Shenkar College of Engineering and Design

Faculty of Engineering

Department of Software Engineering

Live Editing System

Final Project

By
Nofar Shaked

Submitted as Part of the Requirements for receiving the

Bachelor of Science degree (B.Sc.).

2013.09.26

Table of Contents

Gratitude

I would like to thank my project advisor, Dr. Amnon Dekel, for mentoring and advising me over the past year and helping me finish this project.

I would also like to thank Shenkar College for providing me with the tools to accomplish this project and the entire degree.

Abstract

Live Editing System is an application that gives the user the ability to create projects, including in them different types of files, such as images, videos and audio, and implement them in the creation of an original video, filmed using the Kinect camera. The Kinect allows the user to control the different elements on screen with body gestures and voice commands. The system includes, beside the editing and recording, an uploading option, to get a video up on YouTube right after filming. All these features on one system provide a single application with all the tools needed in order to film, edit and upload a unique video with special effect and by this saves the time and effort it usually takes to make these kinds of videos.

The goal of this project is to build a simple system, including all the basic tools needed to sustain the system as described. The most important features in the system are:

- controlling the graphic elements on screen using the user's body gestures and voice commands, captured and recognized by the Kinect.
- Recording videos in program.
- Uploading videos to a defined YouTube account and channel.

Adding the ability to manage projects and files was not crucial for this project since it has no scientific contribution.

1   Introduction

1.1   Functional Specification (Overview)

1.1.1   Project Goal
The Project's goal is to create a system allowing users to record and add effects using gestures and voice commands during recording and by so reducing editing time and giving more room for creativity.

1.2   Incentive
The system is designed to replace all programs and tools used to create a video in the cases in which the process tends to be too long for a simple goal. The system overcomes other tools in a number of ways:

1.2.1   Shortcuts
The main goal of the system is to cut down on long, exhausting editing time. The user is given the choice of different shortcuts on using and creating effects during filming instead of implementing them afterward. Not only are the effects inserted into the video naturally, using gestures and voice commands, without the hard work of making it look that way with classic editing, but it also takes no time at all. The editing time is reduced to zero because it is implemented in the recording session, which should be as long as the final product. So a 5 minute video will take 5 minutes to film and edit, which is impossible with today's editing programs. The most important part is that the user watches the video while filming it since he uses the screen to use the controls. That way there is no need to re-watch it and make sure it's okay.

1.2.2   User Friendly
The system is designed to make the video creation process easier and so it has basic and understandable controls, tutorials and manuals to help the user learn about the system and the entire procedure from start to end is done inside this program. From pre-production, through filming and to uploading, it's all in there. There's no need to even log into Youtube. The user is logging into his channel through the system and uploads his videos through it as well.

1.2.3   Problem
Today, video media creators use numerous programs in order to complete a process of creating a video. Taking Youtube partners as this project's main audience, they make a variety of videos, some are more invested and some are less, which their process' time goes from a few minutes to a few months in the making. First writing then casting, filming, directing and editing. A popular Youtube channel could create a 20 minutes video once every two months or a 2 minutes video twice a week. It all depends on the investment in the video. The problem is the need for more videos more often. The need comes from the viewers, wanting to see more, and from the creators, wanting to create a lot more, either for their need for creativity or for the profit from the creation.

1.2.4 Existing Solutions

In the business of filming, you can never rely on only one tool. The variety of tools offered to a client today is unlimited and one can never master enough of them in order to complete a process by himself. It takes a whole team, with each member having his own part in the process. It is important to show what common tools exist today in order to emphasize the complexity of video creation and explain why certain parts should be implemented in the Live Editing System and some shouldn't.

Below are a few examples of the most known and common tools needed for video creation:

1.2.4.1 Camera

It cannot be denied that most of the videos created in the world every day are made with at least one sort of camera (not including animations and other computer or hand generated content), especially today, with more than half the world's population owning a Smartphone with an HD camera built in. To YouTube only, 100 hours of video are uploaded every minute. It is easy to believe that quite a lot of them are filmed using a Smartphone's camera, since it became so common and popular with today's sharing phenomena, but in almost all cases, a video taken with a phone's camera will not have special effects and barely any editing. A Smartphone is designed mostly for instant upload, right after filming and even has special Apps dedicated for video sharing and basic editing (color filters and video speed altering).

Image 1 - A Smarthpone with a veriaty of lences for advanced filming

Looking into the filming industry a little more professionally, some more expensive cameras are in use for higher quality shooting. When considering perspectives, lightings, resolution and more advanced aspects of filming, a professional camera is needed and in a lot of projects a big part of the budget goes to buying a decent camera, capable of capturing the scene as the director wishes.

Image 1 - http://www.businessinsider.com/this-television-commercial-was-filmed-entirely-on-an-iphone-5-2013-4

Image 2 - Sony's new advanced 3D movie HDTV camera with one lense

When looking at the product's main audience, partners, there is a variety of cameras used, but in most cases the creators use a pretty simple equipment for low budget projects (there isn't always a big income from these projects but there is a demand for frequent content upload from the viewers, so low budget pr^{2}ojects are very common).

In this project, it was important to consider what type of camera should be used in development. Considering the need to give the user control with his body movement, it was important to find a tool that can recognize the user's body and then distinguish between his different limbs and turn them into controllers. The solution is a camera with depth recognition, giving the developer a visualization of the scene and the user standing in it in three dimensions. The first thought was using the Intel's CREATIVE perceptual camera for short distance shooting and gentle hand gesture recognition, including individual fingers, but as it is a relatively new product with a small audience (users and developers) it wasn't a smart decision to use it, as it has very limited abilities and as a camera it has low quality and low frame rate, not suitable for filming and sharing creative content.

The system demanded having a camera of high image quality and depth recognition abilities. The most commonly used camera of this sort in the world today is the Microsoft's Kinect. More known as a gaming device for the Xbox 360, it is still used widely as a developers' tool for the PC. Having a big audience of users and developers, it keeps getting more and more updates to its SDK, allowing developers to enhance their creativity with Kinect projects. All programs, created for the Kinect, gave the users the ability to control technologies like never before, releasing them from the controller and turning them into the controller. These abilities are used a lot of times for visualizing the user as the controller on screen, making the controlling look realistic. This is exactly what the Live Editing System needs: giving the users full

Image 2 - http://www.photoclubalpha.com/2009/10/01/sony-invent-3d-movie-hdtv-camera/

control of what is happening around them on screen and making it look good and [3]realistic. Learning from these kinds of systems lead to the different ideas of controls in the Live Editing System.

1.2.4.2 Editing Programs

Editing programs haven't changed a lot in recent years and there isn't a large variety of products out there. The most known editing program is Adobe Premier. Adobe is a major company for design and editing programs. Premiere is a timeline-based video editing software application. Used by many news channels, TV shows and feature movies. This application is very useful as it supports high resolution video editing and a wide variety of video and audio file formats.

Image 3 - Premiere's layout

The problem with Premiere is that you need to use more applications if you want effects and image editing, such as Photoshop and After Effects.

A simpler program, very common among YouTube partners, is Windows Movie Maker. A straight forward editing system, giving the user the simplest tools for editing, such as cutting and pasting video and audio chunks. The Mac version comes with slideshow themes, giving the user a little more options in creation. This programs is mostly used for home made videos but YouTube's low budget videos embrace it as the free program that's just enough for what they need. It also has a sharing option included, which makes it that much more appealing to YouTube partners. This system needs the minimal understanding of editing to be used and anyone can operate it easily.

Image 3 - http://library.creativecow.net/kobler_helmut/FCP-vs-Premiere-Pro/1
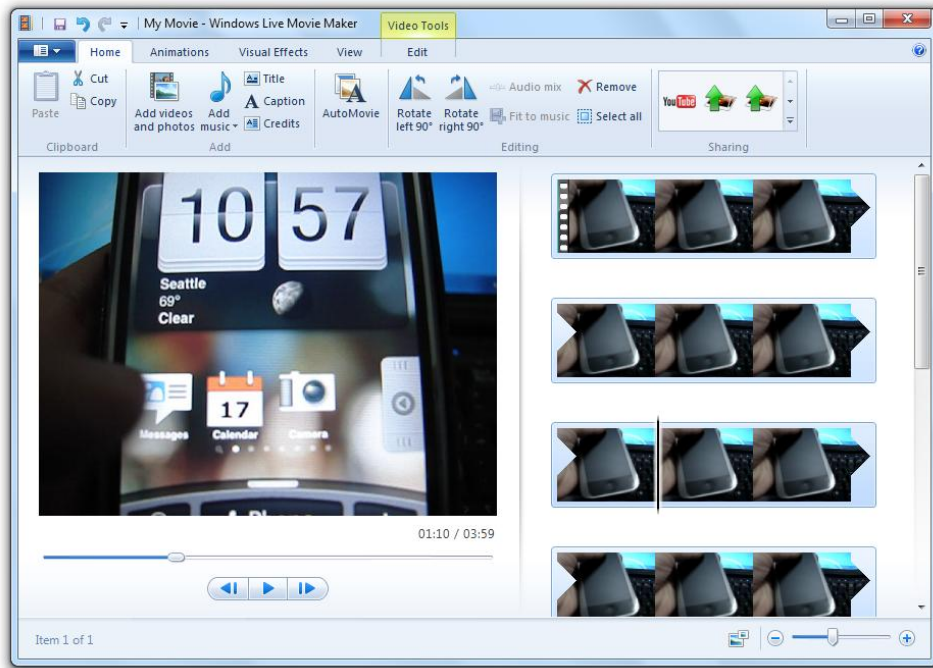
Image 4 - Windows Movie Maker's layout

