



ATENEO DE DAVAO UNIVERSITY

Computer Studies Cluster

Thesis Project

**INNOKI: AN INTERACTIVE LEARNING TOOL FOR PRESCHOOLS USING
MICROSOFT KINECT**

Proponents: BELGA, BRIAN GERALD
RESPECIA, ADULFO EMMANUEL

Course: Bachelor of Science in
Information Technology

Adviser: Mr. Antonio Bulao III

Panelists: Mr. Oneil Victoriano
Mr. Christopher Aris Alviola

School Year: SY 2014-2015

TABLE OF CONTENTS

1. CHAPTER 1.....	3
1.1 Introduction.....	3
1.2 Background of the Study.....	4
1.3 Technology Application Context.....	5
1.4 Objectives.....	6
1.5 Significance of the Study.....	6
1.6 Scope and Limitations.....	6
2. CHAPTER 2.....	7
2.1 Review of Related Works, Literature, and Technologies.....	7
3. CHAPTER 3.....	8
3.1 Research Design.....	8
4. CHAPTER 4.....	11
4.1 Technology Background.....	11
5. CHAPTER 5.....	12
5.1 Results and Discussion.....	12
6. CHAPTER 6.....	21
6.1 Conclusion.....	21
7. CHAPTER 7.....	21
7.1 Recommendation.....	21
REFERENCES.....	21

InnoKi: An interactive learning tool for preschools using Microsoft Kinect

BRIAN GERALD BELGA AND ADULFO EMMANUEL RESPECIA, Ateneo de Davao University

ABSTRACT

This paper explores the potential of Microsoft Kinect as a tool for substituting pencil and paper worksheet activities into an interactive application. The proponents conducted series of tests among the children of Rotary Club of Toril Foundation Inc. Through a close observation of the young child's verbal, non-verbal and facial expressions, the study showed that interactive media technology could potentially improve a preschooler's learning experience. The results also show that improvements are needed to enhance the overall effectiveness of the application.

Keywords

Kinect, WPF, C#, Kinect SDK, Interactive learning.

ACM Reference Format:

Brian Gerald Belga and Adolfo Emmanuel Respecia. 2014. InnoKi: An interactive learning tool for preschools using Microsoft Kinect. *ACM Trans. Appl* (September 2013), 20 pages.

1. CHAPTER 1

1.1 Introduction

It is indeed supposed that learning is presently transcending towards the use of interactive media as a means of introducing lessons to learners of all ages. Evidently, such media is common around children in preschool education. As a matter of fact, children currently in these ages have developed a natural awareness on how to interact with such technology and that it has never been seen in the previous generations. It only proves that technology eventually is affective to these young learners nowadays. The challenge now is to use the current technology and make it educational for today's children.

Though there are some methods on how kids learn without the use of interactive media technology, a bold step has currently been taken to substitute pencil and paper activities for a virtual learning experience.

With the use of Microsoft's Kinect, it is possible to integrate interactive media with learning. Kinect has the potential to enhance classroom participation among students and teachers as it creates opportunities for interaction and discussion. This enhances the relationship between the student and the teacher.

As a tool, Kinect is a low-budget and affordable device. Many users tend to find Kinect enjoyable and interesting in boosting an individual's motivation. Users can employ the information gathered by Kinect with software programs developed by software developers as well as to create highly interactive multimedia applications.

InnoKi is a low-cost tool that could potentially substitute pen and paper worksheets into an interactive media application. It covers preschool topics such as alphabets, counting numbers and basic arithmetic operations as primary subjects of the application. The purpose of this study is to evaluate the effectiveness of today's interactive media, particularly natural interaction, in early childhood education.

1.2 Background of the Study

Kinect is the first gesture-based device that was released by Microsoft Corporation. The innovative technology is a combination of hardware and software contained within the Kinect sensor accessory that can be added to any existing XBOX 360 or Window-based PC. The Kinect sensor is a flat black box, that sits on a small platform, placed on a table or shelf near the television you're using with your XBOX 360. In this line, Kinect was built to simplify things easily in terms of gestures and interactions with computers with this hardware device involved.

With the help of the Kinect SDK, you can simply develop applications using the Microsoft Kinect. The Kinect for XBOX 360 is limited in features compared to the Kinect for Windows because their difference is that the Kinect for Windows is licensed for commercial app distribution and the Kinect for 360 is for development purposes only.

Furthermore, a new generation of learners has already emerged brought about by the advancement of technology. The dynamic learning process in early childhood education has already included interactive media to supplement the child's creativity while learning. In this manner, gesture-based applications open a new door of opportunities in innovating the engagement of young learners to their lessons through interactive technology.

1.3 Technology Application Context

Kinect is considered as the most affordable and cheapest gesture-device ever released in the market. The development of such interactive learning tools could be a substitute to the traditional pen and paper approach in worksheets that are regularly implemented in schools.

Microsoft Kinect is a single-gesture device, a camera system and a peripheral that uses a regular camera to capture images of a subject and its surroundings, and an IR (infrared) sender that can pick up movement within a 3D space. The Kinect uses structured light and machine learning. Inferring body

position is a two stage-process: first compute a depth map (using structured light), then infer body position (using machine learning). A Kinect device is shown in Figure 1.

