POLITECNICO DI MILANO

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

# KINECT ENABLED MOTION BASED TOUCHLESS DRAWING TOOL

Prof. Franca Garzotto

Author:

Binbin Xu, 780149

2013-2014

# Abstract

"Drawing is not primarily a creative means of expression," says Eileen Adams, education officer for the Campaign for Drawing. "It is an intellectual activity - a way of understanding the world. Just as pre-verbal infants babble, so young children make marks and squiggles in an effort to order and refine their thoughts. Ideas that are either unformed or only partly formed at the beginning gradually take shape and develop through the process of drawing. "

According to the above words, children draw means they want to express something. But most importantly, they are thinking. It's a intellectual activity. So in my mind, the main task of drawing is to let children do this by their hearts. Right now, there are many tools for children to draw images. For instance, use pen to draw on paper and use mouse to draw on computer. But actually they are drawing by tools (pen or mouse) not "really" by themselves.

This work tries to develop a program let children "really" draw by their limbs. With the help of Kinect, children only need to move their hands during the whole process of drawing. In this way, children can draw anything what they are thinking about directly by their hands.

# Contents

# Chapter 1: Introduction

According to the research of the modern psychology,young children are often thinking based on thing’s shape, color, sound and action. Children can fully express their feelings and understanding of life around through drawing. Also drawing can help to develop children’s observation memory, imagination, creativity and practical ability. So drawing is widely popular among children and also parents.

Right now the most popular modes of “kid paint” are drawing on paper by pen and on computer by mouse. But these are both need third-party tools like pen and mouse. Actually, most children like to express their ideas directly by their limbs, so if they can draw by their hands is much better.

This project focuses on this idea based on Microsoft Kinect, which can detect user’s skeleton, gestures and movements. With this device, computer can print user’s lambs’ movements. Which means children can draw by their lambs in front of the Kinect.

## 1.1 Context

The general context of this project is the development of interactive between children and computer. Specifically, this work will exploit the paradigm of skeleton detection, gesture detection and motion based interaction. Then we can detect skeleton joints, gestures and track movements of each point.

The basic idea is to have the children's movements and gestures, as detected by motion sensing technology (Kinect), then show the track of hands on the screen with different colors which are selected by children. The system is designed to run a screen with the use of the Kinect, a special device which combines an RGB camera, an infrared projector, and a microphone to perform gesture and voice command recognition.

## 1.2 Purpose

This work develops a innovative tool, called “k-Draw”, that for the first time combines skeleton detection, gestures detection, motion tracking and depth image processing to support children free draw. And it also provides some black and white pictures for children to paint different colors.

The potential of motion based interaction for drawing is grounded on recognizing the relationship between physical activity and cognitive processes, and are supported by a growing body of evidence from psychology and neurobiology.

The evaluation of the prototype have pinpointed the potential of motion-based interactive drawing for children. At the same time, they have identified challenging research directions to expand the original project that will be investigated in this thesis.

## 1.3 Structure

The structure of this thesis is as follows.

Firstly, Chapter 1 is the introduction of this project. Which just simply introduces this thesis.

Then the following is "State of art" in Chapter 2. This chapter contains the technology(Kinect) used in this project and reasons why i choose Kinect. At the end of this chapter i also express my previous experience about this technology.

Chapter 3 is the core part of this document. It includes the description of the whole thesis. In this part, I will introduce what should i do, how I solve the problems and evaluation of this project.

The following Chapter 4 is about conclusion and future works.

In Chapter 5, there are some informations about my professor and tutor. And also there is a user manual to guide users how to use this program exactly.

The last chapter is about references.

# Chapter 2: State of the Art

This chapter focuses on the most relevant technology for children drawing. And reasons why I select this technology. The state of the art is also composed by my previous experiences in this field.

## 2.1 Technology

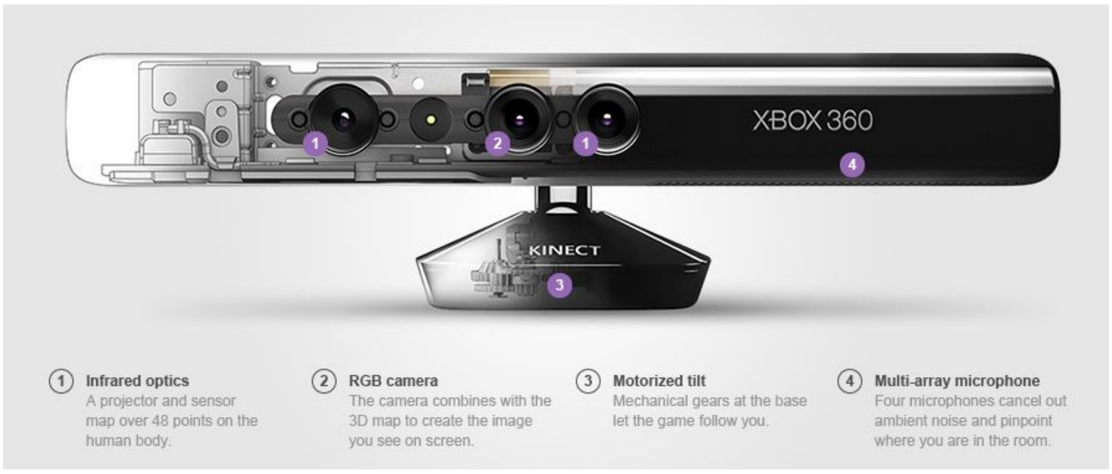
### Microsoft Kinect

Kinect is a motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. Based around a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with the Xbox 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands. The project is aimed at broadening the Xbox 360's audience beyond its typical gamer base. Kinect competes with the Wii Remote Plus and PlayStation Move with PlayStation Eye motion controllers for the Wii and PlayStation 3 home consoles, respectively. A version for Windows was released on February 1, 2012.

Kinect was launched in North America on November 4, 2010, in Europe on November 10, 2010, in Australia, New Zealand and Singapore on November 18, 2010, and in Japan on November 20, 2010. The Kinect claimed the Guinness World Record of being the "fastest selling consumer electronics device" after selling a total of 8 million units in its first 60 days. 24 million units of the Kinect sensor had been shipped as of January 2012.

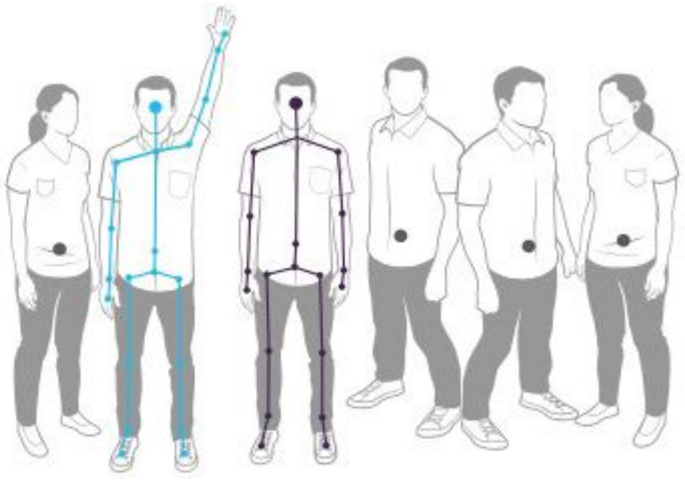
Microsoft released Kinect software development kit for Windows 7 on June 16, 2011. This SDK was meant to allow developers to write Kinecting apps in C++/CLI, C#, or Visual Basic .NET.

The Kinect sensor is a horizontal bar connected to a small base with a motorized pivot and is designed to be positioned lengthwise above or below the video display. The device features an "RGB camera, depth sensor and multi-array microphone running proprietary software", which provide full-body 3D motion capture, facial recognition and voice recognition capabilities. At launch, voice recognition was only made available in Japan, the United Kingdom, Canada and the United States. Mainland Europe received the feature later in spring 2011. Currently voice recognition is supported in Australia, Canada, France, Germany, Ireland, Italy, Japan, Mexico, New Zealand, United Kingdom and United States. The Kinect sensor's microphone array enables the Xbox 360 to conduct acoustic source localization and ambient noise suppression, allowing for things such as headset-free party chat over Xbox Live.


XBOX 360
KINECT
1 Infrared optics
A projector and sensor map over 48 points on the human body.
2 RGB camera
The camera combines with the 3D map to create the image you see on screen.
3 Motorized tilt
Mechanical gears at the base let the game follow you.
4 Multi-array microphone
Four microphones cancel out ambient noise and pinpoint where you are in the room.

The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. The sensing range of the depth sensor is adjustable, and the Kinect software is capable of automatically calibrating the sensor based on game play and the player's physical environment, accommodating for the presence of furniture or other obstacles.

Described by Microsoft personnel as the primary innovation of Kinect, the software technology enables advanced gesture recognition, facial recognition and voice recognition. According to information supplied to retailers, Kinect is capable of simultaneously tracking up to six people, including two active players for motion analysis with a feature extraction of 20 joints per player.



However, PrimeSense has stated that the number of people the device can "see" (but not process as players) is only limited by how many will fit in the field-of-view of the camera.

Reverse engineering has determined that the Kinect's various sensors output video at a frame rate of ~9 Hz to 30 Hz depending on resolution. The default RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280x1024 (at a lower frame rate) and other colour formats such as UYVY. The monochrome depth sensing video stream is in VGA resolution (640×480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect can also stream the view from its IR camera directly as 640x480 video, or 1280x1024 at a lower frame rate. The Kinect sensor has a practical ranging limit of 1.2–3.5 m (3.9–11 ft) distance when used with the Xbox software. The area required to play Kinect is roughly 6 m2, although the sensor can maintain tracking through an extended range of approximately 0.7–6 m (2.3–20 ft). The sensor has an angular field of view of 57° horizontally and 43° vertically, while the motorized pivot is capable of tilting the sensor up to 27° either up or down. The horizontal field of the Kinect sensor at the minimum viewing distance of ~0.8 m (2.6 ft) is therefore ~87 cm (34 in), and the vertical field is ~63 cm (25 in), resulting in a resolution of just over 1.3 mm (0.051 in) per pixel. The microphone array features four microphone capsules and operates with each channel processing 16-bit audio at a sampling rate of 16 kHz.