Windows Movie Maker is simple enough for anyone to use but its abilities are so limited that barely any creativity can be achieved in the editing process.[4]

Adobe After Effects is another expensive gem from the creators of Premiere, but this program is dedicated to adding special effects to videos. This program may not be necessary in most videos on YouTube since it is very advanced and adds mostly cinematic effects used mainly in the movies, but users use it to add special features to their videos to make them look more exciting and invested. This is an even more complicated program than the Premiere and every type of effect needs to be mastered separately.
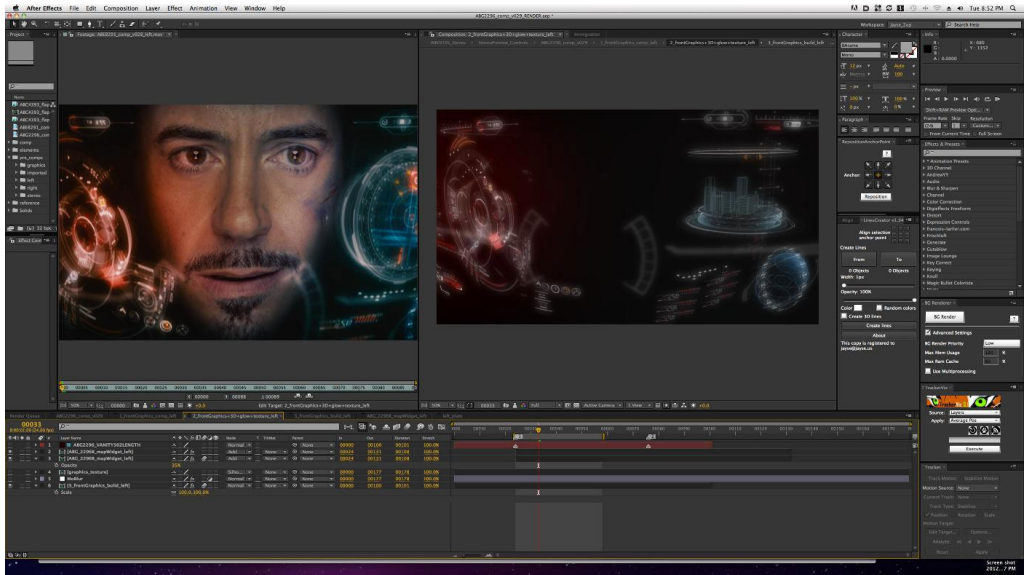
Image 4 - http://www.redmondpie.com/download-windows-live-movie-maker-2009-for-windows-7/

Image 5 - Iroman's HUB and many other effects in the movie were created using After Effects complex system

The users get what they pay for, but there are basically two options for them: pay handsomely and get the best programs in the world that they probably don't need if they are not expert movie editors, or pay little to nothing for a simple program with the same editing powers most phones today have. There is no middle ground for the average user that wished he could have a cheap system with at least the graphic abilities Power Point has, such as adding images, graphic effects, sound effects and some organizing power. Live Editing System tries to include as many basic effects the user needs as possible, with simple controls that don't take weeks to master or hours to apply to a video. The key words here to work by are: simple, fast, natural and creative.

1.2.4.3   Sharing

The largest video sharing site in the world today is Youtube. It is a free to use site where anyone can manage their own channel and upload videos. To share a video on YouTube, all you need to do is set up a channel, press the "Upload" button, appearing on YouTube's main page, and choose a video from your computer (or create a picture presentation). While a video is uploading to the site's database, the user enters identifying details to present the video by, such as a headline, description and sometimes some extra information to help people search for the video (key words and tags). If the user chooses, he can get an email notification when the video is done uploading.

YouTube offers no tools for editing besides adding notes to appear on screen at a certain time on top of the video, that can be used as hyperlinks to direct to other videos, and some image quality enhancement tools. Since YouTube partners are the target audience for the Live Editing System, uploading to YouTube is a major part of the process that should be included in the system and available right after finishing

---

Image 5 - http://community.digitalmediaacademy.org/tag/using-after-effects

editing a video, saving the time of opening YouTube's site and finding the right video file to upload.

1.2.5   Solution

In order to speed up the process, a creator will use the Live Editing System to record himself and use the system's special interface to make the process of adding special effects to the video faster, simpler and more enjoyable (gesture recognition and voice activation while in the process of filming it. Then all that's left for the creator is to upload the video to his Youtube channel, also done using the system.

Following is a detailed example of the system's abilities, at its current state.

1.2.5.1   Recording Screen Layout

The image below shows the entire user interface, including: item menu at the bottom, containing four images and the Start Recording button, and the top bar, used for item control.

Image 6 - Recording screen layout

The entire top and bottom menus do not on the final video so the feed is cropped when recorded.

1.2.5.2 Controls

In order to make an item appear on screen, the user moves his hand over one of the control icons and holds it hover it place until it is activated.

Image 7 - Choosing an item from the Item Menu

The item appears, attached to the selecting hand and dragged by it until it is released.

Image 8 - Dragging an item on screen

In order to release the item to stay on screen, The user spreads their free hand upward to the top bar. A touch for a short time releases the item from the grip.

Image 9 - Touching the top bar with one hand while an item is attached to another

Image 10 - An item is placed on screen

While the item is locked on screen, the user can resize it using both arms, stretched upward to the top bar, moving them farther from one another, in order to grow the item bigger, or bring them closer together, in order to shrink the item.
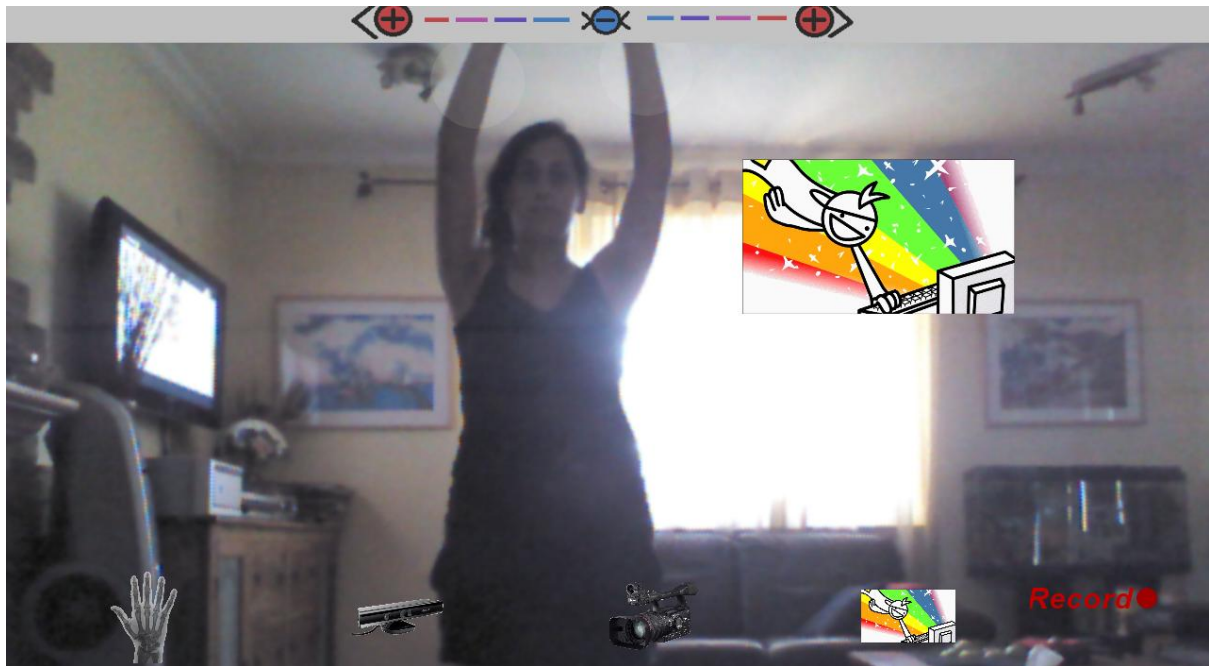
Image 11 - Both hands touch top panel and are drawn closer to shrink the item appearing on screen