Kinect Software Development Kit (SDK) enables software developers to create applications that support gesture and voice recognition, using Kinect sensor technology on computers running Windows 8, Windows 8.1 and Windows Embedded Standard 8

This study mainly focuses on integrating Microsoft Kinect sensor, Kinect SDK and a Windows-based PC application for the purpose of developing an interactive learning tool for preschools. The Kinect SDK is bundled with libraries that enables you to develop apps that are even more powerful with the sensor. The Kinect SDK library is written in different programming languages (it supports C#, C++, Visual Basic.NET) in order to use its certain features particularly the Kinect interaction class, the skeletal tracking, higher depth fidelity, and much more



Figure 1. The Microsoft Kinect Sensor

1.4 Objectives

The general objective of this research is to develop an application that will turn pencil and paper worksheet activities of preschoolers into a gesture-based application with the use of Microsoft Kinect.

The following objectives are identified:

- To be able to make an application that would utilize Kinect Interaction, Kinect Control, Kinect Voice Recognition and other Kinect Toolkit functions from the Kinect SDK.
- To be able to integrate preschool worksheet activities specifically alphabets, counting numbers and basic arithmetic in an interactive application.
- To be able to test the application to preschoolers and assess its effectiveness while gaining results that would help improve natural interactive media technologies for Kinect-driven applications.

1.5 Significance of the Study

From a technological perspective, this study is significant in the field of software development for the purpose of integrating the Microsoft's Kinect sensor. This study extends the functions of the Kinect SDK by developing a WPF application that maximizes the capabilities of the library. Researchers who are interested in developing a Kinect application for future works could make this research as a basis of study.

This research study is beneficial for Kinect users and developers since it will help future researchers to validate its purpose not only in interactive media technology but also in other fields like medicine, engineering, social work and other fields. Future researchers may also help improving this paper since the Kinect technology is new to the eyes of the audience.

In an educational perspective, the proposed study will serve the preschool students a more interacting and enjoyable way of doing activities in their schools. It will also help them understand the lessons that will be tackled in classrooms for the purpose of giving them interaction to their environment especially to their peers. The study also touches how these young learners respond on gesture-based applications and how does it affect their motivation towards school activities.

1.6 Scope and Limitations

This study will not focus on the learning efficiency of the preschoolers but rather on its effectiveness in usability among the respondents. This research only covers preschool topics such as alphabets, counting numbers and basic arithmetic operations. InnoKi is not a game but rather an interactive application based from worksheet activities of selected preschool topics.

The research caters only to children in early childhood education. Thus, the modules developed are within the scope of topics appropriate for the chosen respondents.

2. CHAPTER 2

2.1 Review of Related Works, Literature, and Technologies

The developments of the integration of interactive media in learning through different researchers are notable in this present work.

¹Chen and Li (2009) proposed a Kinect-based game learning development to enhance the perception and learning in kids and bring them to their fullest potential. They have used Kinect to develop several interactive motion control games (IMCG), which can be controlled by body motion or even by gesture.

Such an intuitive user interface makes the game easier to control and suits for kids who are not able to do complicated commands. They can simply move their body and fulfill the game. This gives a great sense of accomplishment so that confidence and motivation are cultivated.

²Sanchez-Nielsen R, Anton-Canalis L, & Hernandez-Tejera M proposed a vision-based framework that was presented to allow users interact with computers through hand gestures. They divided their framework into two major modules: hand posture location and hand posture recognition. Each frame is processed by noise removal, skin pixel labelling and color smoothing. Once the edges of the image are found and had been analyzed in blobs, the blob which represents the user's hand is segmented.

³Lee and Wei (2013) discussed the effectiveness of human-computer interaction and its effectiveness towards children. Child-computer interaction (CCI) can become an adaptive smart application based on learner-centric approach in which children take initiative in exploration with learning stimulus including people, events and materials. In the development and the design of CCU, it focuses on Play, Learning and Communicating (PLC).

⁴Liu M, proposed a study wherein it shows how technology could support children in their learning. It explores the use of interactive multimedia in three-to-five year olds. Although many children encountered such technology for the first time, it was noticeable that they were ready for the technology. It was said that interactive multimedia can provide an effective learning environment to different kinds of learners. The researchers believe that computer-generated graphics can assist young children in developing symbolic representations.

⁵Glaubke, C. R. proposed a study about the effectiveness of interactive media among preschoolers' perception and learning. In definition, interactive media are technology products that are created for a mass audience, that allow user to have some level of input and/or control over the action or content, and that respond to the user's input with either verbal or non-verbal communication.

¹Chen, J. R. Y., & Li, D. Y. Kids Magic Learning: Kinect-based Game Learning Development and Practice.

²Sánchez-Nielsen, Elena, Luis Antón-Canalís, and Mario Hernández-Tejera. "Hand gesture recognition for human-machine interaction." (2004).

³LEE, L. C., & WEI, W. J. (2013). CHILD-COMPUTER INTERACTION DESIGN AND ITS EFFECTIVENESS. *Research & Practice in Technology Enhanced Learning*, 8(1).

⁴Liu, Min. "An Exploratory Study of How Pre-Kindergarten Children Use the Interactive Multimedia Technology: Implications for Multimedia Software Design." (1996)

⁵Glaubke, C. R. (2007). The effects of interactive media on preschoolers' learning: A review of the research and recommendations for the future. Oakland, CA: Children Now. www.childrennow.org/uploads/documents/prek_interactive_learning_2007.pdf.

