and selecting objects for information.

2.2 Existing Systems Study

Apart from the current traditional classroom teaching methods, a number of tutoring systems for wind energy concepts have been developed so far, using different techniques. This section analyses the tutoring system for wind energy mostly those which are freely available. First, this section covers the current prominent systems available to teach wind energy concepts and the section ends with the observations about these systems.

2.2.1. Traditional Classroom Methods

This involves what is usually followed by schools throughout the country for teaching students. The teachers explain the concept by means of textbook and oral explanations. At times they also draw diagrams on the blackboard and test the students' knowledge by means of written worksheets.

2.2.2 Online Methods

In this approach online reading material and simulations are used in the form of 2D or 3D videos to teach Wind Energy concepts.

Some of the prominent 2D and 3D videos are –“3D animation on how wind turbines work” [2], “An overview of the layout of utility-class wind turbine generators” [3], “3D animations on how wind turbines work” [4], “Basics about wind energy” [5], “A game about wind farms” [6] and “Draw and animate 2D windmill” [7].

This is definitely a step up from the traditional classroom teaching method. These videos offer a better visual aid. It is an e-learning platform. More often than not, very technical jargon is used in these videos, which high school students do not understand.

2.2.3 Energy.gov [8]

This website contains an interactive web application, which teaches students concepts about wind energy. It allows the user to see information about different parts of the windmill on Mouse Hover. It also shows the conversion of wind energy to electricity through animation.

Even though this web application is useful, it does not contain an assessment model, does not have touch interactivity and can be run only on the desktop.

2.2.4 Observations on the Existing Systems

1. One Way Interaction

While watching online videos on a streaming site the interaction of the student is only one way, i.e., he receives information from the video and there is no response for the student's query. In case of Energy.gov there is no feedback provided by the system.

2. Poor Interactivity

The concepts about wind energy are taught to the students in a very monotonous manner in the classroom. They are only made to read textbooks and what the teacher writes on the blackboard. This becomes extremely routine and uninteresting to the students as there is no degree of interactivity involved.

In online methods there is limited interactivity, for example only clicking.

3. Large Effort Required for Demonstrating Experiments

In all existing systems more often than not, there is no time to even perform the experiments related to wind energy as each of them involves a lot of preparation and set up.

In traditional classroom scenario there are opportunities for students to perform these basic experiments but these are insufficient to get a complete understanding of the concept.

4. Stationary and Expensive Platform

Most current learning tools work on desktop PCs. Such a platform is not mobile and is also very expensive. Mobile devices are now fast approaching the performance of desktop devices and can run similar type of software. There are cheap mobile platforms such as the AAKASH tablet available in the market which are much more cost effective than the desktops.

Keeping in mind these aspects, we look at the design of our proposed framework trying to address above mentioned problems.

3 THE PROPOSED APPROACH TO TEACH WIND ENERGY CONCEPTS

The 3-D models, animations and interactivity are the core parts of proposed approach for teaching wind energy concepts. Using these principles [1], it was decided to make an interactive e-learning educational application about wind energy for Android devices as well as the desktop. The student will feel like he is performing specific tasks on lifelike 3-D models.

As the application is in the e-learning domain, interactivity and feedback is very important. The student can view the model, zoom in and out, pan the screen, click on parts for information, view the motion of the windmill blades and learn about the flow of energy, which has been represented using arrows. The application will allow students to view the internal parts of a windmill from any desired angle, interact with it and learn about their functions, which will prove to be very educational. After learning concepts, the students can test themselves using the same application. The application aims to satiate students' curiosity and also test their knowledge with appropriate feedback.

3.1 Wind Energy Concepts

Every education system teaches students about the importance of environmental protection. The proposed application teaches wind energy concepts. Concepts for the

first part have been included after going through the Central Board of Secondary Education (CBSE) syllabus from Standards 6 through 10. Information about wind energy for this application has been mapped from Environmental education, Geography and Social Science textbooks. The main focus of Part 1 is to show the conversion of wind energy starting with the motion of the air in the atmosphere till the production of electricity in the houses. It also shows what happens when the wind speed increases.

On the other hand, the second part is intended for mechanical engineers who are already aware of the basic process of energy conversion. For them, the detailed 3-D interactive and animated model of a windmill has been created. An explanation about the various parts of the windmill and what function each part carries out will be shown. For example: Blades lift and rotate when wind is blown over them, causing the rotor to spin.

Looking at the scope of the application now let us view the teaching learning process.