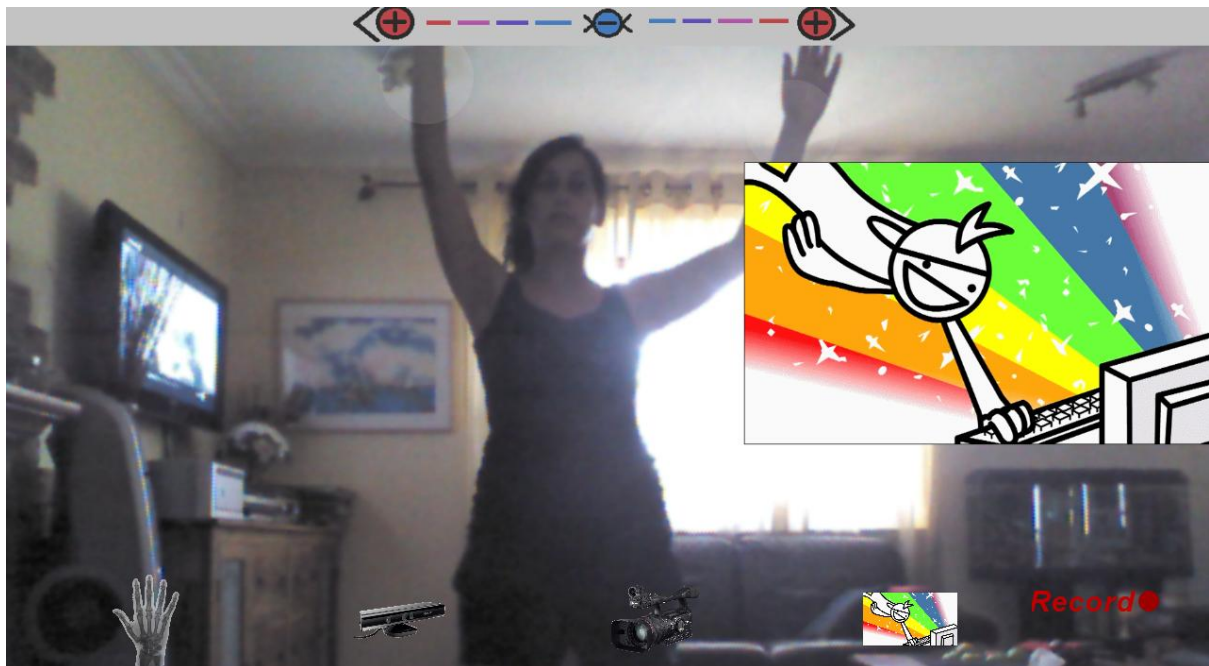Image 12 - Both hands touch top panel and are drawn farther apart to enlarge the item appearing on screen

In order to make the item disappear, the user stretches one arm up to the top bar and touches it for a short time.

Starting and ending recording is activated like all other items, by hovering one hand over the "Recording" button in the item menu (it appears as "Stop" during the recording session).

Image 13 - "Stop" button appears instead the "Record" button during recording session

After finishing recording, the uploading screen appears in which the user can choose either to enter all necessary details before pressing "Upload", record again or going back to the main screen.

Image 14 - Upload screen with desired data inserted into the form

The system, in its current version, allows a user to create a ten minute video, featuring images and videos as controlled items, like in many YouTube videos, in exactly ten minutes (not including choosing and adding the items pre-recording) by inserting all effects during filming and removing editing time completely. Uploading can be done immediately after filming because the user saw the final product on screen while filming it.

1.2.6 Future Development

The system right now includes only the recording and uploading screens. The Project screen was not developed because it was not technologically important for the project's proof of concept. At this point, only image and video items can be used. In the future, special editing effects could be implemented, such as screen split, live feeds, multi-user layouts, etc. Voice activation is also not available in this version but is a necessary element to be added in future versions.

1.2.7 Audience

The system is designed toward Youtube partners that have a need for a single, cheap system, implementing several programs' tools (like Premier and After Effects) to be used quickly in order to create high level videos. Technically, anyone with a Youtube channel could use this system for their own use, even without profit. The system is designed to assist in creating videos of all kinds, such as news, updates, comedy and Vlogs.

1.2.8 Terminology

Kinect - "[A 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"[6].

Gesture – "A gesture is a form of non-verbal communication or non-vocal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with, speech"[7].

Speech Recognition – Translation of spoken words into text.

YouTube Channel – A personal account on YouTube, allowing the owner to upload videos to, sharing them with other people. If a channel is not private, anyone can view any video on it. Creating a YouTube channel is free of charge and only demands agreeing to the terms of use.

YouTube Partners – A YouTube user, owning a channel and a contract with YouTube, saying that the user is required to getting a certain amount of viewers in a certain time and, in return, YouTube will pay the partner for each viewer. A YouTube partner uploads videos regularly, from once a month to a few times a day.

---

[6] http://en.wikipedia.org/wiki/Kinect
[7] http://en.wikipedia.org/wiki/Gesture

Item – The Live Editing System allows the user to choose a few types of files from his machine and include in his projects. These files can be videos, images or audio files. These files can be edited on the Project Screen and save to an Item class, and then presented on the Recording Screen, where they are controlled by the user's motion and voice.

.

2   Architecture

2.1   Description

The system includes 3 screens:

Project
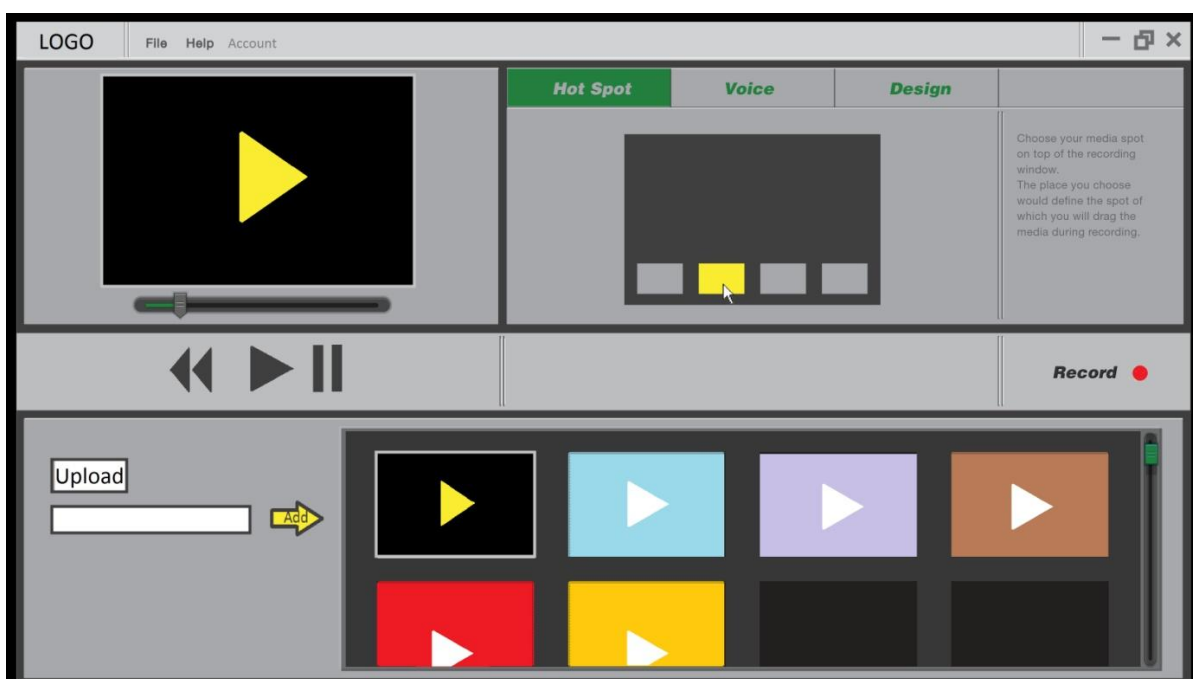
Recording

and Uploading.

When a user activates the program, he is directed to the Project screen, in which he can create, load or save projects. Each project is composed of special elements, which can be images, videos or effects, with special properties for each of them, determining how each item is used or appears in the recording session. The items are loaded or created by the user who edits their properties. This screen contains a record button, leading to the Recording screen.

The Recording screen shows a display of the Kinect's RGB feed on full screen. Besides that, there appears an Item Menu which contains the items programmed to appear in it from the Project screen. These items can be activated using hand gestures, captured by the Kinect depth camera. There is also a Start Recording button which activates the recording session. All user actions can be activated before and during a recording session, the difference is that during a recording session, all data appearing on screen is recorded to a video. During the recording session, a Stop Recording button appears on screen, which, when activated, stops recording and redirects the user to the Uploading screen.

The Uploading screen is a form for editing the last recorded video's uploading details. If the user hasn't registered his Youtube connection details yet, he does that here. The form also requires a name and description for the video and extra information, affecting the appearance of the video in YouTube's search results, if the user is interested. The form contains 3 buttons: Upload (which uploads the video to the user's Youtube channel with the details entered in the form), Record again (starts another recording session for the user to create a new video because he is not pleased with the last one) and Back to Project Screen (directs back to the Project screen, where the user edits the project before trying to record again. Whichever the user chooses, the video is saved anyway to the project's folder for him to use.

2.2 System Users and Roles

The system is useful to anyone who has a Youtube channel or just wants to create videos for their own entertainment.

2.2.1 User Types and Roles

2.2.1.1 Youtube Partners

These users are required to supply new videos to their channel as part of their contract with Youtube. These users can make a variety of videos, such as news updates, commentaries, Vlogs, etc. These are the kind of videos which needs to be uploaded regularly so they must be made fast. Youtube partners can use Live Editing System and connect their home or studio directly to their channel and create new content quickly.

2.2.1.2 Online Tutors

The system has plans for a live feed option, but even without it, users can make educational videos using the system and upload them from their machine to the right sites. The tutors can use graphs, forms, equations, images, documents etc. and present them in their videos to help them teach any material they want. A history

teacher, for example, could show maps and pictures to pass classes to his online students.