3. CHAPTER 3

3.1 Research Design

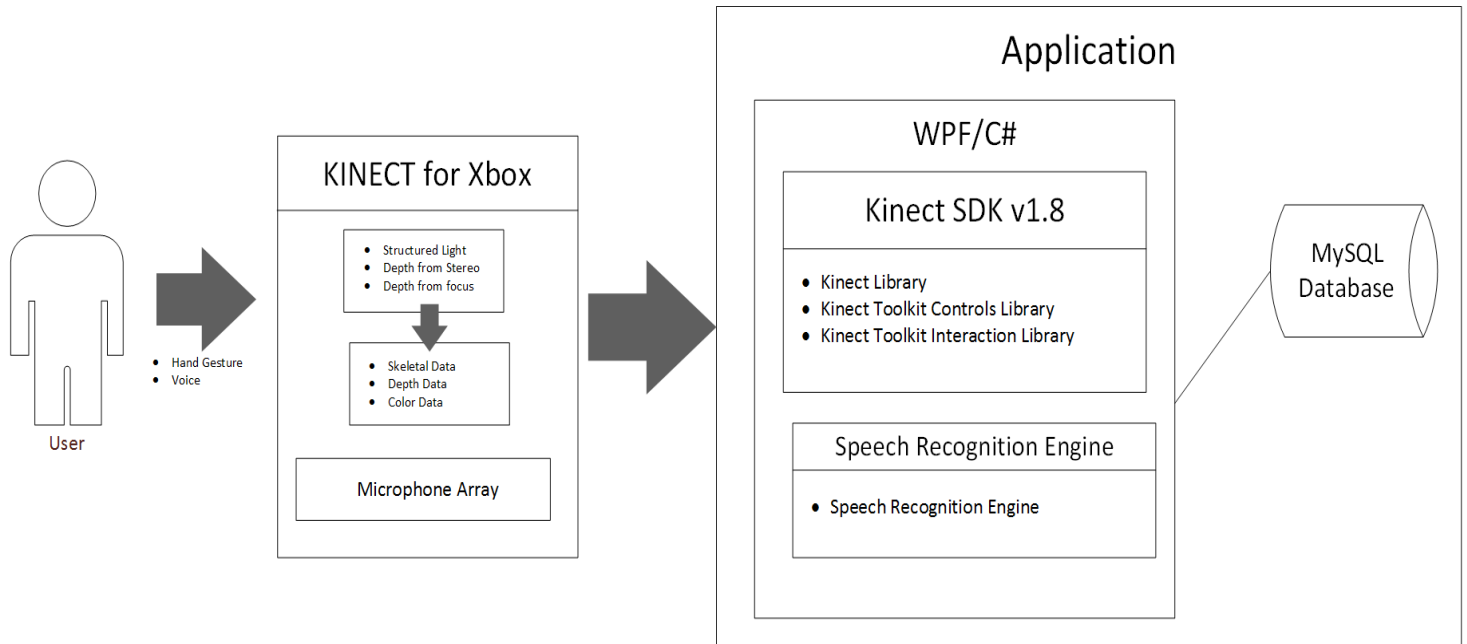


Figure 2. Conceptual Framework

In this section, the proponents will show the technologies/techniques and tools to be used in developing the project.

3.1 Data Capture

The proponents used Microsoft Kinect for XBOX 360 as a 3D sensing device for acquiring skeletal joint data, depth stream and audio stream data.

Microsoft Kinect uses structured light to compute a depth map by analyzing a projected speckled pattern of infrared laser light. It combines structured light with two other computer vision techniques: depth from focus, and depth from stereo. Also, it captures audio through a microphone array located below the RGB cameras and depth sensors. The microphones are built-in with noise cancellation whenever an identified skeleton is tracked. This makes it easier to segment the audio coming from the user.

Moreover, Microsoft Kinect was ported to a computer running on Windows through drivers that come with the Kinect SDK version 1.8.

3.2 Data Processing

3.2.1 Getting skeletal data

Using the built-in Skeleton class in the Kinect SDK, a user skeletal data was initialized and tracked. The class can track and identify users that are within the range of the Kinect sensor and is also responsible with the calculations pertaining the significant joints in the body. Kinect can identify 6 skeletons but can only produce two skeletal data of tracked users. All this was done after initializing a skeletal data stream.

3.2.2 Mapping Skeletal Joints with Depth Data

The mapping of segmented user depth data was handled by the CoordinateMapper class. Once a depth data stream was taken from the device, it was mapped to the selected skeletal joints. The coordinates that were converted in the sequence were used for repositioning elements or controls found in the WPF application.

3.2.3 Using Kinect Toolkit Interaction Stream

The Kinect SDK version 1.8 provides an InteractionStream class that enables the access of data from the user's hands. It gets frame data initialized by Skeletal and Depth frame streams where it uses in calculating hand coordinates including z values, determining hand events and recognizing hand types.

3.2.4 Using Microsoft Speech SDK for Speech Recognition

Microsoft's Speech Recognition SDK contains functions that aids the Kinect Recognizer in processing audio data streams from the Kinect sensor. It includes grammar handling and recognition confidence processing. The proponents used classes in the SDK for key parts of the interactive application that need speech recognition.

3.2.5 Using Kinect Toolkit Controls

The Kinect Toolkit Controls holds a wide range of reusable controls that can be used in an application. It comes with a Kinect region for hand pointers, buttons and scroll viewers. It is included in the developer toolkit released by Microsoft along with other example applications to help first time developers who don't have enough expertise in the Kinect SDK. The proponents utilized these controls to specific parts of the application.

3.3 The Windows Application

The interactive application was developed using C#/WPF in a Visual Studio environment. It is divided into 3 parts according to the 3 subjects namely Alphabets, Counting and Basic Arithmetic. All the media assets except voice recordings used in the application are downloaded from sources online.