3.2 Teaching Learning Process

As mentioned earlier, the entire application has been split into two parts as shown in Figure 1:

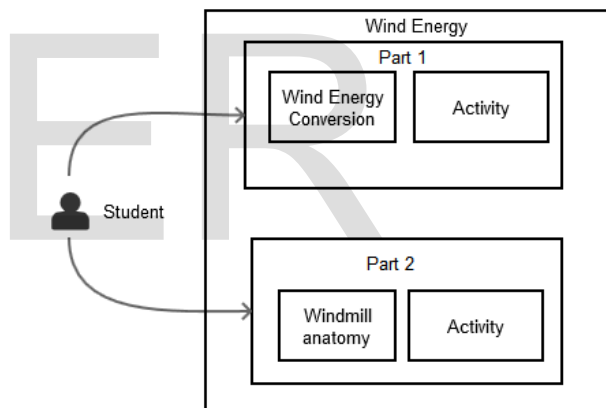


Figure 1: Teaching learning process

The teaching learning model involves the use of interactivity and animations to teach the students and then provide them with feedback and information.

In the first part, the student is provided with an overview of the entire process of creation of electricity from wind energy supplemented with corresponding textual information to explain each step of the process. Once the students understand the process of energy conversion, they are provided with a small game (activity) to test their understanding of this concept. They are shown what happens when the wind speed crosses a certain limit. There is no assessment in this part as the concept is easy to understand.

In the second part, the student is shown the entire detailed 3D model of the windmill. For a better understanding, the student can zoom in and out and pan the model. Each part has been made clickable. When the students click on a part, they are provided with information about it. This enhances

the learning process. After learning and understanding the functioning of the windmill anatomy, the students can assess themselves by answering a quiz on the same 3-D model. In this way, the application assists the students in learning the content in an interesting way.

After understanding the teaching learning process, let us view the proposed architecture.

3.3 Proposed Architecture

The first aspect to take into consideration is the design of underlying structure for web based tutoring system to teach wind energy concepts, which is modelled around the fat client architecture [11]. The layered architecture is represented in Figure 2.

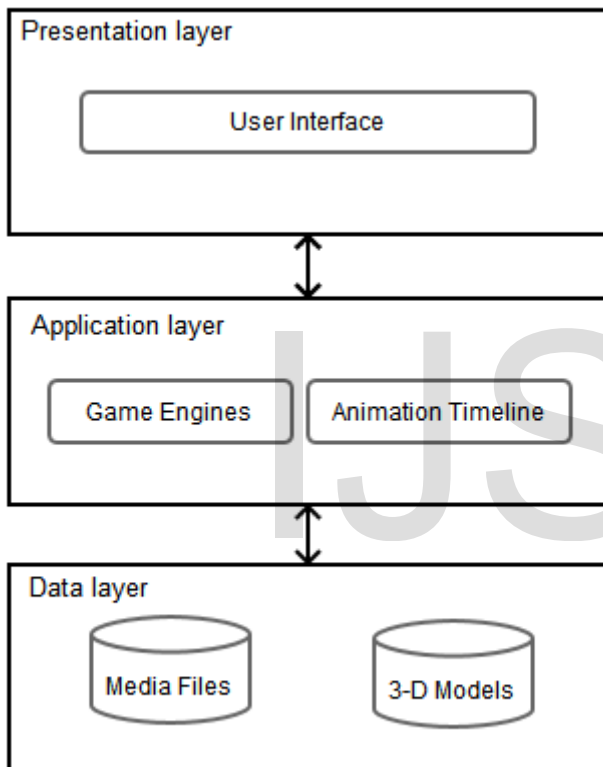


Figure 2: Proposed architecture

An important reason for selecting this framework is that all the functionality occurs on the client side. This section describes in detail the overall architecture of the system and roles played by each layer or model.

3.3.1 Presentation Layer

This layer contains the different UI components that are visible on the screen. It represents how the GUI is displayed and developed for the desktop and for the mobile devices (Android). It serves as the data translator and ensures that data is presented in the correct format. It contains tools, which are used for executing the 3D models present in the data layer. It also acts as the layer for taking input from the user.

For example, when the user clicks on the 3D model the input taken from the presentation layer is passed to the application layer.

3.3.2 Application Layer

It contains components for presentation and business logic. It is used to communicate between the data layer and the presentation layer. This layer consists of the core physics, graphics and game engines. It also involves the use of timeline for animation of models and scenes. It handles the calls to the GPU and synchronization. This layer uses the 3D models, textures and media files from the Data Layer, processes them and gives the output to the presentation layer to display to the user. The input provided by the presentation layer is processed by the game engine to provide a renderable output as per the input provided. If an animation is needed to be played, the animation is taken from the 3D model and processed by the animation timeline and provided as an output to the presentation layer.