2.2.1.3 Update Sites

A lot of sites provide their followers with updates on different matters, such as news sites. These sites could easily use the system to create update videos, presenting stories, interviews and news pieces. All update videos today include a presenter and different windows to show images or videos, either small ones, next to the presenter or ones spread on the entire screen. The system's base features supply these demands.

2.3 System Weaknesses

The system depends mainly on the user, as he is the one to design his own projects, supply them with items and eventually create the video. Tutorials and manuals are needed for the user to avoid problems.

2.3.1 Youtube Account

If the user wants to upload his videos to Youtube, he must have a Youtube account and open up a channel. Without all this and an internet connection, the user can only save videos to his machine and not share them on Youtube.

2.3.2 Kinect's Gesture and Voice Capture

Kinect works with the user's body as a controller, capturing his movements and limbs positioning. The user must create a wide and clear environment to work on, for the Kinect to easily capture him and for himself, to have a comfortable area to move in. For voice commands, the user must talk clearly for the Kinect to understand him. Filming yourself in a small area with a lot of background noises could harm the effectiveness of the system.

3 Design (Software Requirements Specification)

3.1 System Structure

The system is composed of five main libraries and APIs:

3.1.1 Kinect for Windows SDK (Microsoft.Kinect)

Downloadable from the Windows developer downloads[8]: "The SDK includes drivers for using the Kinect for Windows sensor on computers and devices running Windows…".

The Kinect is the eyes and ears of the computer and the Kinect for Windows SDK is the brain that helps it understand the input.

The Kinect has three hardware innovations, working together within the sensor:

- **Color VGA video camera** – Collects RGB data from the environment, detects three different color components: red, green and blue, and aids in all kinds of detection features, such as facial recognition.

[8] http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx

- **Depth sensor** – An infrared projector scatters a large number of tiny dots the area in its view and a "monochrome CMOS (complimentary metal-oxide semiconductor) sensor"[9] captures the scene as a pattern of dots (see image 1). The SDK compares the image from the camera with the pattern that it projected, to calculate the distance of each point from the sensor. This allows the program to get a 3-D view of the room, regardless of lighting conditions. See image 15.
- **Multi-array microphone** – The Kinect has an array of four microphones that can isolate the voices of users from the noise in the room. The SDK can use this to recognize users, place them in the room relatively to the Kinect by sound, and receive voice commands.

Image 15 - Kinect's infrared pattern on a view of a hand

In Conclusion, the SDK gives the developer access to the information the Kinect collects from its environment for him to use in his desired program.

3.1.2 Coding4Fun.Kinect.Wpf

The system is built using WPF (Windows Presentation Foundation), Including graphical layers in the program, which are necessary for displaying the Kinect's video input and the different effects and elements used by the user. This library gives the programmer a variety of tools, commands and sample codes for controlling the Kinect and using its features easily in the program. This saves time in the making of the program and is useful for recording Kinect data if needed.

3.1.3 SpeechLib

This library supplies commands for speech recognition. The system uses this for adding user defined words to a vocabulary, used as an agglomeration of usable commands in the recording session(see chapter two).

---

[9] http://electronics.howstuffworks.com/microsoft-kinect2.htm

3.1.4 YouTube Data API (Google.GData.YouTube)

"The YouTube Data API allows client applications to retrieve and update YouTube content in the form of Google Data API feeds"[10].

The user uses the system for creating original videos, which he wants to share after the making. Entering his YouTube username and password, the uses gives the system access to his account for uploading wanted videos. The system uses this library and a developer key, specific to this system, to perform all tasks needed for uploading, such as requesting connection to the user's account using his details, specifying settings for an upload, uploading videos to the user's account and getting updates regarding the upload's status as it accurse.

3.1.5 BytescoutScreenCapturingLib (Screen Capturing SDK)

This library supplies the developer with a free desktop capturing tool that can be implemented in code and used inside the program. The system uses this tool to record a specific area of the screen during the recording session (see the chapter two), capturing the Kinect's video data presented on screen and the, the visual items that the user activates and controls, and any sound captures by the Kinect, into an AVI file that can later be uploaded to YouTube.

3.2 System Features

The features described here are only partly implemented in the current system version but are planned for the full product.

Implemented features are marked with a "*".

3.2.1 Creating and Editing a Project (Project Screen)

3.2.1.1 Description and Priority

User wants to create a new project, add items and effects to the project, set the menu and commands for the recording, manage other projects and connect to his Youtube accounts. Priority = High.

3.2.1.2 Stimulus/Response Sequences

3.2.1.2.1 Project File

Stimulus: User creates new Project.

Response: As the user names the project, the system checks if that name is already used (at least in the same location), If it does, a message will alert the user and ask him if he wishes to overwrite another project with the same name or not. A new project is created in the Projects folder, without any items, effects or presets in it (if another project is already open and its changes weren't saved, a message will appear, asking if the user wants to save the changes, the options will be: Save/Don't save/Cancel)..

---

[10] https://developers.google.com/youtube/

Stimulus: User saves project.

Response: Name checking, like in project creation. A project, with all its properties, items and recent changes, is saved to the Projects Folder (in its own folder). When done saving, a note appears in the bottom of the window, saying save was done successfully.

Stimulus: User loads existing project.

Response: An existing project opens up on the screen, with all the properties and items in its folder (if another project is already open and its changes weren't saved, a message will appear, asking if the user wants to save the changes, the options will be: Save/Don't save/Cancel).

3.2.1.2.2 Items and Project Options

Stimulus: User loads an item or effect to the project.

Response: A new Item object is created and added to the project's Item List.

Stimulus: User selects an item from Item Window (pressing on a thumbnail, representing that item).

Response: Item properties open up in the editing section of the screen. In that section, the item can be previewed, its position on the recording menu can be selected and the commands that operate it on the recording screen can be edited.

Stimulus: User edits an item through the Item editing section (add a voice command or adds it to the recording screen window or edit its position on screen).

Response: Item's new properties are saved to the Item Object properties (will be later loaded and implemented in the recording session).

Stimulus: User presses on a trash can symbol on the item that is selected in the Items Window.

Response: Item is removed from Items List and its Item Object is deleted.

Stimulus: User presses the Start Recording button.

Response: Project Screen is closed and user is directed to Recording Screen.

Stimulus: User presses the help button.

Response: A help window will appear.

Stimulus: User presses Connect to YouTube Account.

Response: A pop-up window will appear in which the user needs to fill a form with his YouTube username and password. After user fills all details and presses Connect, the system tries to connect using the details. If it succeeds, a message appears saying Connection Successful. If the connection fails, a message appears

saying Connection Failed and the user will remain in the form window in order to try again with a different username or password.

3.2.1.2.3 Functional Requirements

new_project: The system checks if there isn't a project open. If there is and there have been changes that were not saved, a pop-up will appear, asking the user if he wants to **Save** the changes and create the new project, **not save** the changes and **create** a new project or **cancel** and not create a new project. The system will follow the user's choice.

save_project: The system updates all the temp changes to the project's folder.

load_project: The system checks if there isn't a project already open. If there is and there have been changes that were not saved, a pop-up will appear, asking the user if he wants to **Save** the changes and load the other project, **not save** the changes and load the other project or **cancel** and not load the project. The system will follow the user's choice.

add_new_item: Browsing screen appears from which the user selects an item from his computer. The item is added to the project's folder and is shown in the Items Window.

remove_item: A pop-up window will appear, asking the user if he is sure he wants to remove the item from the project. The system will follow the user's choice.

edit_item_voice_command: The user enters a word or phrase that will be added to the project's vocabulary and will activate the item in the record session.

edit_item_menu_position: The user may choose a position on the recording menu in which a certain item will be placed. That position is saved in the Item Object properties.

edit_item_screen_position: The user may choose a position on screen on which the item will appear once user commands it during recording session. That position is saved in the Item Object properties.

switch_to_record_screen: The Project Screen is closed and the user will be directed to the Record Screen.

project_help_menu: A help window will appear with tutorials on how to manage projects.

connect_to_youtube: A connection window will appear with a form. The user

needs to fill in his Username and Password. After the user presses Connect, The system will try to set a connection with Youtube using the user's Username and Password. If it fails, it will alert the user and let him try again. If the connection is successful, a message will alert the user and let him continue. The connection will remain as long as the project is   open.

3.2.2 Recording Session (Record and Upload Screens)

3.2.2.1 Description and Priority

The user can open a help menu or record a video. The recording starts a session in which items can appear and be operated using voice commands and hand gestures (captured by Intel's Depth Camera). Recording can start and end using mouse, voice command or hand gesture. After a recording is finished the user can review the video, publish it, delete it or re-record it.

3.2.2.2 Stimulus/Response Sequences

3.2.2.2.1 Windows redirections

Stimulus: User Presses the Help button.

Response: A tutorial window on how to use the recording and editing system is opened

Stimulus: User Presses the Return to Project button.

Response: The Recording Screen closes and the user is redirected to the Project Editing Screen.

3.2.2.2.2 Recording commands

*Stimulus: User presses Start Recording OR uses the voice command OR uses the Start Recording Hand Gesture.

*Response: The screen shows a short countdown to the beginning of the recording session, the Start Recording button is changed to Stop Recording and the recording starts. The Help buttons is removed.

*Stimulus: User presses the Stop Recording button OR uses the voice command OR hand gesture for stopping the recording session.