3.3.1 Alphabets

Two phases would happen in this section: Read/Listen and Say.

In the Read/Listen phase, the preschooler would hear how a letter and its corresponding object symbol are pronounced correctly. A pre-recorded audio is played during the pronunciation of both the letter and the object (Example: A, Apple)

The Say phase is divided into two parts. The preschooler should first say the appropriate pronunciation of the letter correctly before proceeding to the next part where he/she would do the same thing but this time, in the given object symbol of the letter.

3.3.2 Counting

In this section, a random number with a matching fruit is flashed on top of the screen that pertains to how many times the preschooler would count. The environment will have scattered fruits of three different kinds. The preschooler then is required to grab and drag the identified fruit for n times where n is equivalent to the number of counts. A pre-recorded audio of a current count is played every time a fruit is placed inside the box. The count stops whenever the number of fruits in the box equates to the specified count.

3.3.3 Basic Arithmetic

In this section, there will be a simple equation of either addition or subtraction that will appear at the bottom center of the screen. The answer of the equation is mixed with other different choices enclosed in a balloon. The preschooler should then press the balloon with the right answer. If he/she gets it right, all the incorrect choices would fly up leaving the balloon with the right answer.

3.3.4 Summary

Ideally, a summary of the entire session will appear at the end after the preschooler complete an entire session of the application. The results of every category would be stored in a MySQL database.

3.4 Testing/Evaluation

The application was tested on 12 preschoolers in ages 3-4 at Rotary Club of Toril Foundation Preschool. Each preschooler tested the three different categories of the application.

They are examined through their facial expressions, attitude towards the application and ease of use of the application. The proponents also asked their teachers for feedbacks regarding the application.

A database saves all essential data from the preschooler's use of the application. The data stored is helpful in monitoring child progress in continual use.

4. CHAPTER 4

4.1 Technology Background

This project mainly applies the use of Microsoft Kinect sensor, Kinect SDK and the Visual Studio environment.

First, Microsoft Kinect sensor nowadays is widely used for creating gesture-based and motion-sensing application for the purpose of creating innovative applications in the field of medicine, engineering, social science and other fields of profession. Kinect projects can be categorized into two: First, it can be hand-gestured app and second, it can be a full-body motion app.

Second, Kinect SDK is a software development kit (SDK) that gives businesses the power to create innovative computing experiences that engage their customers and solve business problems through state-of-the-art natural computing technology. The SDK offers a wealth of new tools, samples, and features to help developers' stream line application development and create smarter applications that respond to human voices and gestures in much the same way that people communicate with each other.

Kinect Windows Presentation Foundation Controls (WPF) controls make it easy for developers to build high-quality windows applications. "Push" control to select virtual objects. "Grip" control to pan and scroll selection. Kinect cursor and user viewer. Interaction stream on the other hand enables new functionalities when designing Kinect for Windows Applications, including:

- 1.) Grip Recognition, including the ability to map hand gesture to on-screen sensors. The Kinect for Windows sensor can recognize up to four hand pointers. This allows two people to interact with both hands simultaneously and enables developers to create more complex interactions, including the ability to "zoom."
- 2.) Physical interaction zone, a defined area within which a user can contain their movements to interact with the Kinect for Windows sensor most effectively, similar to the physical area in which you would perform sign language.

5. Chapter 5

5.1 Results and Discussion

In this section, the proponents explained the important parts of implementing the Kinect libraries used in developing the interactive application.

Sample codes were borrowed from Andres Velvart and Jon Somez, contributors of www.pluralsight.com. They are two out of a few credible developers who clearly gave good implementation of the Kinect libraries. The proponents, given their limited knowledge in Kinect application development, incorporated parts of their code from their sample applications which they elaborated on a series of tutorial videos. This was done in order to follow proper programming practice using the Kinect libraries.

Moreover, test results from few preschoolers in Rotary Club of Toril Foundation Inc. were discussed at the later part of the chapter.

5.1.1 Installing Kinect Device to PC

The Kinect Device drivers are automatically installed to the PC after the installation of the Kinect SDK. The drivers are responsible for handling the RGB camera, Depth Camera and Microphone array.

The Kinect SDK also has a viewer for the 3D, depth and color capture data called the Kinect Studio. The Kinect Studio should run along with an application that uses the Kinect Sensor for it returns the current capture information of 3D, depth and the RGB color camera that are used in the application running Kinect.

To test the Kinect Sensor, a developer toolkit is available after installing the SDK. It includes samples and other useful components for developers to test. This research also includes the exploration of these toolkits and incorporate it in the interactive application.

5.1.2 Import Kinect Libraries

```
using Microsoft.Kinect;
using Microsoft.Kinect.Toolkit;
using Microsoft.Kinect.Toolkit.Controls;
using Microsoft.Kinect.Toolkit.Interaction;
```

Figure 3. Include Kinect DLL's to Project

As shown in Figure 3, it is key that Kinect DLL's must be referenced beforehand in the project.

5.1.3 Initialize Kinect Sensor

The proponents had two ways of setting up the Kinect Sensor. The first way was through the KinectSensorChooser class while the other was through the conventional KinectSensor class. In this study, the proponents use both of the classes on different parts of the interactive application.