# Why Kinect?

While the educational value of video games has been debated for years and whether video games could ever be called educational toys, Kinect may have finally made a compelling argument in its favor. Kinect is the latest in gaming technology, although no one could have predicted the potential benefits for those young gamers dealing with paint.

Kinect is an add-on for Microsoft's Xbox, and the game play is motion activated or sensitive. Before Kinect came Wii, but this gaming system is played using a single hand, wireless controller. With Kinect the entire game is powered by the player's motions. For that reason it may be easy to understand how running in place, jumping, ducking and even swinging at things can help a child with their eye-hand coordination as well as other skills.

## Advancing through Avatars

On screen the players appear as a cartoon-like avatar, but the movements they make are powered by the real life movements of the real life players. If the player raises his arm, his avatar does the same at the same time. This process also helps kids grasp body awareness as well as coordination in general. All of these things are helpful for any child to develop and improve. For Autistic children however; this type of learning can be otherwise close to impossible.

Although the intention of Microsoft gaming development was never to promise to be a device to help Autistic kids be able to learn new concepts, no one is complaining. Educators and even child therapists who specialize in working with these challenged children are surprised and amazed at some of the results. In fact, the trend is spreading and even more classrooms and clinics are being outfitted with Kinect gaming systems in an attempt to see a broader base of positive results.

## A therapeutic device

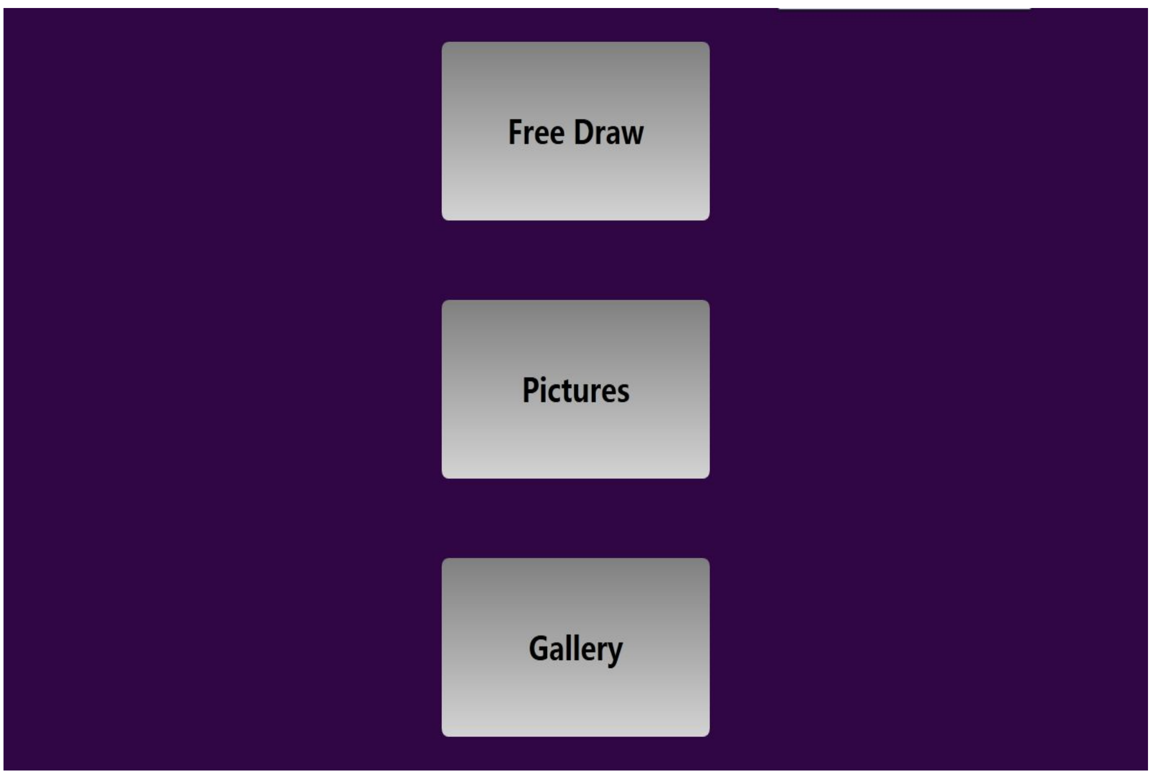
The great news is that the Xbox and Kinect add on still come in cheaper (100$) than the cost of many expensive pieces of equipment that were previously used to have the same results. Could it be that the results are better from this because it stimulates children visually and feeds upon their natural love of video games? For once parents will be thrilled to send their kids off to school and hearing they spent a large amount of that time playing video games.
While it may not quite be labeled a "therapeutic device" researchers have a hard time denying there are some serious benefits and advantages here. The next step could be to expand on the actual educational games that Microsoft offers for the Kinect gaming system. By branching out into subjects such as math and other areas, it will be important to see if this type of training also gets through to young Autistic students. The bottom line is that teachers, therapists and researchers are seeing results from using Kinect that they themselves were not able to get through months of hard work.
Overall we can collect all the advantages of using Kinect for Children in the following terms:

- Social
- Physical
- Creative
- Game's lack of structure and boundaries
- Coordination and body-awareness
- Playful learning
- Highly engaging
- Shared attention
- Safe interaction and predictable environment

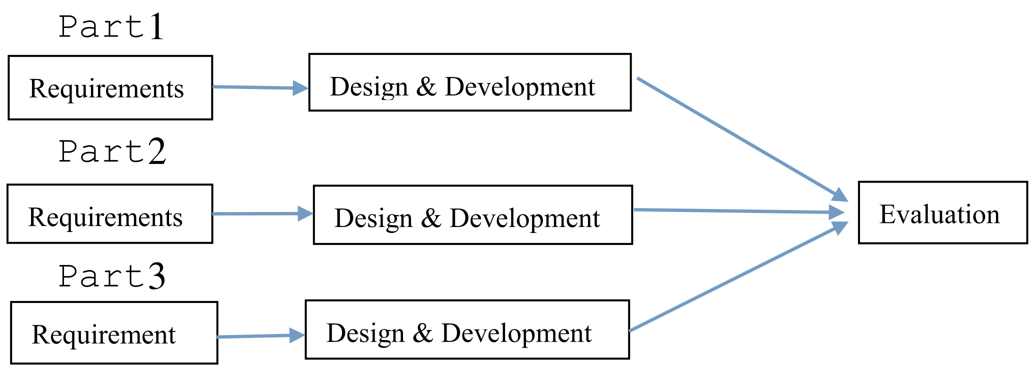
# Chapter 3: Game Description

This chapter includes requirements, technologies i used to design it, implementation and evaluation of this thesis.

The following picture is the main interface of this project,which shows the 3 parts of this software:



The following flow chart shows the main steps:



**Requirements:**
This phase shows the requirements of every part according to different functions. It also includes understanding the system requirements by continuous communication between the customer and the system analyst.

**Design:**
Design phase is used to express how to reach the requirements proposed before.This also contains what technologies are used and how I used them to design the project.

**Evaluation:**
Risk Analysis includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the project, the customer evaluates the software and provides feedback.

## 3.1 Part 1 Free Draw

### 3.1.1 Requirement

The core idea of this part is “Free”, so what is free? I will use some questions and solutions to express requirements of this part.

Q1: How children draw ?

S1: Children can use their right hand to draw anything they want.

Q2: Can children draw colorful images ?

S2: Of course, children can use their lift hand to active the menu which contains many function buttons includes selecting colors.

Q3: How to control the drawing process (start and stop) ?

S3: Thanks to KinectInteraction, which provides some high-level features: Grip and grip release detection, Press detection. In this project, “Grip” means start and “Grip release” means stop.

Q4: How to select a color?

S4: The same as last solution. With the Press detection feature of KinectInteraction, the kinect can detect press button action.

Q5: What can children do after they have drawn a wrong image ? (the same function as rubber)

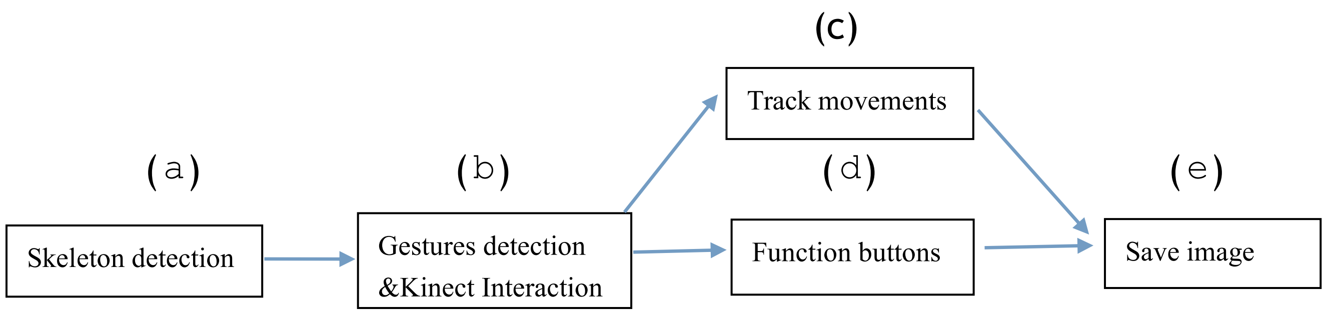
S5: There is a "Clear" button let children clear the drawing board.