As mentioned in the example in section 3.3.1, the click input (3D model) is received from the presentation layer. Following this, the application layer passes it to the game engine, which plays an animation.

3.3.3 Data Layer

It contains all the 3D models and media files necessary for the application. Any textures, images and sound files that are used in the application, are stored here.

The 3D models, images and textures in this layer are loaded initially when the game instance is created. Every time an input is provided the loaded models are changed and rendered as required.

Looking at the design of the system, let us see its implementation.

3.4 Implementation

In this section we will be discussing in detail the implementation of a system as a realization of approach outlined in section 3.2. The application consists of 2 parts - the first one explains the process of conversion of wind energy to electricity and the second provides details on the anatomy of the windmill.

3.4.1 Implementing Part 1

This section explains the implementation of the energy conversion stage. The implementation steps start with creating the model followed by the creation of the various activities. Now let us see these steps in detail.

3.4.1.1 Creating the Models

3D models of the different components such as the house, windmill, transmission towers, power station and representational arrows required in the production of electricity from wind energy were created. Online sources, CBSE textbooks, images and videos were used as reference for the models. Appropriate textures were found and UV mapped onto the different objects and unwrapped.

The .blend file was provided as an input to the Blender Game Engine to create interactions and to the timeline provided by Blender to create animations. The menu has to be changed from Blender Render to Blender Game in the drop down list as shown in Figure 3.

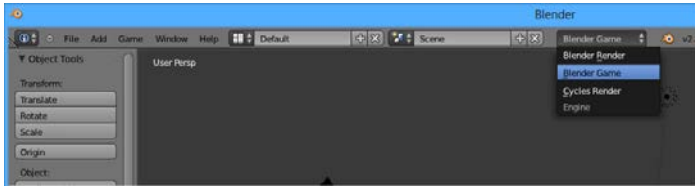


Figure 3: Blender Game Engine

Blender Game Engine (BGE) can be extended for use with Python scripting. This provides a large number of APIs, which have been used in our application, i.e., BlenderRender and GameLogic. BGE can also be extended using Logic Bricks, which consist of sensors, actuators and controllers to incorporate interactivity.

The timeline provided by Blender functions taking frames as its unit. It can be used to set particular frames of animation and change them using the Dope Sheet. This can be seen in Figure 4.

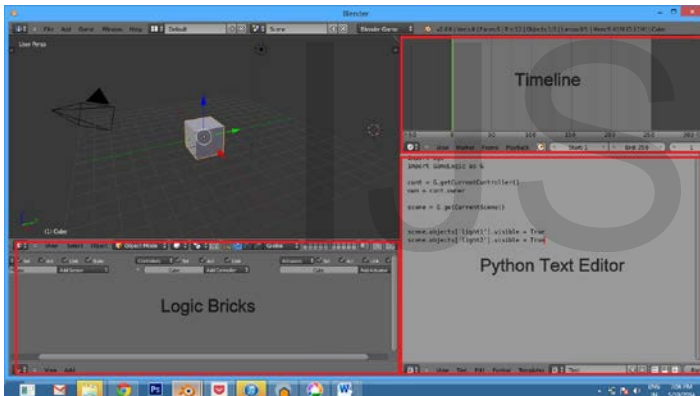


Figure 4: BGE UI

Once the models are ready, one is in a position to create the activities.

3.4.1.2 Creating the Part 1 Activity 1: Energy Conversion

A set of representational arrows show the flow of energy. The timeline was used to synchronize the animation of all these arrows. The arrows were created such that they all appear one after the other on a certain click event. Once the start and end positions were fixed, LocRotScale function was used to lock the positions. All the arrow animations were set within 45 frames for uniformity.

States in Blender were used to show the flow of arrows step by step on the click of the next button. Game properties called "timer" and "check" were created to keep a track of the time and the count of clicks. Each state consisted of a certain set of activities occurring and related animation of arrows being called using message passing. When the next arrow (green) is presented and the user clicks on it, if there

are still states remaining to be viewed, the user is moved to the next state. This is done using the Mouse Click Sensor and Set State Actuator. In a particular state, a timer is set using Python script to make a text box with information visible after a time lag of 5 seconds using "timer" property to view the animation. The Python script is called using a Mouse Over and Mouse Left Click sensor as shown in Figure 5.