*Response: Recording stops and a new window appears in which user can choose what to do with the recently recorded video

3.2.2.2.3 After Recording is done

*Stimulus: User enters a Header and Description for the video in the matching fields. The user can also enter Tags and Key Words. User then presses the Upload button

*Response: The system uses the details the user entered to send an upload request to the user's YouTube account. The system alerts the user about the upload status.

Stimulus: The user presses the Record Again button.

Response: Upload screen is closed and the user is redirected to the Record Screen.

Stimulus: The user presses the Back to Main Screen button.

Response: A popup asking the user if he's sure he doesn't want to upload the video appears. If the user approves, the window closes and so does the Upload Screen. The user is directed to Project Screen.

3.2.2.2.4 Items Controls

*Stimulus: User moves his handover an item in the Item Menu.

*Response: Item is activated and appears on screen, attached to the activating hand and following it. Activated Item is now on Drag Mode.

*Stimulus: The user moves his free hand to the top Control Area and activates it,

*Response: The item is no longer attached to the user's hand and follows it. The item is no longer on Drag Mode.

*Stimulus: The user lift both hands up toward the Control Area and pulls them apart or toward each other.

*Response: Item Changes size by distance between hands. It remains at last size when the user removes his hands from the Control Area.

*Stimulus: The user lifts one hand toward the Control Area and activates it.

*Response: The activated item disappears from screen.

*Stimulus: User uses voice command saved for a certain item X.

*Response: Item X appears on screen on its default (or set) position (if is a sound type, it is simply activated).

3.2.3 Functional Requirements

record_help_menu: A tutorial window appears to help the user learn how to use the different features in the Recording Screen.

switch_to_project_screen: System redirects the user back to the Project Screen.
start_recording: System records everything on the screen (camera input and items that appear. Doesn't show the Items menu. While recording, The Start Recording button is replaces with a Stop Recording button. **stop_recording**: System stops recording and saves video to the Project Folder. User is directed to the Upload Screen.
return_to_record_screen: Upload Screen is closed and user is redirected to the Recording Screen. **return_to_project_screen**: Upload Screen is Closed and user is directed back to the Project Screen.
KinectButton(num)_Clicked: Selected item (num represents the position of the item in the Item Menu) appears on screen and is attached to the hand that activated it. **touchup_Clicked (one hand, while item is dragged)**: Item is no longer attached to a hand and remains at last position. **touchup_Clicked (one hand, while an item is on screen)**: Item disappears. **touchup_Clicked (two hands, Moving hands apart or closer)**: The hands' positions are checked as long as they are both on the Control Area. If they draw apart, activated item is enlarged. If they move closer to one another, activated item gets smaller. **speechRecognized**: The system recognizes a word from the project's vocabulary and activates a matching item, item appears on screen in predefined position.

3.3 System Design

3.3.1 Hardware and Communication Interfaces

28

③

Monitor

L.E.S Running on a PC

④

YouTube Server/Account
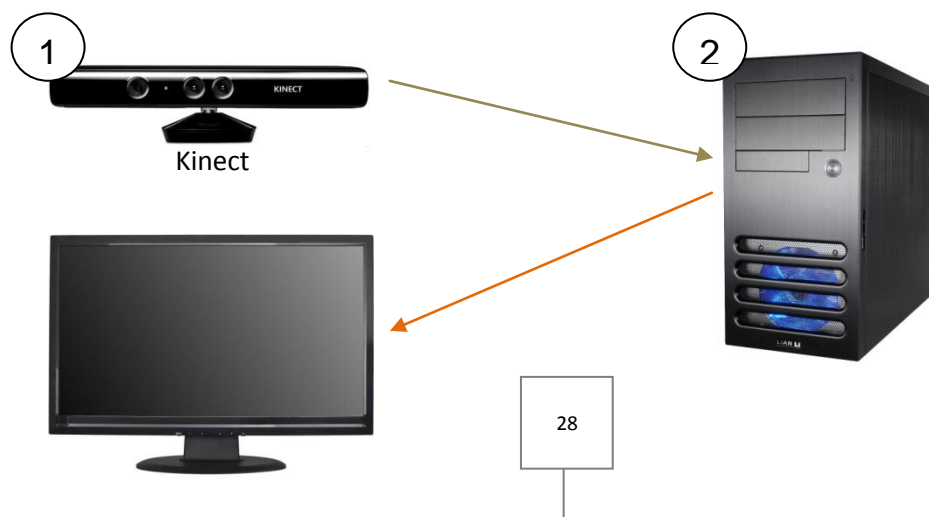
3.3.1.1 Camera

Throughout the process of creating the system, a few hardware changes were made. Using a depth camera was an early decision, for advanced capturing of the user's body. The first camera tested for the task was Intel's CREATIVE perceptual camera, designed for close range capturing of hands, including fingers. Having a limited technological power, it was decided to replace the CREATIVE camera. A simple web cam technology was considered, using image processing instead of the depth feature. Developing to a web cam could open up the system to a large audience; given most people today have a web cam installed already on their computer. The problem here was that most tools for image processing, especially in video, cost money or are used only for studies in the field and not for free use. Developing this tool especially for this system would have required such an effort and time, just this part would be considered a project on its own.

Choosing the Kinect was important for this project to succeed, given its huge base of libraries, forums, tutorials, examples and users, making it the most natural choice for an educational driven project. The reason it wasn't chosen in the first place is that using it required changing the entire design of user activation, originally directed at close range for precise hand movement capturing, but as the camera changed, so did the purpose of the system, now used for wider shots of the user and using the user's arms as controls, instead of palms and fingers.

3.3.1.2 Computer

A PC machine runs the L.E.S system while having the Kinect plugged in, drawing RGB input from it to show on screen and using its depth input and the Kinect SDK to simulate the control of the user over graphical items. During recording session, all graphical info on screen and audio are recorded to a video file, saved in the project's folder. The system uses the computer internet connection to send the file to YouTube and adding it to the user's channel.

3.3.1.3 Monitor

The monitor is located close to the Kinect in order for the user to be able to see himself on screen while being recorded and to always look like he is staring right at the camera. A common problem with having cameras connected to a computer is

that while being filmed, the user tends to look at the screen and not directly at the camera, causing him to look unfocused while being filmed. In order to prevent that, the camera must be placed close enough to the screen's center so the user's eyes can be focused on screen while still look like they are focused on the camera.

3.3.1.4 YouTube Servers/Account

Having YouTube partners as the system's main audience, connecting the system to YouTube and the users' accounts was a crucial goal to reach. Part of the thinking behind the implementation of this feature was about saving more time in the process. Including the uploading session in the system to save the users the trouble of connecting to their accounts on the site, and saving the login data in system for the users to quickly upload again in every project without re-entering their username and password every time.

3.3.2 User Interface

The System splits to three screens: Project Screen, Recording screen and Uploading screen.

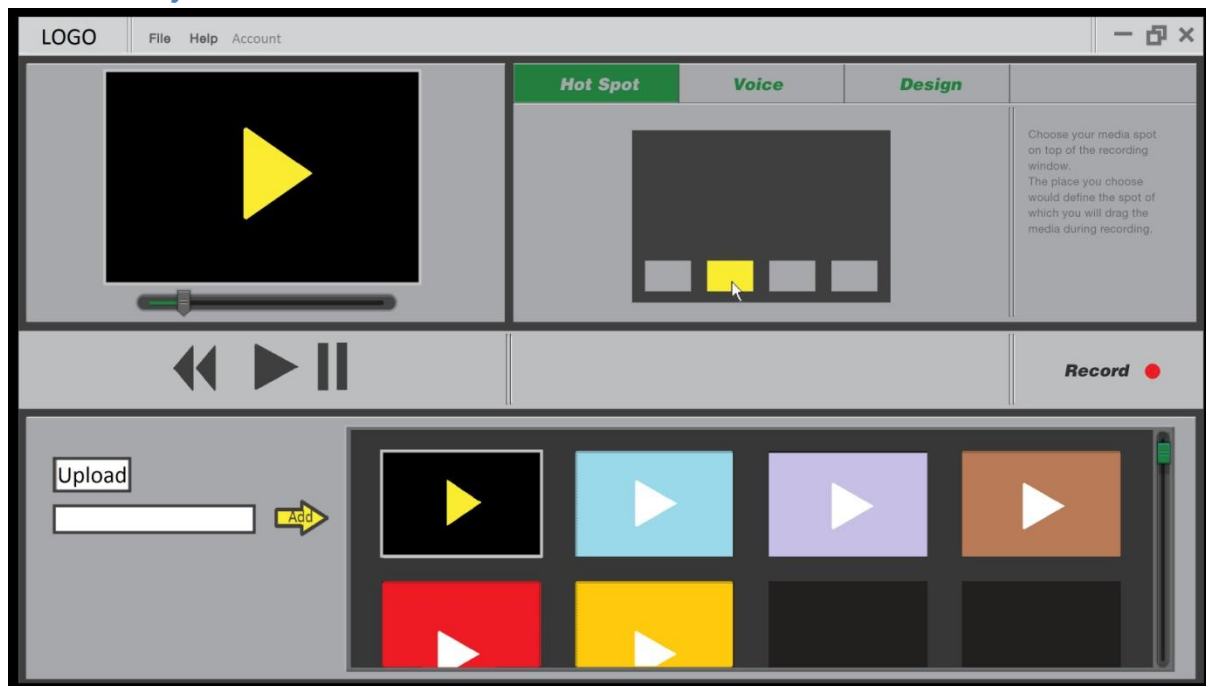### 3.3.2.1 Project Screen

Image 16 – Early design of the Project Screen interface

This screen is meant to manage a project's items list. As a user creates a project, they can edit their YouTube username and password for future uploads, via the "Account" tab or edit the project.