Q6: Can children save drawn images ?

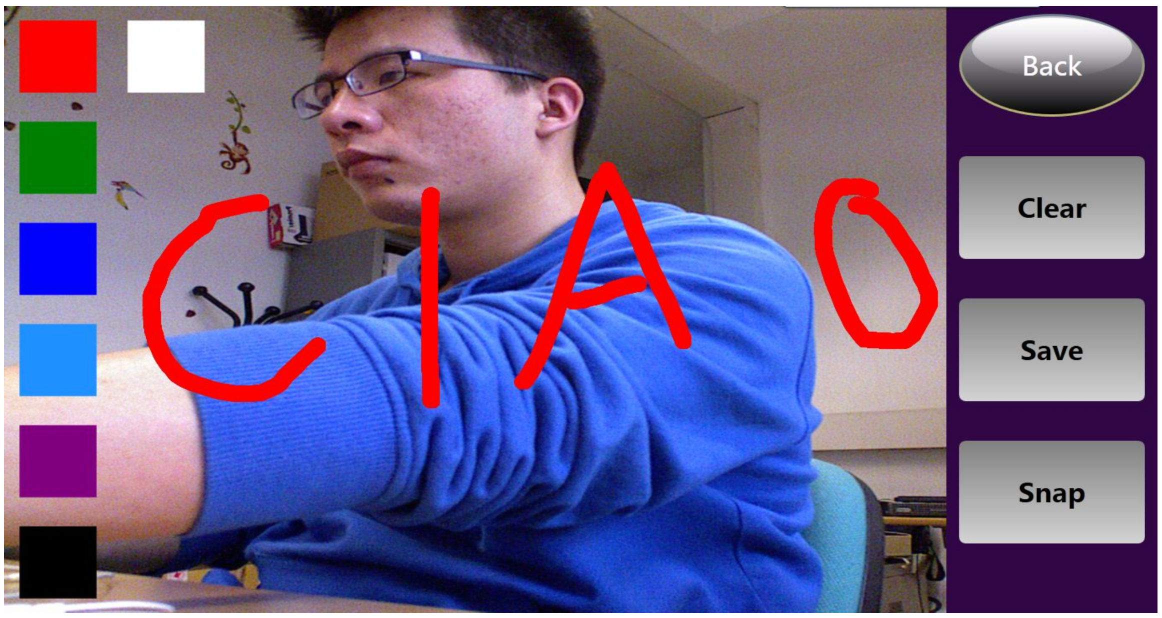
S6: Yes. Children can push the "Save" button to save the image just be drawn and "Snap" button to save the screen shot which contains the drawn image and the background.

## 3.1.2 Design

Here I will express how I solve the questions proposed in last part and it also contains all necessary technologies.



The main interface of "Free Draw" is the follows picture:



## (a). Skeleton Detect

At first the Kinect should makes sure who is drawing ?

The Kinect engine can recognize 6 persons, but it chooses the first two skeletons available for tracking, which is not always desirable largely because the selection process is unpredictable. The 3 kinds of Skeleton-Trackingstate Values of skeletons before the Kinect:

| SkeletonTrackingState | What is Means |
|---|---|
| NotTracked The | Skeleton object does not represent a tracked user. The Position field of the Skeleton and every Joint in the joints collection is a zero point (SkeletonPoint where the X, Y and Z values all equal zero). |
| PositionOnly | The skeleton is detected, but is not actively being tracked. The Position field has a non-zero point, but the position of each Joint in the joints collection is a zero point. |
| Tracked | The skeleton is actively being tracked. The Position field and all Joint objects in the joints collection have non-zero points. |

Then I use the AppChoosesSkeletons property and ChooseSkeletons method to select which skeleton to track. Because in this project, only one person draw at one time.

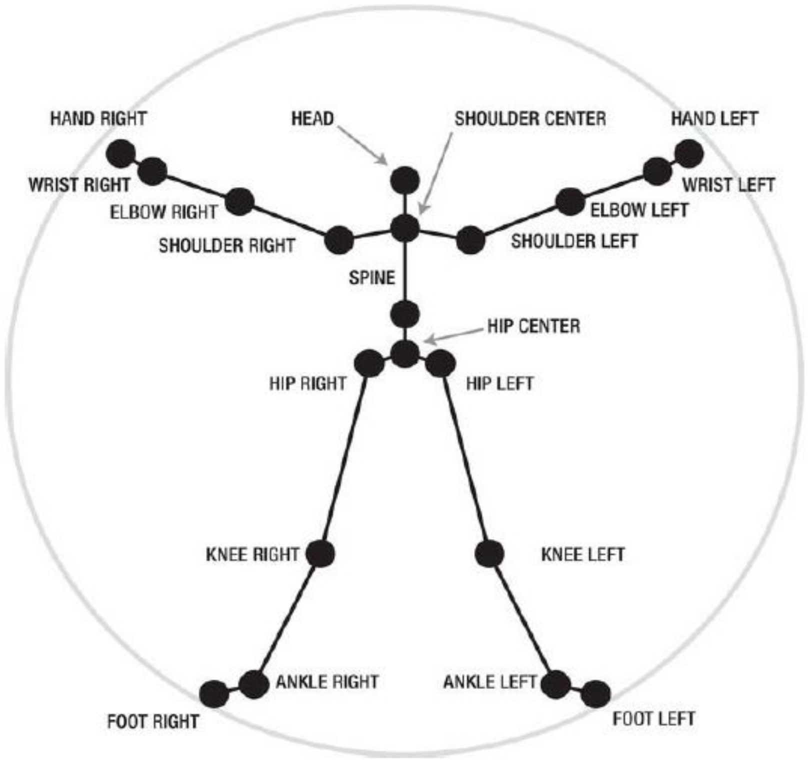
To manually select which skeletons to track, set the AppChoosesSkeletons property to true and call the ChooseSkeletons method passing in the TrackingIDs of the skeletons you want to track. The ChooseSkeletons method accepts one, two, or no TrackingIDs. The skeleton engine stops tracking all skeletons when the Choose-Skeletons method is passed no parameters. Here the engine always tacks the first skeleton, and set a TrackingID for this skeleton. Then during the following time, the Kinect engine only care operations of this ID until this skeleton walk out of view.

## Joints

Each Skeleton object has a property named Joints. This property is of type JointsCollection and contains a set of Joint structures that describe the trackable joints (head, hands, elbow and others) of a skeleton. An application references specific joints by using the indexer on the JointsCollection where the identifier is a value from the JointType enumeration. The JointsCollection is always fully populated and returns a Joint structure for any JointType even when there are no user's in view.

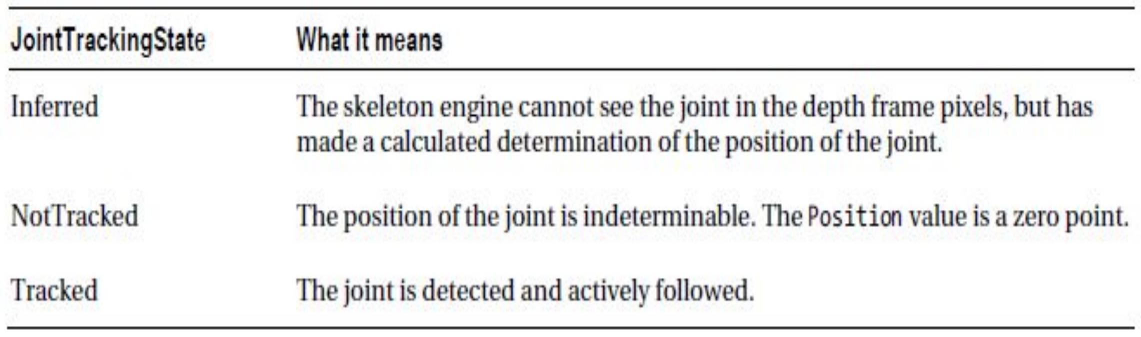
## Joint

The skeleton tracking engine follows and reports on twenty points or joints on each user. The Joint structure represents the tracking data with three properties. The JointType property of the Joint is a value from the JointType enumeration. The following figure illustrates all trackable joints.



Each joint has a Position, which is of type SkeletonPoint that reports the X, Y, and Z of the joint. The X and Y values are relative to the skeleton space, which is not the same as the depth or video space. The KinectSensor has a set of methods (described later in the Space and Transforms section) that convert skeleton points to depth points. Finally, there is the JointTrackingState property, which describes if the joint is being tracked and how.

The following table lists the different tracking states:



| JointTrackingState | What it means |
|---|---|
| Inferred | The skeleton engine cannot see the joint in the depth frame pixels, but has made a calculated determination of the position of the joint. |
| NotTracked | The position of the joint is indeterminable. The Position value is a zero point. |
| Tracked | The joint is detected and actively followed. |

Actually, in this project I only need track 3 points: ”Hand Right” ,”Shoulder Left” and “Hand Lift”. In the following parts I will explain why and how to use them.

(b).Gestures Detect& KinectIteraction

Then i should detect gestures to make sure what the user wants to do.

*"A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed."*

Gesture recognition software has the goal of interpreting human gestures by mathematical algorithms. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

Gesture recognition will enable children to communicate with the machine and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to move a hand at the computer screen so that the cursor will move accordingly.