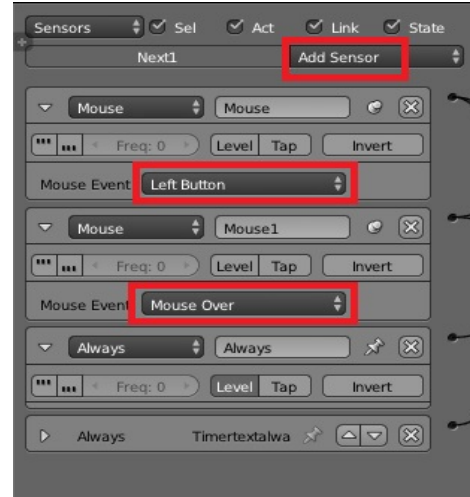


Figure 5: Sensors in the logic bricks of the next button

All the text boxes and components not to be viewed at that time are made invisible and the colour is changed using Python script and the "Always" sensor. The "Visibility" is changed using message passing as shown in Figure 6.

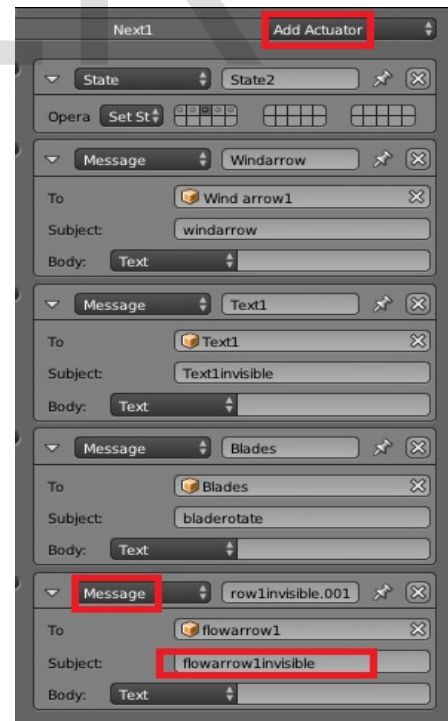


Figure 6: Actuators in the Logic bricks of the next button

A reset button is activated to restart the game using the "Game" actuator to view the entire flow again.

3.4.1.3 Creating the Part 1 Activity 2: Wind Speed

A python script is written which captures the position of the mouse on the screen and calculates it after referencing it to the height and width of the rendering window. The x and the y position of the co-ordinates of the mouse cursor are calculated. These are used to move the slider handle over the slider bar on the screen.

When the bar is at a particular position, based on its co-ordinate values, certain actions are carried out such as the speed of the blades of the windmill increases/decreases, the battery power level increases/decreases, the number of windows lit up increases/decreases and text appears/disappears. This is done in Python scripting by changing the visibility of the different objects using message passing.

3.4.1.4 Deployment of Part 1

After creating the .blend files, the files are run on Android devices directly by using BlenderPlayer for Android. The user interface was created using Java programming on the Eclipse platform and generating an APK.

To run the files on the Desktop, an HTML website was created using the Dream Weaver platform. CSS, JavaScript and HTML coding was carried out to design the user interface to match the one on the Android device. To embed the .blend files in the webpage, burster plugin was used.

3.4.2 Implementing Part 2: Windmill Anatomy

This section explains the implementation of the windmill anatomy stage. The implementation steps start with creating the model followed by the creation of the part 2 activities. Now let us see these steps in detail.

3.4.2.1 Creating the Model

A 3D model of the anatomy of the windmill was created. Textbooks and magazine articles were used as references to create the model. The model has been verified by an expert in order to ensure accuracy of the model.

The .blend file was provided as an input to the LibGDX game engine where the UI was created using Java coding and interactivity was incorporated for gesture recognition as is discussed in detail in the next section.

3.4.2.2 Steps to Create LibGDX Application and Convert Blender models

- 1) First the Blender model must be created with the animations, UV Maps, Materials and Textures that are required by the user. All the meshes present in the model must be given a material and saved as a .blend file.
- 2) The model must be exported in the FBX format. The FBX exporter is built in the newer Blender versions 2.68 and above.
- 3) A FBX converter is provided by Badlogic games and is used to convert the FBX file to a .G3DB file.

- 4) Double click to open the LibGDX jar file and select the "create" option. Enter the name of the packages and the Application the user would like under the "configuration" tab. Select the destination folder of the applications. Also select the platforms to which the user would like the application to be exported to, as shown in Figure 7.

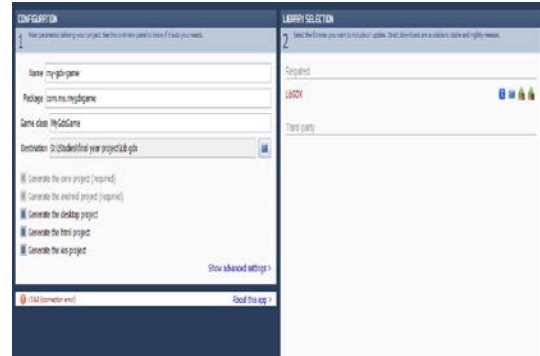


Figure 7: Creating the project and choosing platforms

- 5) Select the LibGDX libraries which are provided on the Badlogic games website. These are used to create the projects.