3.3.2.1.1 Adding Items to the Project

Clicking on the "upload" button allows the user to choose items from their machine. These items can be images, videos or sound files. Each file added to the project is inserted to the project's items list, presented as a thumbnail on the Project Screen item window. The items list data is saved in a text file inside the project folder. This file contains for each item its position on the machine, file type, position to appear on screen, position on the items menu on the recording screen and voice command for activation.

Another type of item planned for development is an effect. An effect is not presented as a file, it is a graphical effect chosen from the system. Some of the effects planned for the system are:

- Split screen, for the type of videos that show the presenter in a small window in the corner of the video, on top of another video or live feed of another camera or a computer screen. This type of effect is usually used for narrating videos.

- Avataring, meaning replacing the presenter with an avatar, usually a three dimensional model with a rig connected to the user's skeleton, created by the Kinect's camera and SDK. In the same way, all kinds of three dimensional items can be added and controlled by the user's skeleton. These types of effects are used commonly and take a lot of time to implement in videos, the Kinect's abilities save a lot of time for animators and editors by placing and animating the models for them. These effects are used in a lot of Kinect games and programs.

3.3.2.1.2 Editing an Item

Pressing on an item thumbnail in the item window opens up a preview of that item in the preview window and its properties in the properties window. In the properties window the user can:

- Define how to activate the item in the recording session, choosing between placing it in the item menu, appearing at the bottom of the recording screen, or choosing a voice command for activation during recording session.
- Choose a location on screen for the item to appear on activation by voice command.
- Add a simple extra design for the item, as a border or added text.

3.3.2.1.3 Moving to the Recording Screen

Pressing on the "Record" button on the project screen directs the user to the recording screen.
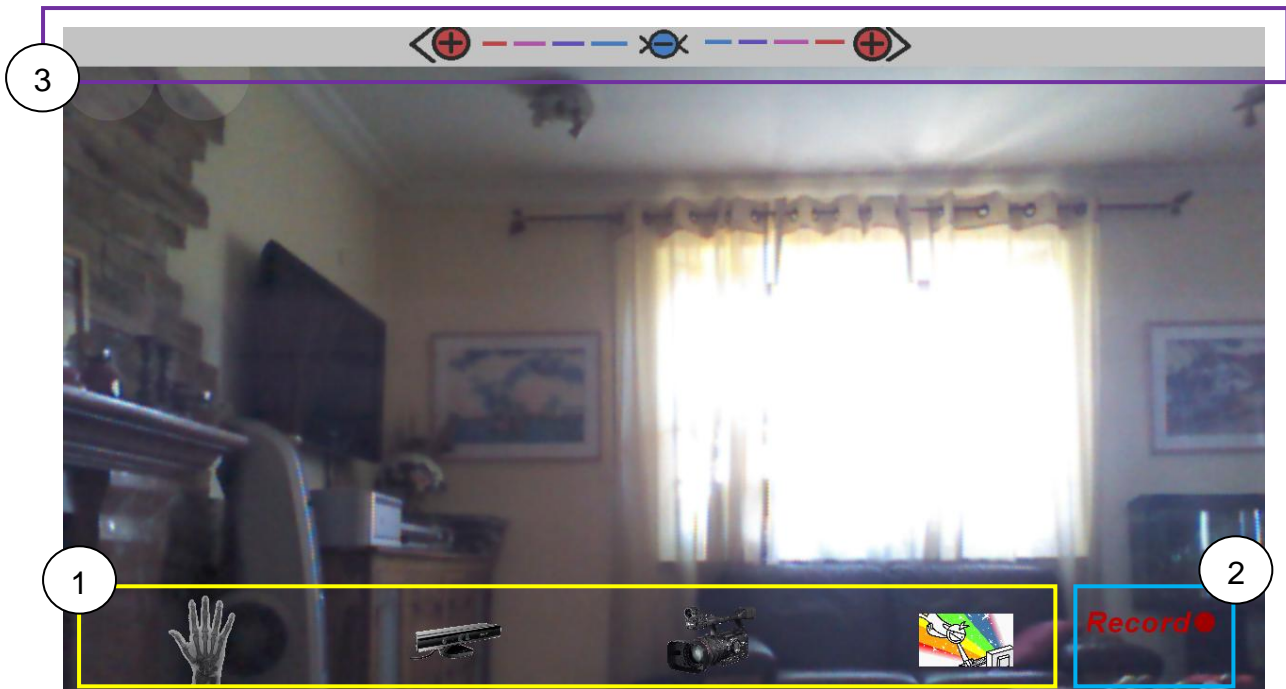
3.3.2.2 Record Screen

The recording screen development was the main goal in this project, creating a proof of concept for the idea of a live editing system with special effects controlled by the user's body. On this screen is a live video feed from the Kinect's RGB camera and on top of that the different tools the user can use to create a unique video.

[Note: A help button was planned to be added to this screen, containing links to manuals and tutorials on using the different tools on this screen, but was not implemented as it wasn't necessary for the proof of concept. A user manual is included in this paper.]

3.3.2.2.1 Item Menu

At the bottom of the screen appears a menu with up to four items. These items were selected on the Project Screen to appear on this menu in their specified positions. A user can activate an item from this menu by hovering one hand over it for less than a second, until the item appears larger, attached to the choosing hand. The Item can be attached to either hand, depends on which one hovered over it. The items on the menu appear as images but could also represent videos that will start playing on activation.

Notice that other items could be activated beside those on the menu, using voice commands, defined in the Project Screen.

3.3.2.2.2 Record Button

The record button can be activated like the items in the items menu, by hovering any hand over it for a short while. Activating this button will start recording session, in

which all audio and all graphic data on screen inside the recording area will be recorded to a video file (recording area is a large area of the screen that does not include the tools appearing on screen). On activation, the button changes to "Stop Recording" button, which can be activated the way by will stop the recording, save the video file and open up the Uploading Screen.

3.3.2.2.3 Top Bar Controller

The top bar can be used in a few ways, giving the user more control over items appearing on screen.
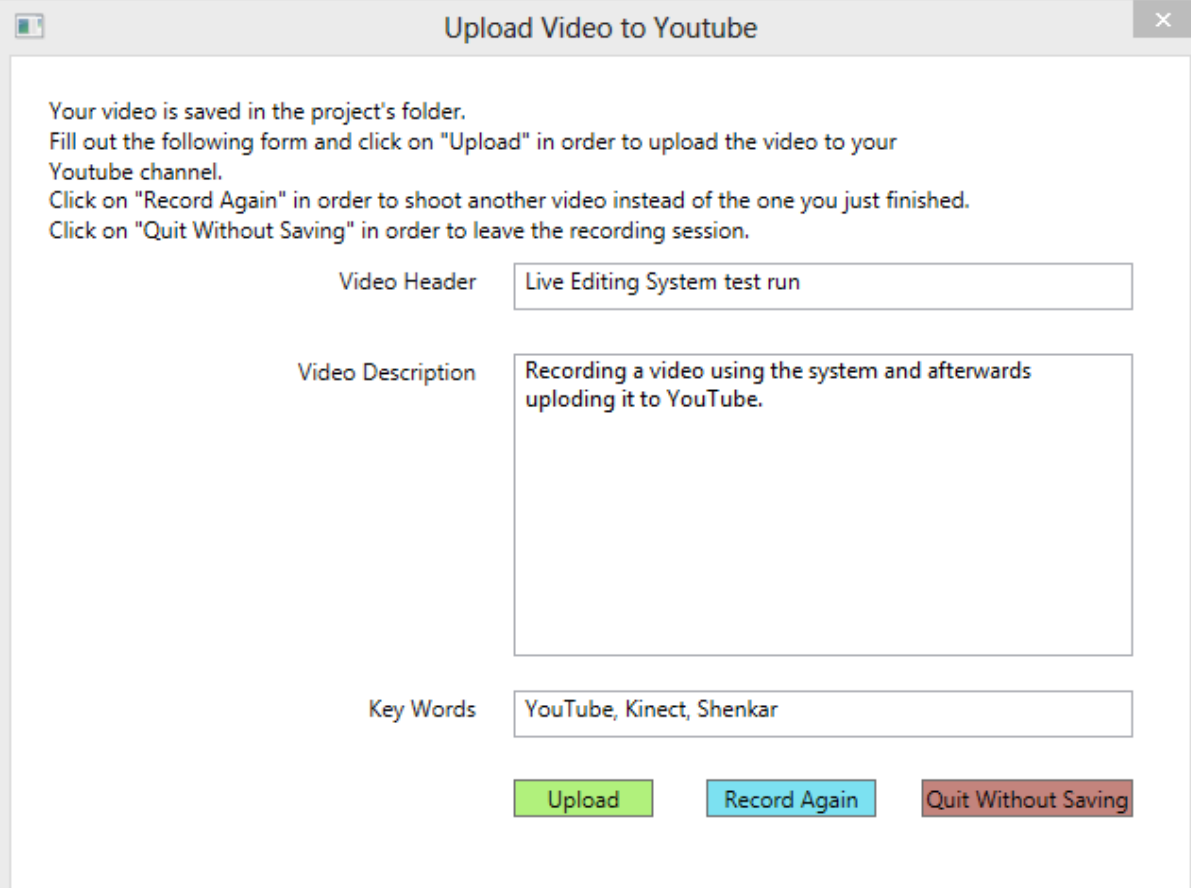
- Locking an item – As a user chooses an item from the item menu, that item will stay attached to the activating hand, dragging it across the screen. In order to release the item and keep it locked on one spot, the user stretches up one arm (preferably not the one dragging the item) , touching the top bar (hovering over it for a short period of time). This releases the item.
- Removing an item – After locking an item on screen, the user will want to make it disappear after a while. To do that, the user touches the top bar again, with one hand. The same action will remove a locked item.
- Resizing an item – in order to enlarge or shrink an item locked on screen, the user will stretch both arms up to the top bar and move them apart from one another, to enlarge, or closer together, to shrink.