Our ongoing work is in the computer vision field on capturing gestures or more general human pose and movements by a Kinect sensor connected to a computer. Our gesture detection software should be able to accomplish the following requests:

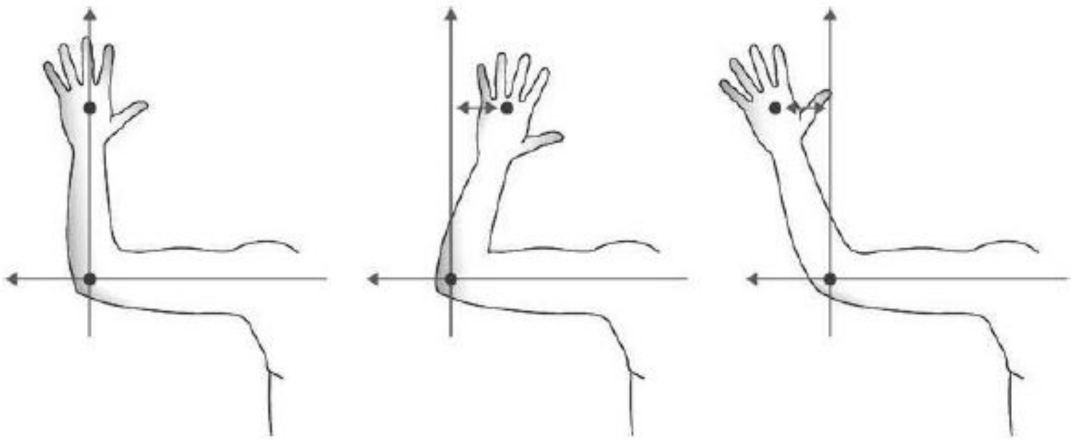
In the following requirements (Picture and Gallery), there are more images than a screen can show. Then children need use "wave" gesture to review and select all images.

## Wave

Anyone who has played a Kinect game on the Xbox has performed the wave gesture. The wave is a simple motion that anyone can do regardless of age or size. It is a friendly and happy gesture. Try to wave and be unhappy. A person waves to say hello and good-bye. In the context of gesture application development, the wave tells the application that the user is ready to begin the experience.

The wave is a basic gesture with simple movements. This makes it easy to detect using an algorithmic approach; however, any detection methodology previously described also works. While the wave is an easy gesture to perform, how do you detect a wave using code? Start by standing in front of a mirror and waving at yourself (as this author has, admittedly, done repeatedly in the course of writing this section). Take note of the motion you make with your hand. Pay attention to the relationship between the hand and the arm during the gesture. Continue watching the hand and the arm, but now observe how the entire body tends to move while making the gesture. Think of the different ways other people wave that is different from your wave. Some people wave by keeping their body and arm still, and oscillate the hand from side to side at the wrist. Others keep the body and arm still, but move the hand forward and backward at the wrist. There are several other forms of a hand wave. Research the wave gesture by observing the way others wave.

The wave gesture used on the Xbox starts with the arm extended and bent at the elbow. The user moves the forearm with the elbow as a pivot point back and forth along a plane that is roughly inline with the shoulders. The arm is parallel to the floor. At the midpoint of the wave gesture, the forearm is perpendicular to both the upper arm and the floor.The following figure illustrates this gesture. The first observation from these images is that the hand and wrist are above the elbow and the shoulder, which is consistent for most wave motions. This is our first testable criteria for a potential wave gesture.



## KinectInteraction

KinectInteraction is a term referring to the set of features that allow Kinect-enabled applications to incorporate gesture-based interactivity. KinectInteraction is NOT a part of the stand-alone Kinect for Windows SDK 1.7, but is available through the associated Kinect for Windows Toolkit 1.7.

KinectInteraction provides the following high-level features:

- Identification of up to 2 users and identification and tracking of their primary interaction hand.
- Detection services for user's hand location and state.
- Grip and grip release detection.
- Press detection.
- Information on the control targeted by the user.

The Toolkit contains both native and managed APIs and services for these features. The Toolkit also contains a set of C#/WPF-interaction-enabled controls exposing these features, which enable easy incorporation of KinectInteraction features into graphical applications.

## Hand Tracking

The skeletal tracking itself is not enhanced by KinectInteraction, the combination of depth information and skeletal tracking information allows KinectInteraction to track a user's hands.

In addition to tracking hands, KinectInteraction can also detect and report hand and arm state, allowing natural gestures such as gripping, releasing, and pressing to be identified. Therefore, a user can now interact with a Kinect-enabled application in a touch-free manner, and at the normal operating range of the sensor (0.4 meters in near mode, up to 3-4 meters in normal mode).

The Kinect sensor must be able to see a hand to track it. The interaction features work best when the user is facing the sensor, and when the user has their hand palm-first to the sensor.

## What Gets Tracked?

The interaction layer is capable of tracking both hands of 1 or 2 users. One of these users is designated as the primary user, typically the first of the two to interact with the system. The primary user keeps control of the interaction until the system determines that the user is no longer engaged with the application. The primary user is assigned a primary hand (the one the user is using to control the experience), although both hands are tracked. Only the primary hand can be used to control the experience. Since both hands are being tracked, the user can switch their primary hand, by disengaging the current primary hand (say, by lowering that hand to their side) and raising the other hand into the PhIZ.

The application can also tell the interaction layer which of the tracked users is primary.
The user is assigned a user ID based on the input from the skeleton tracking stream.

## Grip and Release

In the grip interaction, the user has their hand open, palm facing towards the sensor (ideally), and then makes a fist with their hand. This is recognized as a grip, and binds the hand tracking to whichever control is indicated, until the user releases the grip.

The release interaction is the opening of the closed fist.

Release

(do nothing)

Grip

(drawing)

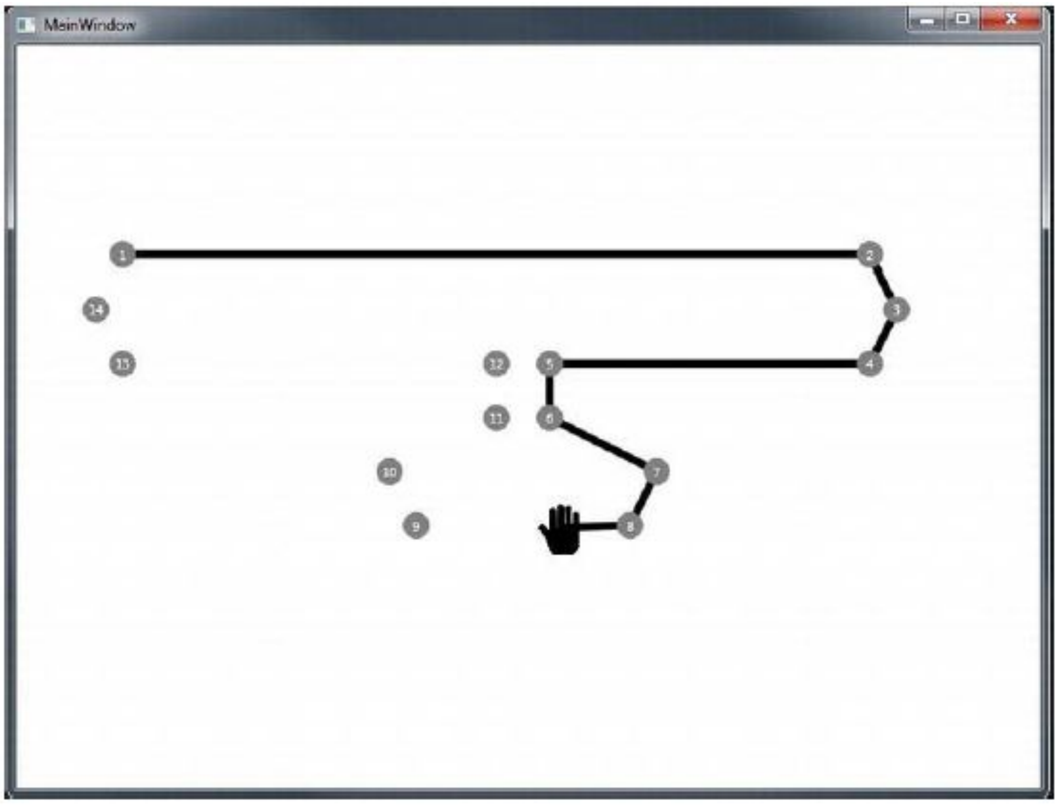
## Press

In the press interaction, the user has their hand open, palm facing towards the sensor (ideally), and arm not fully extended towards the sensor. The user then extends the hand toward the sensor, indicating a press.

Due to this project is about children paint, children should move their hands all the time. And they are young, it is hard for them to make sure wether their hands have already extended or not. To make this project more easier, I use "Grip" instead of "extend". Which means, when child put his right hand on the sensor and then grip his hand, indicates he wants to press the button.

## (c). Track movement

Right now, we can go to the "draw" part. We should show what have been drawn by children.



This part we only need to care the right hand. At first the right hand must in the drawing zone. Then when children "Grip" their right hand means start to draw. When they "Release" right hand, do nothing.

We can understand the drawing technology in a simply way, in this part I just print the point of "Hand Right" joint of every frame. And then link all points together according to the orders of frames.(1 second has 30 frames)

## Bézier curve

This technology is very important for drawing lines. With this i can draw smooth curves, not direct link two points to a straight line.

A Bézier curve is a parametric curve frequently used in computer graphics and related fields. Generalizations of Bézier curves to higher dimensions are called Bézier surfaces, of which theBézier triangle is a special case.

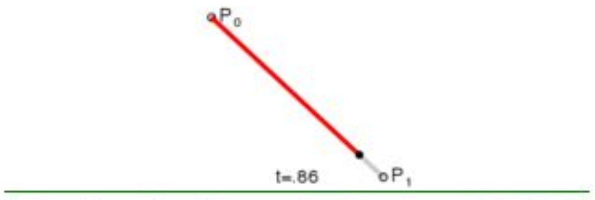
In vector graphics, Bézier curves are used to model smooth curves that can be scaled indefinitely. "Paths", as they are commonly referred to in image manipulation programs,[note 1] are combinations of linked Bézier curves. Paths are not bound by the limits of rasterized images and are intuitive to modify. Bézier curves are also used in animation as a tool to control motion.