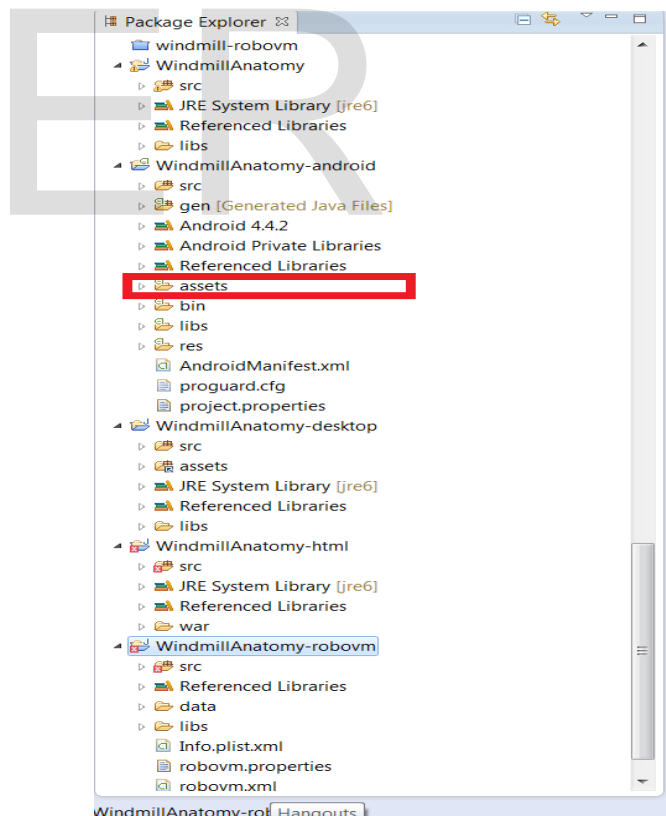


Figure 8: Adding models to assets folder

- 6) The created applications must be imported to the Eclipse IDE. The .G3DB file must be added to the "assets" folder of the android application created (in step 5) as shown in Figure 8.

3.4.2.3 Creation of Part 2 Activities

This section explains the logic behind the learning activity and the testing activity of Part 2 of the application.

3.4.2.3.1 Creating the Learning Activity

The G3DB model imported is loaded on creation of the main game activity. There are different screens created for each activity layer. The screen for the learning activity displays the model that has been loaded in the main game activity. Also the background lighting and the environment are initialized. A Perspective Camera is also created which can be zoomed and panned.

After this the G3DB model object is passed to the touch function. This function creates input listeners for the model. In the touch function, Ray casting is done which gives names of all the objects which the ray passes through. Also the distance from objects to the point of origin of the ray is calculated. Whichever object is at the least distance from the touch point is selected and information regarding that object is displayed.

3.4.2.3.2 Creating the Test Activity

The test activity has its own Screen class different from the learning activity. The G3DB model that is loaded on creation of the main game activity is displayed in the TestActivity screen as well. The environment and the lighting are initialized here. While creating the perspective camera the feature for panning has been disabled, only the pinch zoom feature is allowed. There are a list of questions and answers that have been stored in arrays. When this activity begins, a label displays the question at the bottom which is run in the render method. When the user selects the windmill part as an answer, the touch function explained in section 3.2.3.1 is called. The name of the object selected is compared to the correct answer, which is stored in an array. If they are the same, the selected parts' material changes to yellow and a "Correct" message is displayed on the label. If the wrong answer is chosen, the material of the selected object turns red and an "Incorrect" message is displayed in the label.

After understanding the proposed approach, let us see the walkthrough of the application.

4.0 EVALUATION AND RESULTS

This section starts with a walkthrough of the system as a user. The section ends with an evaluation study conducted to assess how well this approach compares to other modes of learning the concept of wind energy.

4.1 Working with System

After clicking on the application icon on the tablet, a splash screen with the image of a windmill appears as shown in Figure 9. The windmill signifies the concept behind which the application revolves.