5.1.3.1 Using KinectSensorChooser Class

After creating an instance of the KinectSensorChooser class in the Loaded Event Handler of the WPF application, the object must initialize a KinectChanged event handler for starting up depth and skeleton streams. Code snippet is shown in Figure 4.

```
1 reference
void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
    sensorChooser = new KinectSensorChooser();
    this.sensorChooser = new KinectSensorChooser();
    this.sensorChooser.KinectChanged += SensorChooserOnKinectChanged;
    this.sensorChooser.Start();
}
```

Figure 4. Set up the Kinect Sensor through KinectSensorChooser Class.

The KinectChanged Event Handler manages cases whenever a Kinect is plugged or unplugged from the computer. It could also set whether the sensor would detect in Near mode or track skeletons in seated mode. But these settings only apply to Kinect for Windows, and since this study used the sensor from Xbox 360, then there was no need to set the detection to either in Near or Seated mode. Code snippet for KinectChanged Event Listener is shown in Figure 5.

```
private void SensorChooserOnKinectChanged(object sender, KinectChangedEventArgs args)
{
    bool error = false;

    if (args.OldSensor != null)
    {
        try
        {
            args.OldSensor.DepthStream.Range = DepthRange.Default;
            args.OldSensor.SkeletonStream.EnableTrackingInNearRange = false;
            args.OldSensor.DepthStream.Disable();
            args.OldSensor.SkeletonStream.Disable();
        }
        catch (InvalidOperationException)
        {
            error = true;
        }
    }
    if (args.NewSensor != null)
    {
        try
        {
            args.NewSensor.DepthStream.Enable(DepthImageFormat.Resolution640x480Fps30);
            args.NewSensor.SkeletonStream.Enable();
            args.NewSensor.AllFramesReady += NewSensor_AllFramesReady;
        }
        catch (InvalidOperationException)
        {
            error = true;
        }
    }

    if (!error)
        kinectRegion.KinectSensor = args.NewSensor;
}
}
```

Figure 5. Initialize Data Streams in KinectChanged Event Listener.

5.1.3.2 Using KinectSensor Class

The KinectSensor class is the main class used to access the Kinect sensor. Again, it is important to check if the Kinect is plugged or unplugged from the computer. Also, it is necessary to set which Kinect device is used in the application since multiple Kinect devices can be used simultaneously. Code for manually setting up Kinect through the KinectSensor Class is shown in Figure 6.

```

if (KinectSensor.KinectSensors.Count > 0)
{
    KinectSensor.KinectSensors.StatusChanged += (o, args) => { };
    sensor = KinectSensor.KinectSensors[0];
}

var parameters = new TransformSmoothParameters
{
    Smoothing = 0.1f,
    Correction = 0.0f,
    Prediction = 0.0f,
    JitterRadius = 1.0f,
    MaxDeviationRadius = 0.1f
};

sensor.SkeletonStream.Enable(parameters);

sensor.DepthStream.Enable();
sensor.ColorStream.Enable();
sensor.AllFramesReady += sensor_AllFramesReady;

sensor.Start();

```

Figure 6. Set up Kinect Sensor through KinectSensor Class.

5.1.4 Manipulating Data Streams

Two data streams were mainly used throughout the application: Skeletal data stream and Depth data stream. The data streams were handled after firing the AllFramesReady Event, as shown in Figure 7, of the Kinect Sensor.

This part of code was used in developing the Counting Numbers section of the application. Manual manipulation of mapped skeletal joints and hand interaction data were necessary to allow two hands to be used as hand pointers. The data also served as triggers to the interactive application, allowing users to do grab and drag events in this particular section of the application.

```

sensor.AllFramesReady += sensor_AllFramesReady;

```

Figure 7. Firing the AllFramesReady Event.

5.1.4.1 Getting user Skeletal Data

The proponents got the captured Skeletal frame data from the AllFramesReady event and stored it to a Skeleton array. After checking which skeleton was tracked from the Skeleton array, an instance of the Skeleton class was used to store the tracked skeleton's data. Code is shown at Figure 8.

```

using (var frame = e.OpenSkeletonFrame())
{
    if (frame == null)
        return;
    var skeletons = new Skeleton[frame.SkeletonArrayLength];
    try
    {
        frame.CopySkeletonDataTo(skeletons);
        var accelerometerReading = sensor.AccelerometerGetCurrentReading();
        interactionStream.ProcessSkeleton(skeletons, accelerometerReading, frame.Timestamp);
    }
    catch (InvalidOperationException)
    {
    }

    skeleton = skeletons.FirstOrDefault(s => s.TrackingState == SkeletonTrackingState.Tracked);
    if (skeleton == null)
        return;
}

```

Figure 8. Capture Skeletal Frame Data to a Skeleton Array.

5.1.4.2 Mapping through CoordinateMapper

Using the `CoordinateMapper` class, points and frames can be mapped from other points and frames of either a Color Image, Depth or Skeletal point or frame. This is a new class that is only available from version 1.7 to the latest version of the Kinect SDK. The proponents used the class to map the left and right hand skeletal joint positions to its depth points. Since skeletal joint positions only return raw coordinates of x, y and z, an understandable coordinate system is needed to reposition controls and elements in the application.