Bézier curves are widely used in computer graphics to model smooth curves. As the curve is completely contained in the convex hull of its control points, the points can be graphically displayed and used to manipulate the curve intuitively. Affine transformations such as translation and rotation can be applied on the curve by applying the respective transform on the control points of the curve.

**Linear Bézier curves:**
Given points P0 and P1, a linear Bézier curve is simply a straight line between those two points. The curve is given by

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , \ t \in [0,1]$$

and is equivalent to linear interpolation.



Animation of a linear Bézier curve, $t$ in $[0,1]$

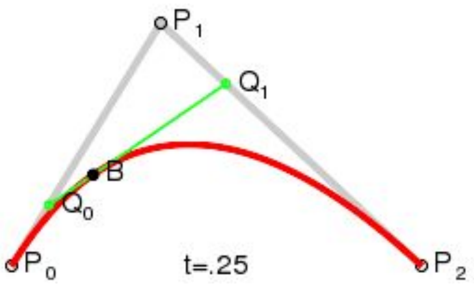
## Quadratic Bézier curves:

A quadratic Bézier curve is the path traced by the function B(t), given points P0, P1, and P2

$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]\ ,\ t \in [0,1]$$

which can be interpreted as the linear interpolant of corresponding points on the linear Bézier curves from P0 to P1 and from P1 to P2 respectively. Rearranging the preceding equation yields:

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2\ ,\ t \in [0,1].$$



## Cubic Bézier curves:

Four points P0, P1, P2 and P3 in the plane or in higher-dimensional space define a cubic Bézier curve. The curve starts at P0 going toward P1 and arrives at P3 coming from the direction of P2. Usually, it will not pass through P1 or P2; these points are only there to provide directional information. The distance

between P0 and P1 determines "how long" the curve moves into direction P2 before turning towards P3.
Writing BPi,Pj,Pk(t) for the quadratic Bézier curve defined by points Pi, Pj, and Pk, the cubic Bézier curve can be defined as a linear combination of two quadratic Bézier curves:

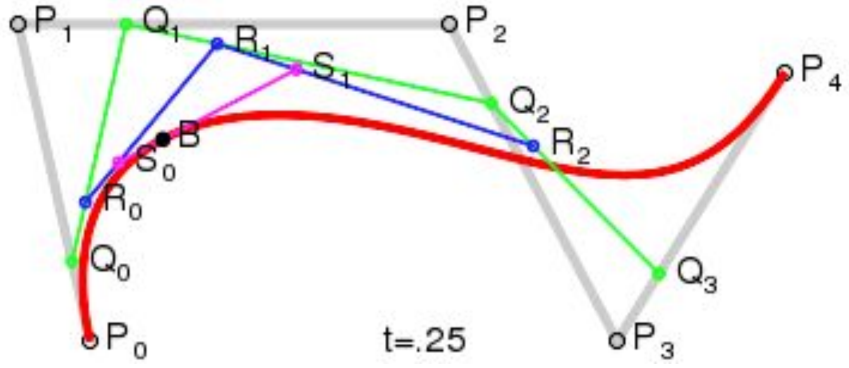
$$\mathbf{B}(t) = (1-t)\mathbf{B}_{\mathbf{P}_0,\mathbf{P}_1,\mathbf{P}_2}(t) + t\mathbf{B}_{\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3}(t) \text{ , } t \in [0,1].$$

The explicit form of the curve is:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \text{ , } t \in [0,1].$$

For some choices of P1 and P2 the curve may intersect itself, or contain a cusp.
Any series of any 4 distinct points can be converted to a cubic Bézier curve that goes through all 4 points in order. Given the starting and ending point of some cubic Bézier curve, and the points along the curve corresponding to t = 1/3 and t = 2/3, the control points for the original Bézier curve can be recovered.[3]
The derivative of the cubic Bézier curve with respect to t is

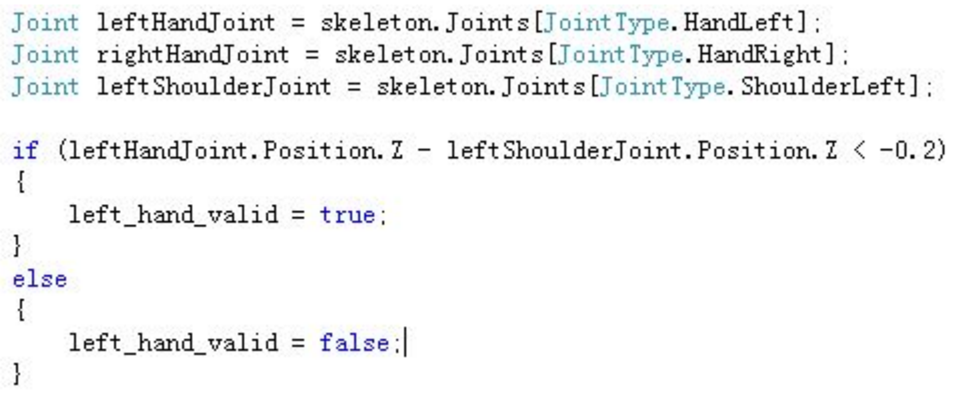
$$\mathbf{B}'(t) = 3(1-t)^2(\mathbf{P}_1 - \mathbf{P}_0) + 6(1-t)t(\mathbf{P}_2 - \mathbf{P}_1) + 3t^2(\mathbf{P}_3 - \mathbf{P}_2) \text{ .}$$



Till now children can use their right hand to "Free Draw" in front of the Kinect. Actually, here i have already finished the main task of "Free Draw".

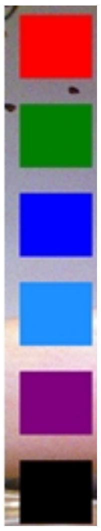
(d). To reach really “Free Draw”, other functions are also necessary. For example, firstly, children wanna draw colorful images. So selecting different colors is necessary. Also children need a rubber to clean the paint zone. Finally, we should save drawn images.

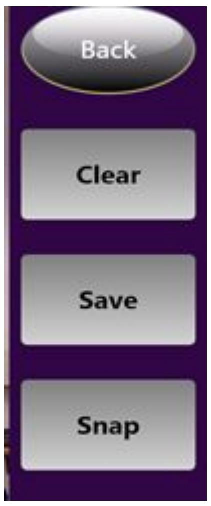
In this part i should detect the position of “Hand Left” and “Shoulder Left”.
We can review the following code:



```
Joint leftHandJoint = skeleton.Joints[JointType.HandLeft];
Joint rightHandJoint = skeleton.Joints[JointType.HandRight];
Joint leftShoulderJoint = skeleton.Joints[JointType.ShoulderLeft];

if (leftHandJoint.Position.Z - leftShoulderJoint.Position.Z < -0.2)
{
    left_hand_valid = true;
}
else
{
    left_hand_valid = false;
}
```

Popular said, as soon as the child rises his left hand higher(20cm) than his left shoulder. The system will active the event to show the following function buttons until the “back” was pressed by the right hand.





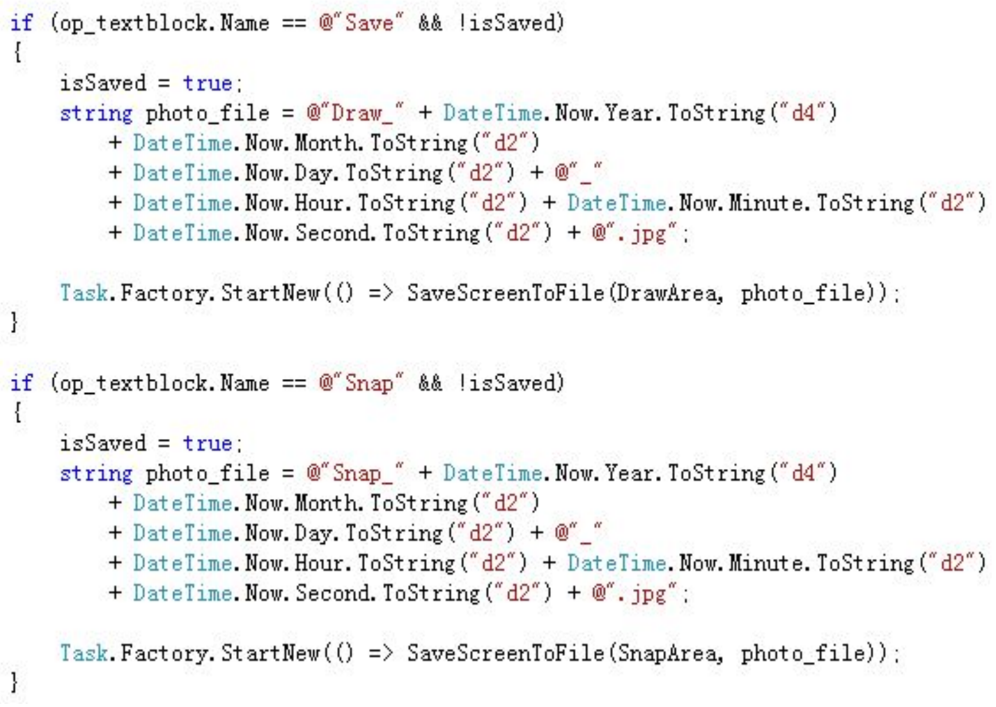
When child moves his right hand on these buttons and “Grip” his right hand. Then the “Press” buttons event will be activated as we discussed in part (b). This action is the same as click by mouse in PC.

## (e). Save images

This is the last step of “Free Draw”:save drawn images. Here I provide “Save” and “Snap” buttons for save function.

After children press “Save” or ”Snap” button, all images are saved in the “bin” folder in this project. And the name of saved image is the time when pressing button, the format is “jpg”.