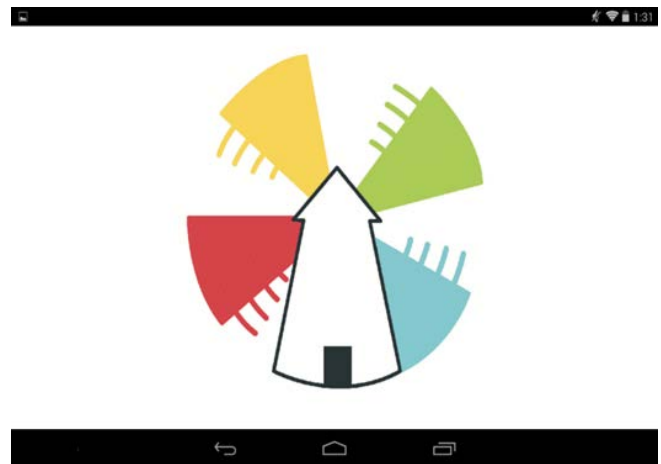


Figure 9: Splash screen

4.1.1 Main Screen

After the splash screen disappears, the student will see a menu with 5 buttons – start, activity, concept, information and exit. Using these buttons he can navigate to any part of the application he wishes to, as shown in Figure 10.



Figure 10: Part 1: Wind Energy application main screen

4.1.2 Part 1 Activity 1: Wind Energy Conversion

On clicking on the "Start" button, the student interacts with the first part. There he is shown the layout of the different main components involved in the production of wind energy, i.e., the windmill, transmission cables, power station, rotor and eventually the house to which electricity is to be provided. All these components are shown using 3-D models. A text box at the bottom of the page explains the flow of the energy in a textual format. This text is synchronised with brightly coloured arrows flowing from one component to the other for better understanding. The student presses the green next arrow to view the next step in the conversion process as shown in Figure 11.

In this way the application allows him to interact via touch

and in turn he learns the steps involved in the energy conversion.

information is provided about the selected part as shown in Figure 13.

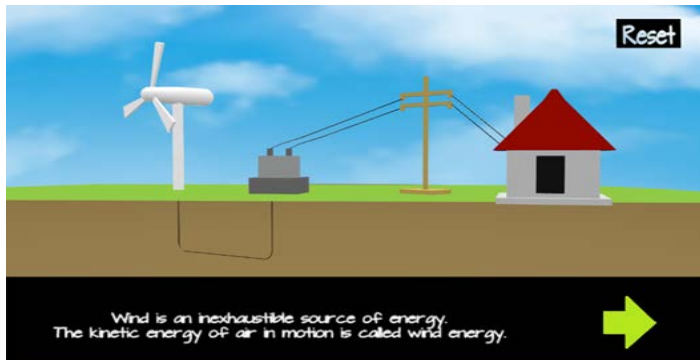


Figure 11: Part 1 Wind Energy Conversion

4.1.3 Part 1 Activity 2: Wind Speed

On clicking the "Activity" button in the main menu, the student can see how different wind speeds affect the amount of electricity generated. As shown in Figure 12, he can view this by dragging the slider from low to high and observing the effects. When the student increases and decreases the slider, the wind energy changes, which causes a corresponding change in the speed of the blades, the battery level, the power produced and eventually the number of houses electricity is provided to.

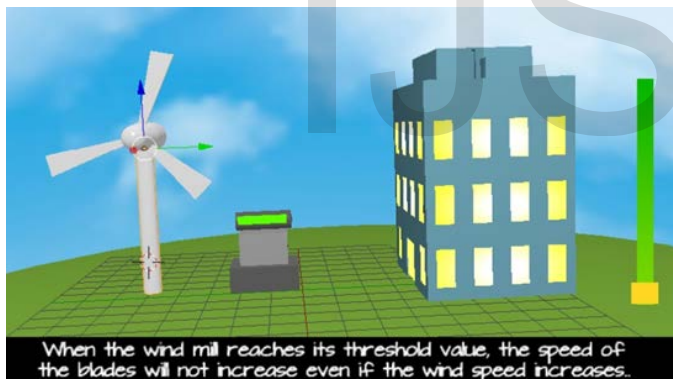


Figure 12: Part 1 Wind Speed Activity

4.1.4 Wind Energy Conversion Concept

On clicking the "Concept" button in the main menu, theoretical concepts about the flow of wind energy will be displayed to the student. He will be able to read and understand the entire process, right from the point where wind energy blows to the point where it is converted to electricity and supplied to houses.