In this study, the proponents managed to reposition Image controls of both the right and left hand depending on the mapped skeletal joint position of the user's left and right hand on every frame. Code is found at Figure 9.

```

var jointHandRight = skeleton.Joints[JointType.HandRight];
var jointHandLeft = skeleton.Joints[JointType.HandLeft];
var positionHandRight = skeleton.Joints[JointType.HandRight].Position;
var positionHandLeft = skeleton.Joints[JointType.HandLeft].Position;
var mapper = new CoordinateMapper(sensor);
var depthPointRight = mapper.MapSkeletonPointToDepthPoint(positionHandRight, DepthImageFormat.Resolution640x480Fps30);
var depthPointLeft = mapper.MapSkeletonPointToDepthPoint(positionHandLeft, DepthImageFormat.Resolution640x480Fps30);

Joint scaledJointRight = jointHandRight.ScaleTo(1366,768, .3f,.3f);
Canvas.SetLeft(RightHand, depthPointRight.X);
Canvas.SetTop(RightHand, depthPointRight.Y);

Canvas.SetLeft(RightHand, scaledJointRight.Position.X);
Canvas.SetTop(RightHand, scaledJointRight.Position.Y);

Joint scaledJointLeft = jointHandLeft.ScaleTo(1366, 768, .3f,.3f);
Canvas.SetLeft(LeftHand, depthPointLeft.X);
Canvas.SetTop(LeftHand, depthPointLeft.Y);

Canvas.SetLeft(LeftHand, scaledJointLeft.Position.X);
Canvas.SetTop(LeftHand, scaledJointLeft.Position.Y);

```

Figure 9. Mapping Skeletal Points to its Depth Points through `CoordinateMapper` Class.

5.1.4.3 Kinect Interaction Stream

The InteractionStream class is a separate class from the Kinect Toolkit Interaction Library of Kinect which is available only from version 1.7 to the latest one of the SDK. The proponents applied its capability to detect hand types and hand event types such as grip and grip release. It could also check if the hand is in pressing motion.

In this study, the InteractionStream class was used to classify a hand event type of either the left or right hand. But first, an Interaction Stream class must be initialized with a sensor and an instance of the IInteractionClient interface, an interface that sets the desired Interaction info.

```
interactionStream = new InteractionStream(sensor, new DummyInteractionClient());
interactionStream.InteractionFrameReady += interactionStream_InteractionFrameReady;
```

Figure 10. Initializing the InteractionStream Class.

Skeletal and depth data streams were copied to a UserInfo class which the InteractionStream class uses to get needed skeletal and depth data. An InteractionFramesReady event handler was fired to handle the data taken from a user's hand. Code is shown in Figure 11.

```
private void interactionStream_InteractionFrameReady(object sender, InteractionFrameReadyEventArgs e)
{
    using (var iaf = e.OpenInteractionFrame()) //dispose as soon as possible
    {
        if (iaf == null)
            return;

        iaf.CopyInteractionDataTo(userInfos);
    }
    foreach (var userInfo in userInfos)
    {
        var userID = userInfo.SkeletonTrackingId;

        var hands = userInfo.HandPointers;

        foreach (var hand in hands)
        {
            var lastHandEvents = hand.HandType == InteractionHandType.Left
                ? _lastLeftHandEvents
                : _lastRightHandEvents;

            if (hand.HandEventType != InteractionHandEventType.None)
                lastHandEvents[userID] = hand.HandEventType;

            var lastHandEvent = lastHandEvents.ContainsKey(userID)
                ? lastHandEvents[userID]
                : InteractionHandEventType.None;

            if (hand.HandType == InteractionHandType.Left)
                left = lastHandEvent;

            if (hand.HandType == InteractionHandType.Right)
                right = lastHandEvent;
        }
    }
}
```

Figure 11. Fire InteractionFramesReady event and copy Skeletal and Depth Data streams to UserInfo Class.

5.1.4.4 Speech Recognition SDK

The proponents used the Speech Recognition SDK to process audio from the microphone of Kinect and compared it to words that were added in the grammar. This part of code was used mainly in the Alphabets section of the application where the preschooler should pronounce a letter and a corresponding object symbol correctly.

First the speech recognizer was set to the Kinect Recognizer with the appropriate language culture (Code shown in Figure 12). Then, the words were added to a grammar builder which was then appended to an instance of the Grammar class. Next, the instance of the Grammar class was passed on the speech recognition engine, as shown in Figure 13.

In the SpeechRecognized Event Handler, the result confidence level was compared to a value of 0.2. If the word pronounced is less than 0.2, then the method will return.

```
private RecognizerInfo GetRecognizer()
{
    foreach (var recognizer in SpeechRecognitionEngine.InstalledRecognizers())
    {
        string value;
        recognizer.AdditionalInfo.TryGetValue("Kinect", out value);
        if ("True".Equals(value, StringComparison.OrdinalIgnoreCase) &&
            "en-US".Equals(recognizer.Culture.Name, StringComparison.OrdinalIgnoreCase))
        {
            return recognizer;
        }
    }
    return null;
}
```

Figure 12. Set up Speech Recognition Engine.

```
recognizer = GetRecognizer();
speechRecognitionEngine = new SpeechRecognitionEngine(recognizer.Id);
speechRecognitionEngine.SpeechRecognized += speechRecognitionEngine_SpeechRecognized;

var choices = new Choices();
choices.Add(PublicVariables.letterObjRel[PublicVariables.letters[PublicVariables.indexes[PublicVariables.count - 1]]]);

var grammarBuilder = new GrammarBuilder { Culture = recognizer.Culture };
grammarBuilder.Append(choices);
var grammer = new Grammar(grammarBuilder);
speechRecognitionEngine.LoadGrammar(grammer);
```

Figure 13. Set up Speech Recognition Grammar.

```
void speechRecognitionEngine_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
    if (e.Result.Confidence < .2)
    {
        SoundPlayer tryAgain = new SoundPlayer("Try-Again.wav");
        tryAgain.Play();
    }
}
```

Figure 14. Set recognition confidence level in SpeechRecognized Event Handler.