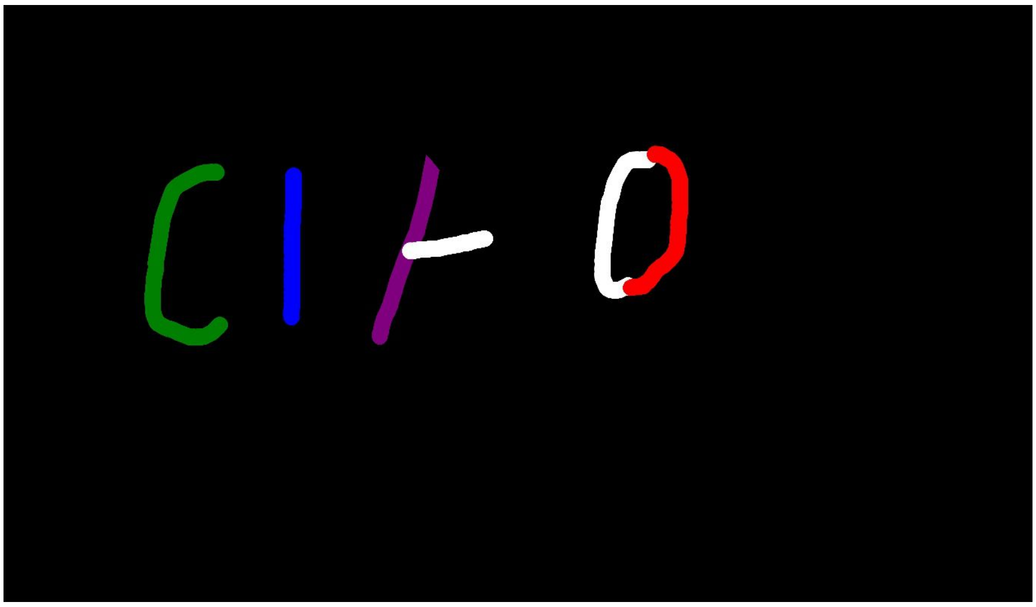
We can understand clearly in the following code:



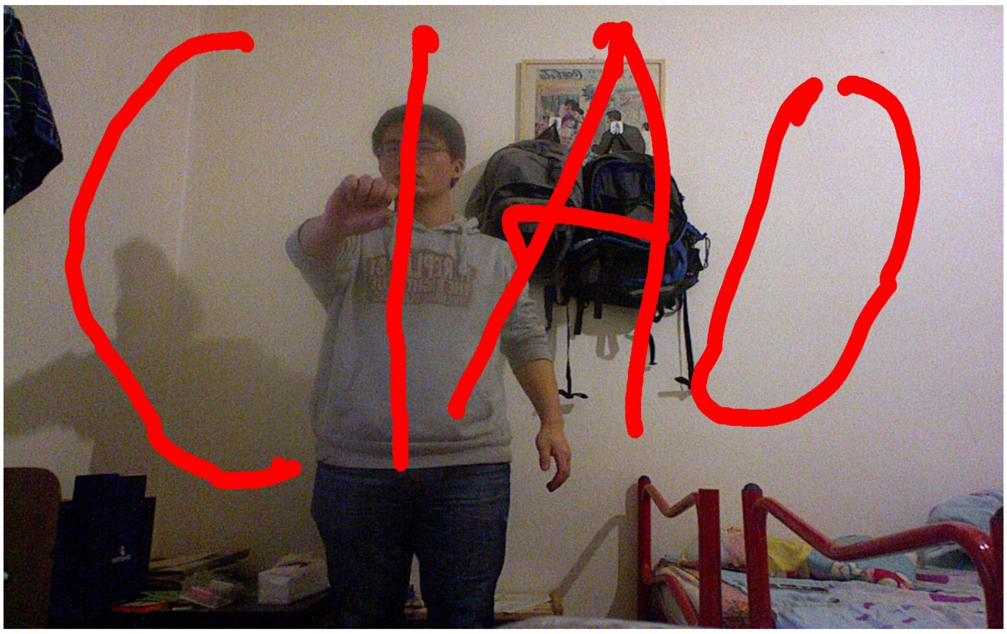
```
if (op_textblock.Name == @"Save" && !isSaved)
{
    isSaved = true;
    string photo_file = @"Draw_" + DateTime.Now.Year.ToString("d4")
        + DateTime.Now.Month.ToString("d2")
        + DateTime.Now.Day.ToString("d2") + @"_"
        + DateTime.Now.Hour.ToString("d2") + DateTime.Now.Minute.ToString("d2")
        + DateTime.Now.Second.ToString("d2") + @".jpg";

    Task.Factory.StartNew(() => SaveScreenToFile(DrawArea, photo_file));
}

if (op_textblock.Name == @"Snap" && !isSaved)
{
    isSaved = true;
    string photo_file = @"Snap_" + DateTime.Now.Year.ToString("d4")
        + DateTime.Now.Month.ToString("d2")
        + DateTime.Now.Day.ToString("d2") + @"_"
        + DateTime.Now.Hour.ToString("d2") + DateTime.Now.Minute.ToString("d2")
        + DateTime.Now.Second.ToString("d2") + @".jpg";

    Task.Factory.StartNew(() => SaveScreenToFile(SnapArea, photo_file));
}
```

To reach different demands, in this project “Save” and ”Snap” buttons have two different functions. “Save” is only to save images which are drawn by children, but the picture saved by “Snap” contains the background of the screen. We can easy distinguish them according the following pictures:

Save:



Snap:



## 3.2 Part 2 Paint in pictures

### 3.2.1 Requirement

Till now people may feel the "Free Draw" is just a scrawl tool, as my professor said, this project should have more education significances for children. So I improved the "Free Draw" part to let children get more interesting and benefits from this part.

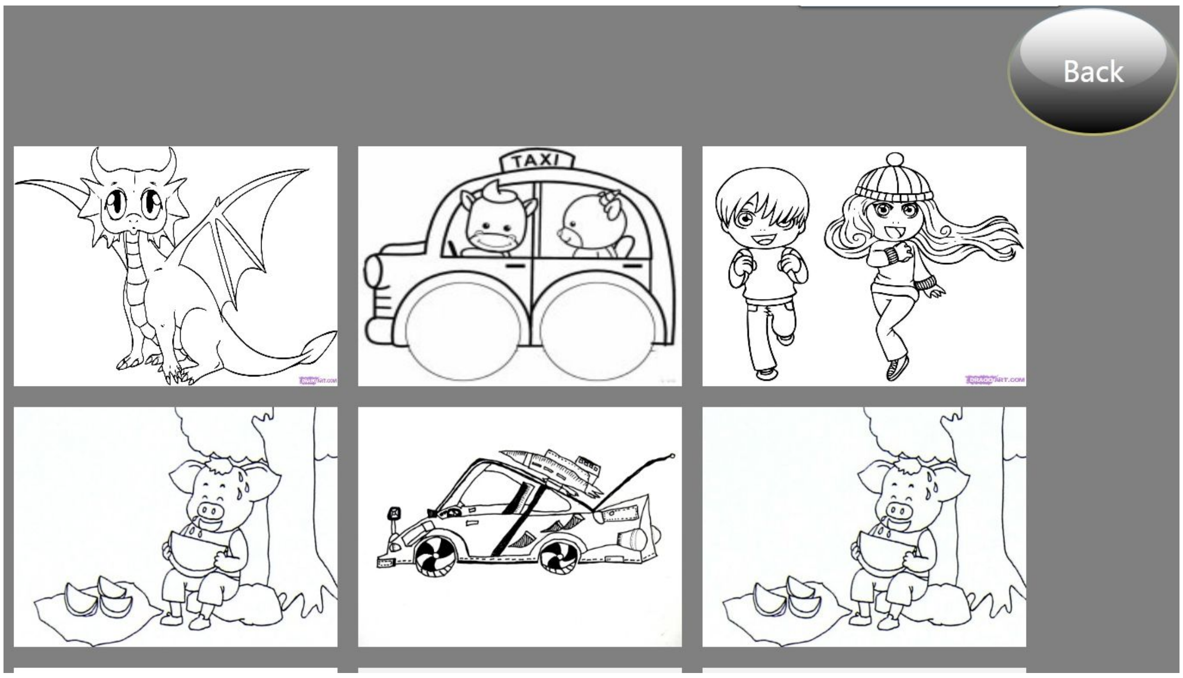
In this part, the project provides some black and white pictures for children to paint different colors.

It is very important to children's education, because teachers or parents can teach children which object should be which colors in normal days. And also if children want they can imagine a new colorful object, like a red or blue race.

Requirements of this part are similar to "Free Draw", but here in the whole screen there should be only one black and white picture for children to draw.