4.1.5 Part 2: Windmill Anatomy

The student is presented with a detailed 3-D model of the anatomy of a windmill on clicking on the application icon on the Android screen. The student can zoom in and out of the model to view parts clearly as shown in Figure 13. He can pan the screen by swiping and view the model from different angles. On clicking on a part of the windmill,

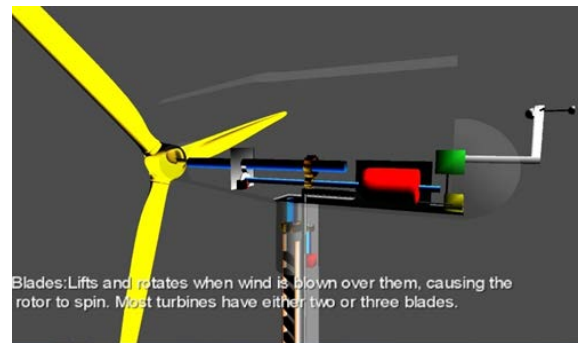


Figure 13: Part 2: Windmill Anatomy

4.1.6 Part 2: Windmill Anatomy Testing Activity

On selecting to perform the activity, the student will be asked questions as shown in Figure 14. The questions are based on what he previously learnt. He has to click on the part he thinks is correct.

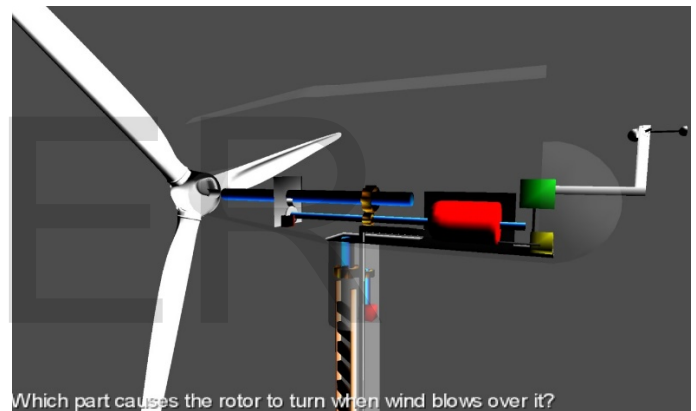


Figure 14: Part 2: Windmill Anatomy Testing Activity

If the part selected is correct, the student will see a result saying, "Correct! Great job!" as shown in Figure 15, or else it will display, "The answer is incorrect" and move to the next question.

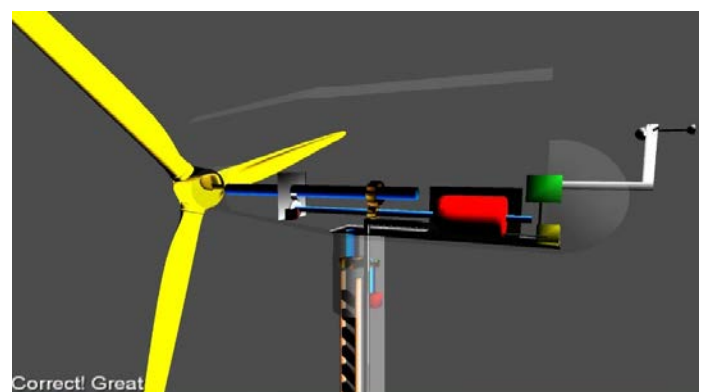


Figure 15: Phase 2: Windmill Anatomy Testing Activity Result

In this way, the dialogue and constructivist feedback helps the student to learn wind energy concepts.

4.2 Experimental Evaluation

The application has been tested using certain test case parameters which are shown in Table 1.

Table 1: Experimental evaluation

Sr. No.	Test Case Parameter	Method Used	Test Results
1.	Application Design	Students and teachers tried out the application.	Attractive layout and design scheme.
2.	Usability	Students and teachers tried out the application.	Easy use and navigation.
3.	Accuracy	Professor of mechanical engineering was consulted about the content and models.	The 3D windmill model and its working is accurate.
4.	Adherence to Syllabus	Extensive research on the syllabi of ICSE, CBSE, SSC science for classes 6,7,8,9 and 10.	Strictly maintained textual concepts and activities as given in the textbooks.
5.	Response time & Speed on various platforms	Tested the application on Desktop/ laptops and Android devices (Aakash tablet, Nexus, S4)	Smooth functioning and consistency in all devices.
6.	Functioning of activity	Students took up the activity before and after studying the concept	Students displayed a sharp increase in their overall score.

An experimental study was conducted in which 50 students were asked to weigh the application on the following parameters - interactivity, improvement, learning and testing, user friendliness and adherence to syllabus. The students were asked to compare this application to existing methods (blackboard teaching, 2D videos and Energy.gov). The chart in Figure 16 shows the comparison. This study proves that, the application provides much better interactivity than the existing system. The improvement of the student after using the created application was also

evident. They had a better understanding and were able to explain wind energy concepts in a better manner after using the application. The students preferred the method of learning and teaching provided by the application. Students found the application easy to use and navigate and did not get confused with the flow. The syllabus covered by the application was found to be the same as that covered in the textbook.