This control bar was located at the top and not one of the sides because reaching to the side with an arm forces the user to move too much, even take a few steps, since the Kinect's camera field of view is quite wide. This interrupts the user and breaks his concentration, while stretching his hands up is much easier, can be done from anywhere on screen and this location is basically in the same distance from the body's center as the item menu.

3.3.2.2.4 Other Non visual Controls

As mentioned earlier, this screen includes voice commands. These commands are words or sentences defined in the Project Screen for certain items. Each of these commands is added to the project's lexicon. Saying any of these commands on Recording Screen will activated the appropriate command, such as activating an item and placing it on a pre-defined coordination on screen (also defined on the Project Screen) or starting/ending recording.

3.3.2.3 Upload Screen

Image 188 – Upload screen layout

The system includes the uploading part in order to simplify the process and add an immediate sharing option. That's why the Upload Screen is the simplest one, containing a simple form with all the basic fields needed for uploading a video to YouTube. A simple guide, appearing at the top of the window, provides the steps to using this form.

The user is given a choice of either to upload the recently recorded video, record another one or quit without uploading the video.

4 Application

This chapter will specify the different parts building the system.

4.1 Classes

4.1.1 Main Window

This class builds the Recording Screen layout and manages its entire interface and controls.

- This class uses the Kinect SDK and other Kinect tools and libraries.
- On loading, the item file is scanned, retrieving all items for the items list, placing the right items in the items menu and adds all voice commands to the project's lexicon. In the following code the item file is loaded and each row of it is extracted and set into a new GItem, which is an item to be used in the recording session. Each new GItem is added to the items list and set in the items menu if it should appear there.

```csharp
using (StreamReader sr = new StreamReader("C:\\Users\\nufar\\Desktop\\Final Project
DAMN IT\\folderForSaveFiles\\file.txt"))
            {
                while (sr.Peek() >= 0)
                {
                    string str;
                    string[] strArray;
                    str = sr.ReadLine();

                    strArray = str.Split(',');
                    GItem gi = new GItem();
                    gi.setPath(strArray[0]);
                    gi.setType(strArray[1]);
                    gi.setScreenPositionX(strArray[2]);
                    gi.setScreenPositionY(strArray[3]);
                    gi.setMenuPosition(strArray[4]);
                    /*new item in item menu*/
                    if (gi.getMenuPosition() > -1 && gi.getMenuPosition() < 4)
                    {
                        itemMenuPaths[gi.getMenuPosition()] =
    String.Copy(gi.getPath());
                    }
                    gi.setVoiceCommand(strArray[5]);
                    /*new voice command*/
                    if (gi.getVoiceCommand() != "")
                    {
                        voiceCommands.Add(gi);
                    }

                    _itemsList.Add(gi);
                }

                 sr.Close();

                 /*set item menu icons*/
                 if (!itemMenuPaths[0].Equals("BlackButton-Active.png"))
                 {
                     kinectButton1.ImageSource = itemMenuPaths[0];
                     kinectButton1.ActiveImageSource = itemMenuPaths[0];
                 }
                 else
                     kinectButton1.ImageSource = "";

                 ...//other buttons are set as well…

            }
```

- The speech engine saves different commands as voice commands, each activating a different function.
- Each of the buttons on the items menu is monitored for "clicks", activating the right item on hovering. Same for the record button.
- Each hand position is monitored at any moment and on activation of any button, the activating hand is monitored as well.
- On this class exist two items, dynamicImage and dynamicVideo. Both are transparent as long as no item is active. When an image is activated, the dynamicImage is changed to according to the image file the item represent

and this dynamicImage is controlled on screen by the user and gets transparent as the user closes the item. The same for the dynamicVideo, only for video files. As you can see in the following code sample, the first button on the item menu was clicked. The functions checks which hand activated the item and set the rightActivetedItem flag for true if the right hand activated or false otherwise. It then checks which of the items on the items list is represented by this button. The path of the item file is inserted as a bitmap image to the source of the dynamicImage, its default size is reset and it is relocated outside of the screen view before receiving new coordinates from the controlling hand. After setting the dynamicImage, dragFlag is raised to confirm that the item is now dragged by a hand.

```csharp
void kinectButton1_Clicked(object sender, RoutedEventArgs e)
 {
     foreach (GItem gi in _itemsList)
     {
         if (IsItemMidpointInContainer(kinectButton1, rightEllipse))
         {
             rightActivatedItem = true;
         }
         else
         {
             rightActivatedItem = false;
         }

         if (gi.getPath().Equals(itemMenuPaths[0]))
         {
             dynamicImage.Source = new BitmapImage(new Uri(@gi.getPath()));
             dynamicImage.Width = 230;
             dynamicImage.Height = 310;
             Canvas.SetLeft(dynamicImage, -500);
             Canvas.SetTop(dynamicImage, -500);
         }
     }
     dragFlag = true;
 }
```

- The top bar has three uses:
  - Locking an item by lowering the drag flag if the flag is raised and one hand is hovering over the top bar.
  - Removing an item from screen if it is locked and the user hovers one hand over the top bar (by lowering the dragFlag and setting the dynamicImage and dynamicVideo position outside of the screen view.
  - Resizing an item by checking distance differences between both hands touching the top bar. A zoom parameter is calculated by the movement of the hands, setting farther or closer while stretched up. This parameter is added to the Scale parameter of the active item. This

parameter could be either positive or negative, resulting in enlarging or shrinking or the item. In the following code, the parameter st represent the dynamicImage render transform, meaning it is a parameter with control over the appearance of the rendered item dynamicImage, also controlling its scale. The if statement checks if the ratio between the difference between the two hands before this test to the difference between the hands as they are at the current moment is not 1. If it is larger, then the hands drew apart, meaning the user wanted to enlarge the item. This sets the zoom parameter to +0.03. if the ratio was smaller than 2, then the hands were drawn closer, meaning the user wanted to shrink the item. This sets the zoom parameter to -0.03. The st parameter scale on X and Y is added the zoom parameter, resulting with the new scale. The current hands positions are set to the old zLeftX and zRightX, for the next test. This test runs over and over until the hands leave the top bar.

```
if (Canvas.GetTop(rightEllipse)< 100 && Canvas.GetTop(leftEllipse)< 100)
{
        double zoom = 0;
        var st = (ScaleTransform)dynamicImage.RenderTransform;
        if((zRightX - zLeftX) / (Canvas.GetLeft(rightEllipse) -
        Canvas.GetLeft(leftEllipse)) != 1)
              zoom = (zRightX - zLeftX) / (Canvas.GetLeft(rightEllipse) -
        Canvas.GetLeft(leftEllipse)) > 1 ? -.03 : .03;
         st.ScaleX += zoom;
        st.ScaleY += zoom;

        zLeftX = Canvas.GetLeft(leftEllipse);
        zRightX = Canvas.GetLeft(rightEllipse);
}
```

The upload screen is a simple form, each entering into a field is saved to a parameter. On pressing "upload video", these parameters are entered to a Video object, which is a YouTube class used for uploading. A request for an upload is made and a status update is received every second until the file is done uploading. The following code shows the lines for requesting an upload and status update.

```
Video createdVideo = request.Upload(newVideo);
Status pop = new Status(createdVideo);
pop.Show();
```

newVideo is the parameter containing all data about the video file, including header, description, tags, key words and of course the video file itself. The request function

returns a reference to the video on YouTube, which gives access to changing the file and its details and video itself and also gives an update on the download, which the Status class takes from it.

4.1.3 GItem

As described before, a GItem is an Item object, containing all set data on an item from the Project Screen.

4.2 Item File

The item file is located in the project folder. This file is created by the Project Screen on addition of items to a project. It describes all data about an item:

- Item file path – where the file is located on the machine.
- Item type – varying from Image, Video, sound and effect.
- Screen position X/Y – a default appearance position, for when the item is activated by voice command.
- Menu position – where will the item appear on the item menu on record screen (set to -1 if not set).
- Voice command – a word or phrase, activating the item on record screen (blank if none).