## 3.2.2 Design

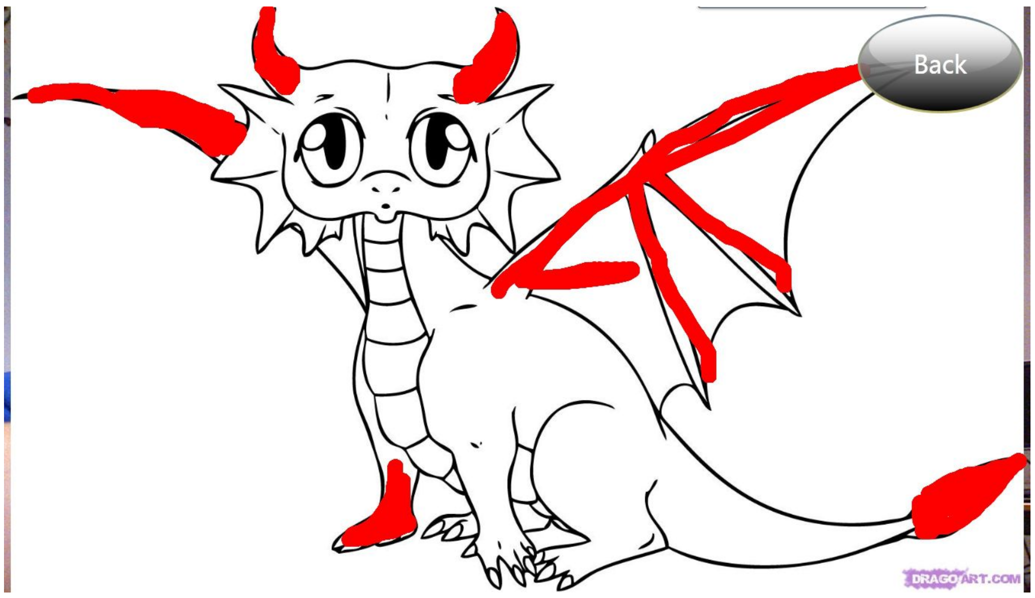
(a). At first, I need provide different black and white pictures for children to select. The main interface is following:



Here I just load the pictures in the “Asset” folder in this project. And parents or teachers can also add and delete pictures in this folder.

As shown in the picture, the screen can only present six pictures. If children want to select others, they have to “Wave” their right hand. This gesture has been discussed in the last requirement.

(b). After children selected a picture, they can paint different colors right now.



The main technologies is the same as last requirement. Use right hand to draw("Grip":start "Release":do nothing) and left hand to active function buttons. Also children can select colors, clear and save pictures with these buttons.

## 3.3 Part 3 Gallery

### 3.3.1 Requirement

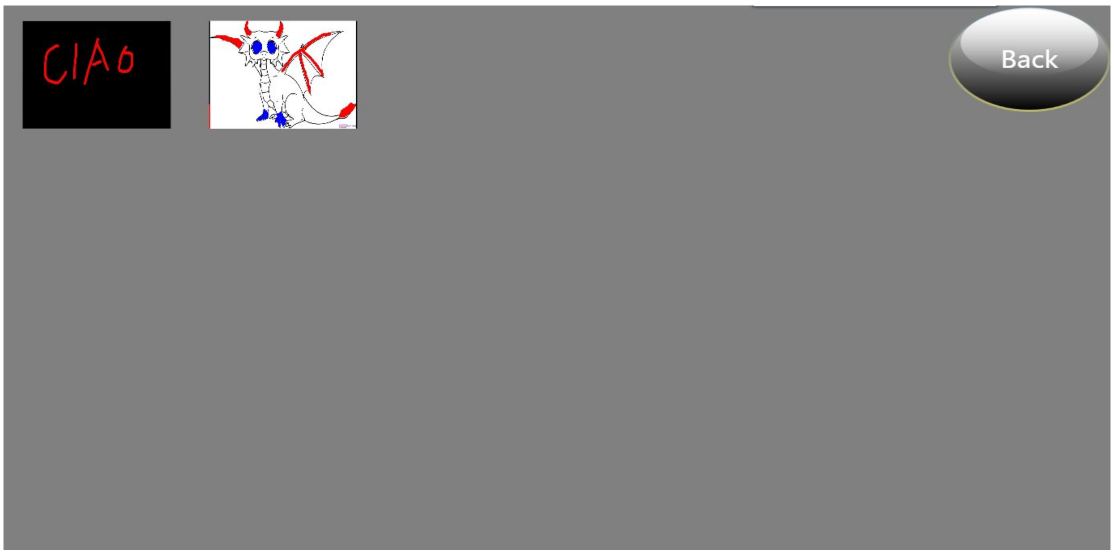
As I proposed before, all drawn images are saved in the "bin" folder in this project. But it is hard for children to open the folder and review images on the computer.

As I tested in China with a 6 years old girl. After about 1 hour teaching, she could use her hands to draw images both in part 1 and part 2. But when I wanted to teach her how to review drawn images in the folder, she said she know nothing about pc. Then how can she use pc to open the folder and find images ?

So the final task of this project is to let children review all saved images by their hands in this system, just like a "Gallery".

## 3.3.2 Design

To this requirement I only need to load all saved images in the “bin” folder as the following picture:



Children can use right hand to do the “Press” action on one of them to review the bigger picture.

I must to say here, children only can see these images but cannot edit them. Here only “show” images.

## 3.4 Evaluation

I finished this project in China during the "Spring Festival", then I tested this system in 2 families in China and gathered some feedbacks. This was an evaluation of the current system.

1. Fang (10 years old, medium functioning level)
2. Wang (6 years old, low functioning level)

Fang was playing with blocks while i was installing the Kinect.

- As soon as I took out the Kinect, he yelled "Is this the Xbox?" and got all excited about it. I had to call him back every time.
- The table was higher here and the angle level had to be set all the way down. Once i got the system running, the first thing i put was the live image of the child on white background.
- Memorable moment: As soon as he saw himself on the screen he jumped in delight, immediately turned back.
- At first, i show the "Free Draw" to Fang. He was very excited about this function. But it took me much time to teach him how to use his hand to draw and change colors.
- Then when we went to "Picture" part, it was very hard to let a 10 years old boy to draw based on the black and white pictures. He always draw anything he wanted.
- Another notable behavior is that in this class was that the kids were moving more towards the board. The Kinect doesn't detect at such close distance so I always had to tell them to move away if they wanted to see "themselves on the screen".

Wang was having lunch when I went to her house.

- I moved the desks, installed the system and explained how to use it.

• Her computer’s processing power was slower compared to other systems and the software took about 10s to start after double clicking.

• She tried out each feature and went over their pros and cons with us.

• She suggested I can add music during the “Free Draw” part.

• Her mother also checked up to what distance it was ok to let Wang stand behind the Kinect’s “dedicated stage”.

• Her mother said she would run it at least twice a day and send me a summary everyday, which she did!

These would be discussed in details in the next meeting with them after 2 weeks before I went back to Milan. Points to discuss on next visit:

• Feedback on the system- How much did they use? What worked? What didn’t?

• Collect log files from the two computers.

• What developments are needed during drawing? Make gestures easier?

# Chapter 4: Conclusion

The project is a continuing work, initial development and usability testing are promising. The ability to easily create or modify learning material around a child's special interest is seen as a major advantage to engaging individuals in social centric activities. In addition, the platform’s ability to record drawing interaction is very promising, as it allows for the collection of data that may reveal learning ability that otherwise would not be available.

I expect that my methodological choices and the establishment of a learning environment that supports active roles for children, body and soul, will gain new scientific knowledge of the possibilities and limitations of technologies in education.

In conclusion, with a little bit of space and a small outlay in money an interactive area can be set up to try to engage our children and help their interactions, creativity and movement.
It can help children to express their thoughts and develop their interests to draw out their thoughts. What is important is that, it is good for intelligence development. It can encourage children to think more.

Anyway, the Kinect is not replacing other methods but adding a whole new level of opportunity to the tools at our disposal. But I am certain that given the resourcefulness and inventiveness of the special needs teaching community worldwide and the support of the business and technical community that these opportunities will multiply and flourish.

# Chapter 5: Appendix

## 5.1 People

Franca Garzotto is an Associate Professor of Computer Engineering at the Department of Electronics and Information, Politecnico di Milano. She has a Degree in Mathematics from the University of Padova (Italy) and a Ph.D. in Computer Engineering from the Politecnico di Milano. Since she joined the Politecnico di Milano, her theoretical research has focused on topics related to (hyper) document modeling, hypermedia design methods, usability engineering, multichannel web application models, adaptive hypermedia, human-computer interaction, while her applied research has mainly addressed the domain of e-learning and e-culture. In the last years part of her research in focusing on story-telling learning technologies and autism-related activities in collaboration with Matteo Valoriani, a PHD student of Politecnico di Milano.

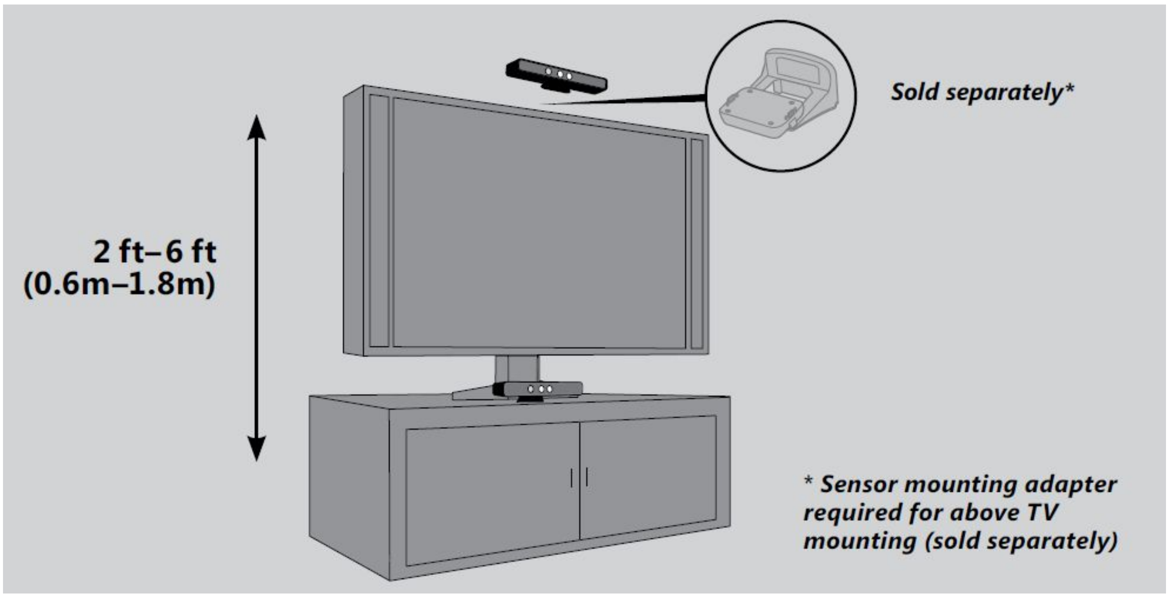
## 5.2 User manual

Following are common instructions position and plug the Kinect sensor, set up the play space taking care of lights, install the software and use the system.
Further development foresees that all of these instructions will be provided inside the S'COOL website in a more organized way.