5.1.4.5 Kinect Toolkit Controls

To have easy access to Kinect controls, add the schema of Kinect in the XAML code of the window of the page, as shown in Figure 15.

```
xmlns:k="http://schemas.microsoft.com/kinect/2013"
```

Figure 15. Add schema to Designer Page.

The proponents used Kinect Buttons in the interactive application. These buttons can be interacted with a predefined hand cursor if declared inside a KinectRegion control. The KinectRegion Control can only add one child so it is necessary to group buttons in a grid or canvas. Code shown in Figure 16.

```
<k:KinectRegion Name="KinectRegion" Margin="10,0,26,75" Grid.ColumnSpan="2">
    <Grid>
        <Label Content="Welcome to INNOKI" HorizontalAlignment="Left" Margin="130,40,0,0"
        <k:KinectTileButton x:Name="cn_button" HorizontalAlignment="Center" VerticalAlig
            <k:KinectTileButton.Effect>
                <DropShadowEffect/>
            </k:KinectTileButton.Effect>
        </k:KinectTileButton>
        <k:KinectTileButton x:Name="alph_button" HorizontalAlignment="Center" VerticalAl
            <k:KinectTileButton.Effect>
                <DropShadowEffect/>
            </k:KinectTileButton.Effect>
        </k:KinectTileButton>
    </Grid>
</k:KinectRegion>
```

Figure 16. Add KinectRegion in the designer with Kinect Buttons under a Grid or Canvas.

5.1.5 Test and Evaluation

In order to assess the effectiveness of the gesture-based interactive application to the preschoolers, the proponents observed three things: facial expression, attitude and ease of use. Observations are shown in Table 1. These would tell how the preschooler behaves toward the application verbally or non-verbally. The proponents then analyzed how these observations come into play and how it can affect the overall experience of the preschooler while using the application.

The proponents tested the application twice on the preschoolers of Rotary Club of Toril Foundation Inc.

Furthermore, results of all the sessions are stored in a database.

InnoKi: an interactive learning tool for preschools using Microsoft Kinect

Name:	Facial Expressions:	Attitude:	Ease of Use:
--------------	----------------------------	------------------	---------------------

John	Focused/Determined/Curious	Like	Normal
Hershey	Bored/Focused/	Dislike	Difficult
Christian	Focused/Happy	Like	Normal
Ryker	Happy/Focused	Like	Normal
Chloe	Curious/Bored	Dislike	Difficult
Keana	Bored/Focused	Dislike	Difficult
Ferizalyn	Focused/Curious	Like	Normal
Jimarie	Focused/Bored	Dislike	Difficult
Sven	Happy/Bored	Like	Normal
Danimar	Focused/Curious	Like	Normal
Eliniel	Focused/Curious	Like	Normal
Zai	Happy/Focused	Dislike	Difficult

Table 1. Evaluation on preschoolers in their Facial Expressions, Attitude and Ease of Use.

5.1.5.1 Preschoolers and their facial expressions

The proponents listed all the facial expressions that the preschoolers made while using the application. Among the facial expressions observed were focused, happy, bored, curious and determined.

Most of the preschoolers were in focused. Their facial expression didn't change even when they had a hard time using the application. It showed that the preschoolers were patient enough in learning how to use the application.

At some point, they got excited when they saw the hand pointers whenever they were able to move it in the screen. Moreover, it was evident in the look of their faces that they were trying their best to master the application.

5.1.5.2 Preschoolers and their Attitude towards the Program

The proponents later asked the preschoolers if they liked the interactive application or not.

7 out of 12 preschoolers liked the application while the others didn't like it. But on observation, preschoolers who liked the application were the ones who were able to interact with the program. These preschoolers were able to push buttons and give out correct answers. For the others who don't like it, they were having a hard time figuring out how to use the program.

It further proves that the attitude of the preschoolers toward the application depended heavily on their experience of using the application.

5.1.5.3 Preschoolers and their Ease of Use

The proponents had identified three classifications in assessing the ease of use of the application namely Easy, Normal and Difficult. In easy, the preschooler is able to navigate through the application with very minimal supervision. In Normal, the preschooler needs guidance in using the application. Lastly in Difficult, the preschooler, even with guidance and supervision, still finds it a very tedious task to use the application.

Almost all of the preschoolers were having a hard time finding their hands in the screen. They needed assistance and direction. They moved and waved their hands trying to find their hands in the screen.

After a few guidance and direction, some managed to figure out how to use the application, given that all of them encountered the application for the first time. Unfortunately to some, they weren't able to do so leading them to be bored and uninterested.

5.1.5.4 Session database

The proponents were able to set up a MySQL database for the session results of the preschooler. This was for the purpose of giving out a summary of results after the preschooler's use of the application. Also, it would help in monitoring a young learner's progress in using the application. But since none of the preschoolers were able to finish one session in two tries due to lack of mastery, no summary results were displayed.

5.1.5.5 Problems Encountered

One of the major problems during testing was the failure of hand detection among preschoolers. This led to negative moods, facial expressions and even the disliking of the application.

The grab and drag feature of the Counting Numbers section of the application wasn't workable for the preschoolers. None of the preschoolers were able to grab one fruit to the box. This was primarily because the application can't detect the preschooler's hand grip state, a key function for the grab event.

Also, the speech recognition on the Alphabets section was interfered by so much background noise from the classroom. This results to most of the failure in recognizing the preschooler's pronunciation.