In this way, the study showed that the application created is a better learning method, provides better interactivity, is more user-friendly and also adheres to the textbook syllabus.

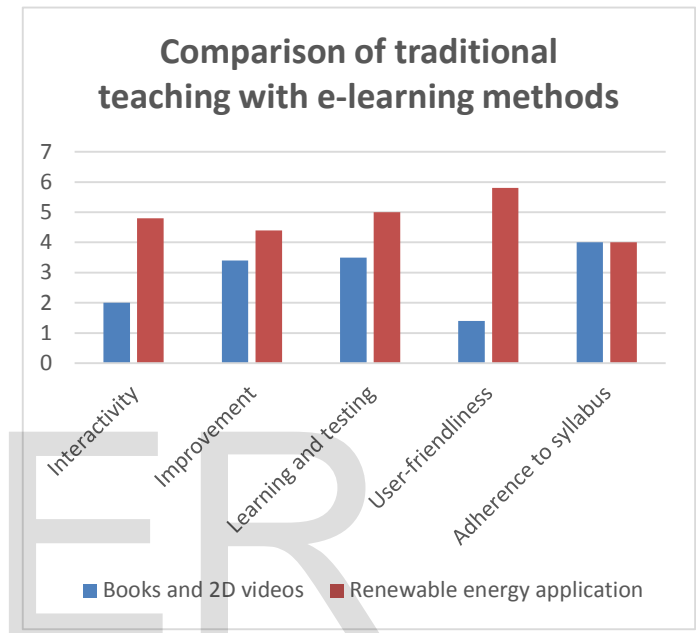


Figure 16: Comparison of methods

5 CONCLUSION AND FUTURE SCOPE

The Wind Energy application provides students with a strong understanding of concepts about wind energy using 3-D models. Since the application is being deployed on Low Cost Access Devices, a student who cannot afford expensive technology or other costly resources will find this to be an economical solution.

The interactivity aspect of the application allows the student to not only observe but also experiment with the various components and in turn learn the functioning of the windmill.

Another important factor is that the application allows students to view everything at different angles.

Although the application covers most of the requirements of an e-learning application, it is possible to expand it for other renewable sources of energy and add more activities to test the students on the concepts. Further research on 3-D game engines on Android can be carried on and LibGDX can be explored in more detail.

6 REFERENCES

[1] S. O. J. G. J. H.-. M. I. o. T. Eric Klopfeir, "The

instructional power of digital games social networking simulations and how teachers can leverage them," Boston.

[2] This 3d animation shows how wind turbines work and explains the main parts, "<http://vimeo.com/28237234>". [last accessed on 16 September, 2013].

[3] An overview of the layout of utility-class wind turbine generators - where are the major components, what do they do, and what differences can be found between models and size ranges. "<http://www.youtube.com/watch?v=LNXTm7aHvWc>". [last accessed on 30 August, 2013].

[4] 3D animation on how wind turbines work, "<http://www.youtube.com/watch?v=eXejxcW-XGo>". [last accessed on 15 September, 2013].

[5] Teaches students the basics about wind energy, "Weatherwizkids, "<http://weatherwizkids.com/weather-wind.htm>". [last accessed on 2013 September 2013].

[6] A game about wind farms, "<http://greenpeace.org.uk/climate/wind-farm-game>". [last accessed on 5 September, 2013].

[7] How to draw and animate a simple 2D windmill, "Flash tutorials, "<http://www.flashvault.net/tutorial.asp?ID=38>". last accessed on 5 September, 2013].

[8] Information on how a wind turbine works, "http://www1.eere.energy.gov/wind/wind_animation.html". [last accessed on 8 September, 2013].

[9] B. K. M. P. F. S. B. W. a. M. W.-I. o. M. I. U. u.-. Naim Asaj, "Research Trends on Media Informatics," Ulm university, 2010.

[10] G. E. a. M. R. Ananya Misrai, "Microphone as a sensor in mobile phone performance".

[11] V. Gruhn and A. Kohler, "Aligning Software Architectures of Mobile Applications on Business Requirements," Web Information Systems Modeling, pp. 1118-1120, 2006.

[12] R. Ahmed and A. Jonas, An Evaluation of the Framework Libgdx when developing a Game prototype for Android Devices, Sweden, 2011.

ACKNOWLEDGMENT

The authors wish to thank Prof. Sridhar Iyer and Mr. Sameer Sahasrabudhe of the Computer Science department of IIT Bombay for their assistance and provision of facilities necessary for the research conducted in this application.