### 5.2.1 Position the Kinect sensor

Kinect needs to see your entire body

• Place the sensor near the edge on a flat, stable surface.
• Position the sensor between 2 feet (0.6m) and 6 feet (1.8m) from the floor. Ideally, the sensor should be within 6 inches (15 cm) above or below your TV.
• Avoid positioning the sensor in direct sunlight or within 1 foot (.3m) of audio speakers.
• Do not manually tilt the sensor; it adjusts automatically.
• Be careful not to drop the sensor.

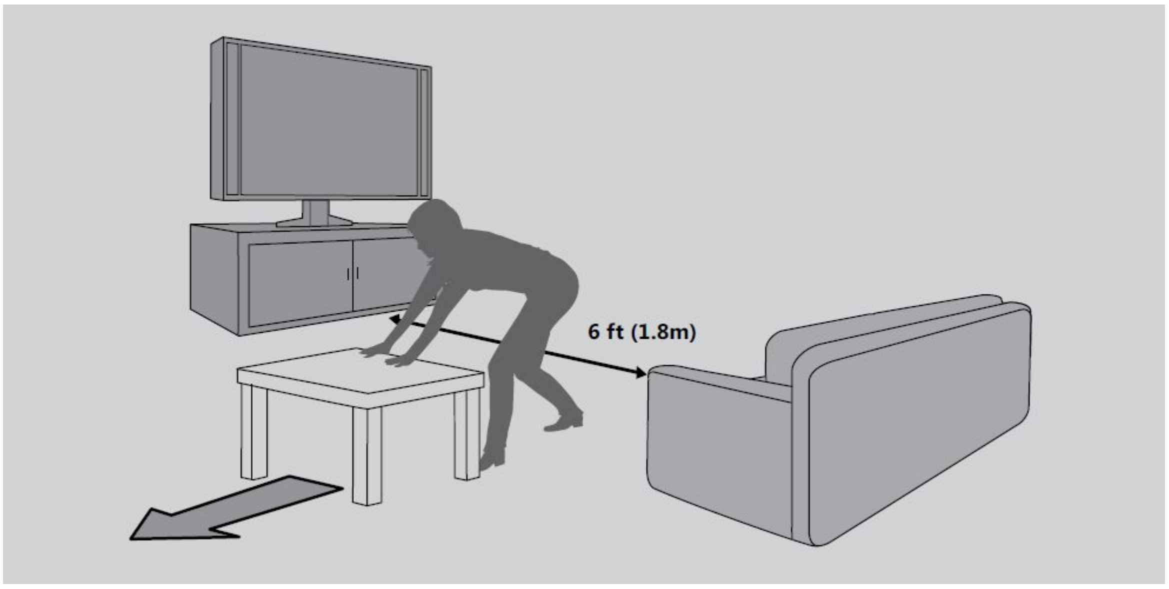

Sold separately*
2 ft–6 ft
(0.6m–1.8m)
* Sensor mounting adapter required for above TV mounting (sold separately)

Note: In smaller rooms, try to position the sensor as close to 6 feet (1.8m) as possible.

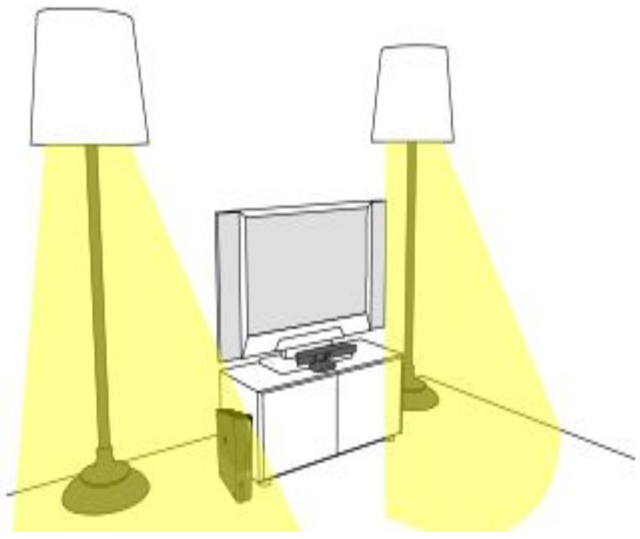
## 5.2.2 Set up your play space

Kinect needs to see your entire body

- Clear the area between the sensor and the players.
- One player: Stand back 6 feet (1.8 m).
- Two players: Stand back 8 feet (2.4 m).
- Make sure that the play space is at least 6 feet (1.8 m) wide, and not wider or longer than 12 feet (3.6 m).
- Make sure the room has bright, even lighting.



## 5.2.3 Room lighting



Make sure your room has enough light so that your face is clearly visible and evenly lit. Try to minimize side or back lighting, especially from a window.
Illuminate players from the front, as if you were taking a picture of them.
Make sure the room is brightly lit.
Some lighting conditions can make it difficult for Kinect to identify you or track your movements.
For best results, avoid positioning either the players or the sensor in direct sunlight.

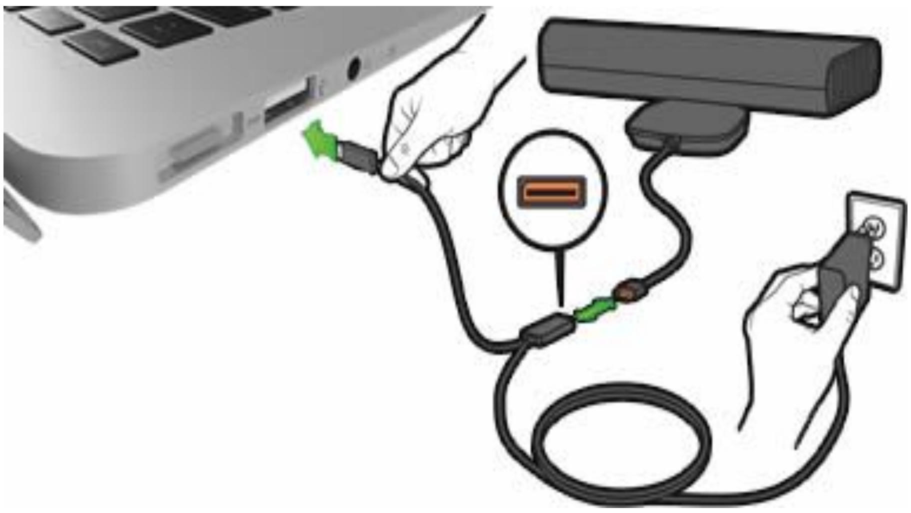
## 5.2.4 Install the software

- Open your internet browser (Chrome, Firefox, Explorer, Safari….).
- Type in your address bar: www.microsoft.com/en-us/Kinectforwindows/.
- Download the latest drivers.
- Plug in your Kinect and wait for the drivers to install.

Note:Here you must download Kinect toolkit 1.7 or 1.8 for the KinectInteraction.

## 5.2.5 Connect the cables

Plug one end of the cable into the USB port on the USB plug of your pc and the other end into an electrical outlet.



# Chapter 6: References

Inside this document bibliography references have the prefix [b] while site links start with [s].

## 6.1 Bibliography

[b1] Jerrett Webb and James Ashley. <<Beginning Kinect Programming with the Microsoft Kinect SDK>>

[b2] David Catuhe. <<Programming with the Kinect for Windows Software Development Kit>>

[b3] Clemente Giorio and Massimo Fascinari. <<Kinect in Motion - Audio and Visual Tracking by Example>>

[b4] Abhijit Jana. <<Kinect for Windows SDK Programming Guide>>

[b5] Rob Miles. <<Start Here! Learn the Kinect API>>

[b6] Ramos Melgar and Ciriaco Castro Diez. <<Arduino and Kinect projects>>

[b7] <<Augmented Reality with Kinect>>

[b8] Jeff Kramer. <<Hacking in Kinect>>

[b9] Rob miles. <<Using Microsoft Kinect in XNA>>

[b10] Jared St.Jean. <<Kinect Hacks>>

## 6.2 Site links

[s1]http://msdn.microsoft.com/en-us/library/dn188671.aspx

[s2]http://code.msdn.microsoft.com/Kinect-SDK-Skeletal-b90f9cd9

[s3]https://github.com/OpenNI/OpenNI/blob/master/Samples/NiUserSelection/SceneDrawer.cpp

[s4]http://social.msdn.microsoft.com/Forums/en-US/kinectsdknuiapi/thread/631aff06-cb4c-42ff-84ee-d895c94c5da4/

[s5]http://social.msdn.microsoft.com/Forums/en-US/459012d7-d51e-460d-aa6a-a1b80c27626c/how-to-lock-in-a-single-skeleton-in-microsoft-kinect-sdk-v-10?forum=kinectsdknuiapi

[s6]http://www.openni.org/docs2/Reference/smpl_user_tracker.html

[s7]http://stackoverflow.com/questions/9387881/creating-a-gesture-definition-for-a-simple-static-gesture-using-kinect

[s8]http://blog.csdn.net/zouxy09/article/details/8161617

[s9]http://blog.csdn.net/dingluseu/article/details/11638683

[s10]http://channel9.msdn.com/Forums/Coffeehouse/Who-Kinect-see-Counting

[s11]http://blog.csdn.net/chenxin_130/article/details/6706630