6. CHAPTER 6

6.1 Conclusion

The proponents has successfully developed a program that turned worksheet activities of selected subject areas from preschoolers into an interesting interactive application with the use of Microsoft Kinect.

Functions from the Kinect SDK were maximized in order to achieve desired results from gesture inputs of the user. Given the above observations among the preschoolers, the overall experience of the young learner depends greatly on his attitude towards the application. In addition, the preschooler's attitude is affected by his/her ease of use of the application.

The proponents therefore assess that the current version of InnoKi isn't fully effective to preschoolers since some of the young learners find navigating throughout the application difficult, resulting to negative attitudes and non-verbal responses. However, some preschoolers were able to interact with the application and showed promising results that could further enhance the study.

7. CHAPTER 7

7.1 Recommendation

The following recommendations are presented as promising ways to improve this study:

- Improve program to easily detect body joints.
- Find algorithms that would detect hand gesture from smaller subjects efficiently.
- Full-body interaction functionalities to exercise preschoolers' motor and coordination skills.
- Additional subjects for preschools to use.
- Teachers can create their lesson plan from the subjects in the application.
- The application should incorporate social and problem-solving skills that can be useful in real-world applications.

REFERENCES

- [1] Hsu, H. M. J. (2011). The potential of Kinect in education. *International Journal of Information and Education Technology*, 1(5), 365-370.
- [2] Liu, M. (1996). An Exploratory Study of How Pre-Kindergarten Children Use the Interactive Multimedia Technology: Implications for Multimedia Software Design.
- [3] Glaubke, C. R. (2007). The effects of interactive media on preschoolers' learning: A review of the research and recommendations for the future. Oakland, CA: Children Now. Retrieved from http://www.childrennow.org/uploads/documents/prek_interactive_learning_2007.pdf.
- [4] Sandanayake, T. C., Lenagala, G. J. Y., Liyanaarachchi, W. L. A. D. V. A., Atukorala, A. D. K. R., & Weerathna, R. S. (2014). An Effective System to Acoustically Convey Sign Language.
- [5] Tölgyessy, M., & Hubinský, P. (2011, September). The Kinect sensor in robotics education. In *Proceedings of 2nd International Conference on Robotics in Education* (pp. 143-146).
- [6] Boutsika, E. (2014). Kinect in Education: A Proposal for Children with Autism. *Procedia Computer Science*, 27, 123-129.
- [7] Ayala, N. A. R., Mendivil, E. G., Salinas, P., & Rios, H. (2013). Kinesthetic Learning Applied to Mathematics Using Kinect. *Procedia Computer Science*, 25, 131-135.
- [8] Tashev, I. (2013). Kinect Development Kit: A Toolkit for Gesture-and Speech-Based Human-Machine Interaction. *IEEE Signal Process Mag*, 30(5), 129-131.

- [9] Chang, Y. J., Han, W. Y., & Tsai, Y. C. (2013). A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy. *Research in developmental disabilities*, 34(11), 3654-3659.
- [10] Lee, J., Gu, H., Kim, H., Kim, J., Kim, H., & Kim, H. (2013, October). Interactive manipulation of 3D objects using Kinect for visualization tools in education. In *Control, Automation and Systems (ICCAS), 2013 13th International Conference on* (pp. 1220-1222). IEEE.
- [11] Sommoool, W., Battulga, B., Shih, T. K., & Hwang, W. Y. (2013). Using Kinect for Holodeck Classroom: A Framework for Presentation and Assessment. In *Advances in Web-Based Learning-ICWL 2013* (pp. 40-49). Springer Berlin Heidelberg.
- [12] Tsai, C. H., & Yen, J. C. (2013). The Development and Evaluation of a Kinect Sensor Assisted Learning System on the Spatial Visualization Skills. *Procedia-Social and Behavioral Sciences*, 103, 991-998.
- [13] Wei, T. W. (2012, October). Detecting the hand-mouthing behavior of children with intellectual disability using Kinect imaging technology. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility* (pp. 295-296). ACM.
- [14] Cassola, F., Morgado, L., de Carvalho, F., Paredes, H., Fonseca, B., & Martins, P. (2014). Online-Gym: a 3D virtual gymnasium using Kinect interaction. *Procedia Technology*, 13, 130-138.
- [15] Zeid, A., Taqi, A., ElKhatib, O., Al-Yaseen, T., & AlMayyan, T. (2014). KinEd: A Kinect-based E-learning Platform to Enhance Collaborative and Kinesthetic Learning. *International Journal of Computer Applications*, 88(10), 1-6.
- [16] Altanis, G., Boloudakis, M., Retalis, S., & Nikou, N. (2013). Children with Motor Impairments Play a Kinect Learning Game: First Findings from a Pilot Case in an Authentic Classroom Environment. *Interaction Design and Architecture (s) Journal-IxD&A*, (19), 91-104.
- [17] Li, C., Yen, H. Y., & Fang, I. Y. The Different Involvement of Kinect Games in Physical Education Courses Effect on the Baseball Batting Skill Learning Performance of High School Students.
- [18] Chang, Y. J., Wang, C. C., Luo, Y. S., & Tsai, Y. C. (2014, June). Kinect-based Rehabilitation for Young Adults with Cerebral Palsy Participating in Physical Education Programs in Special Education School Settings. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Vol. 2014, No. 1, pp. 792-795).
- [19] Berry, R. (2013). Exploring the potential of interactive technology in early stage education with the Microsoft Kinect.
- [20] Hicks, B., Kissinger, J., & Lee, R. Microsoft Kinect Interface for Children's Education.