4.3 User

The system saves the user's YouTube username and password after he enters them on the Project screen (if he hasn't, he will be requested to do so on the Uploading screen before being able to upload). The user can change those details on the Project screen if he wants to. On entering new details, the system will try to connect to YouTube and inform the user if the details are correct and if a connection was created successfully.

4.4 Testing

In order to complete this project, a few goal points were set. The main major goal was to complete the recording screen and completing a record session with a finished video up on YouTube.

4.4.1 Getting Video Feed From The Kinect– Learning The Kinect SDK

The Kinect is the heart of the system, giving the user all the power over it, so of course implementing the SDK was top priority. The first thing needed for using the

Kinect's features was getting the live feed from the camera and presenting the wanted RGB video feed on screen for the user to see himself.

4.4.2 Body Capturing

Learning from the Kinect's Developers' Toolkit and the many example codes in it lead to a first test run of the system. This version included two control nodes, attached to the hand joints of the skeleton created by the Kinect SDK. These controls supplied instant coordination of the user's hands for as long as he appeared in the Kinect's view.

4.4.3 Creating The Item Menu

A simple Kinect buttons layout was created and functions monitoring each button were created, giving the option of knowing if a button was activated by a hand. The time needed to hover over the buttons with a hand to activate them was set at this stage. The recording button was also implemented but was not yet active.

4.4.4 Creating The Item Class And The Controls

The information each item should keep was already decided at the design stage. A graphical object was created that can change content according to selected item and as long as no item was active, that object was not visible. This Item's state is decided by certain flags, determining if it Is active and dragged by a hand (and by which hand). The new system version allowed the user to select an item from the Item Menu, have it attached to a hand and drag it.

4.4.5 Top Bar Control

Some more control was needed for the user so the top bar was created. This bar is a button, allowing the user to deactivate a dragging or remove an item from screen and even resizing items.

4.4.6 Recording Session

The recording button is now activating a recording tool. A few different recording tools were tested and the one good enough for the system was selected, giving the option of setting the borders of the recording area, which was important for this system's requirements. Now pressing on the recording button starts a recording session and pressing on it again stops the recording and saves the video to the project's folder.

4.4.7 Uploading to YouTube

In order to upload a video to YouTube using a program, a developers key is needed and the YouTube API. After getting those, uploading the video was no problem but

an uploading screen was needed in order to give the user the ability to add data to the video, like a header and a description. A simple form was created in a separate window from the recording screen and that would appear on pressing the record button a second time (and by so ending the recording and having a video ready for upload).

4.4.8 Voice Commands

The voice commands were implemented last as an extra controller. Adding a certain word or phrase to the project's lexicon would allow the user to activate a user by saying that certain word or phrase. An item activated by a voice command would appear on screen in a specified coordination from the item data file added to that certain item class.

4.4.9 Not Implemented In The System

The Project Screen was not created and has only a rough design because it was not important for this educational project and not crucial for a proof of concept, so the items in the item data file can only be altered manually and the YouTube login details can only be changed through the code at this moment.

5   Installation and User Manual

The first thing to do is install the Kinect on Windows. A manual can be found here:

http://www.codeproject.com/Articles/148251/How-to-Successfully-Install-Kinect-on-Windows-Open

This page also has links to download all necessary drivers and files for the installation.

The system can only be started with the entire Visual Studio project files, using the SkeletalTracking application file located on SkeletalTracking\bin\Release, while the SkeletalTracking folder is the Visual Studio project folder.

When starting the application, the recording screen opens up, showing the live video feed from the Kinect camera, the item menu, top bar and record button.

Image 19 – Recording screen opens up on starting the system

The user should position himself in front of the Kinect, making sure it is located close to the computer screen and he can see himself clearly on it. The user now can start testing the controls, and he should do so before starting to record so he can have a good rehearsal.

In order to choose an item from the item menu, the user must hover one hand over it and wait for less than a second. A circle will appear around that item in the menu, filling itself up until it is chosen.

Image 20 - Choosing an item from the Item Menu

The item appears, attached to the selecting hand and dragged by it until it is released. The user can move around freely but must notice that the item is still attached to him.

Image 21 - Dragging an item on screen

In order to release the item to stay on screen, The user spreads his free hand upward to the top bar. Like the item button on the item menu, hovering one hand over the top bar for a little less than a second will activate it, releasing the item and

sticking it to its current place. A circle will fill up around the top bar as well, to note that a hand is hovering over it.

Image 22 - Touching the top bar with one hand while an item is attached to another

Image 23 - An item is placed on screen

In order to resize an item appearing on screen, the user stretches both arms upward, touching the top bar with both of them. While both hands are on the top bar, the user can move them farther from one another in order to enlarge the item, or move them closer together in order to shrink the item. The user can use a round movement of the hands in order to keep shrinking or enlarging the item.

Image 24 - Both hands touch top panel and are drawn closer to shrink the item appearing on screen
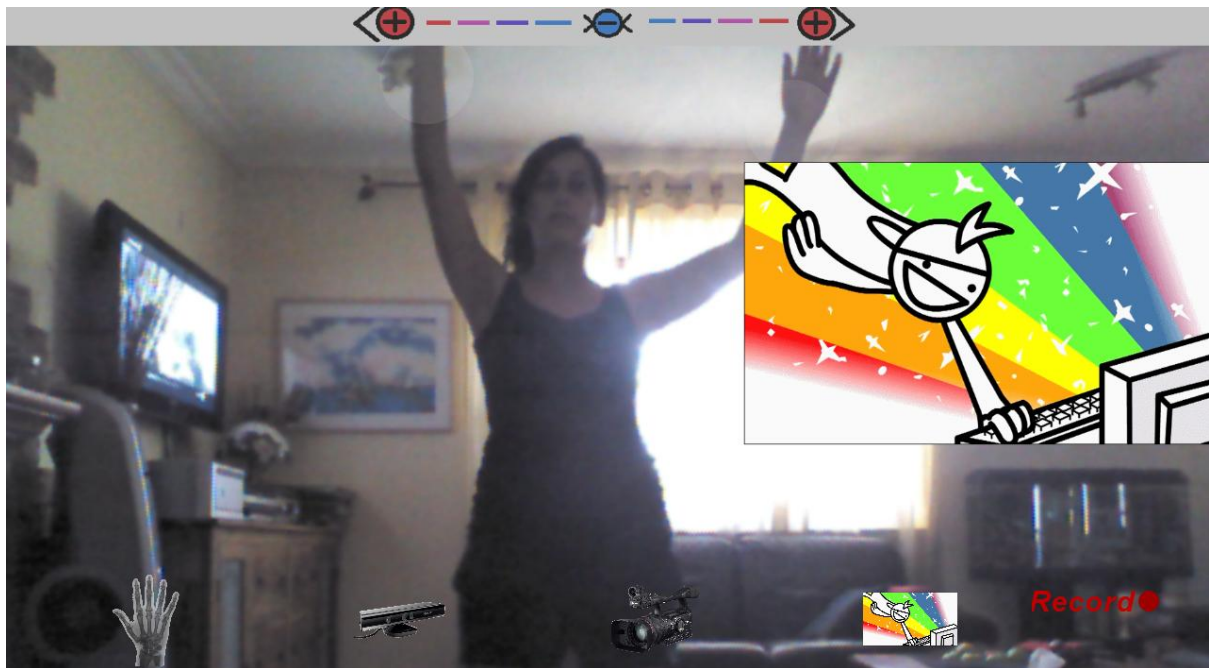

Image 25 - Both hands touch top panel and are drawn farther apart to enlarge the item appearing on screen

In order to make the item disappear, the user stretches one arm up to the top bar again for less than a second until it is activated and the item disappears.

In order to start recording the user hovers one hand over the "Recording" button in the item menu for a second. A circle will fill up around the button and it will activate, changing to a "Stop" button.

Now that the recording has started, the user can use all these tools in order to create a video of his choice, using all the items.

In order to stop the recording, the user presses the "Stop" button, by hover a hand over it for less than a second, activating it like other buttons. This stops recording and opens up the Uploading Screen.

On uploading screen, in order to upload the video to YouTube, the user enters all required details (Header, description, tags and key words) and presses on the "Upload" button. Now he just has to wait until the video appears on YouTube (depending on the length of the video. A one minute video will upload shortly while a ten minute video could take a few minutes to upload).

The current YouTube channel the system uploads to is:

http://www.youtube.com/channel/UC9fkuGr3j3YoFhPe986n24w?feature=watch

The other options on the Upload Screen are not active at the moment.

6   Bibliography

Wikipedia -
   http://en.wikipedia.org

An introduction to Kinect -
   http://www.microsoft-press.de/productinfo.asp?replace=false&cnt=productinfo&mode=2&type=2&id=mp-6396&index=2&nr=0&sid=6fd99a5542f17cb1ff480d3db9f6f45c

Bytescout Screen Capturing library -

   http://bytescout.com/products/enduser/screencapturing/screencapturing.html

YouTube Data API -

   https://developers.google.com/youtube/

Microsoft's Developer's Network -

   http://msdn.microsoft.com

תקציר מנהלים

Live Editing System היא תוכנית הנותנת למשתמש את היכולת ליצור פרויקטים, לצרף להם סוגים שונים של קבצים, כגון תמונות, סרטונים וקבצי אודיו, ולהשתמש בהם ליצירת סרטו וידאו מקורי, המצולם באמצעות מצלמת הקינקט. הקינקט נותן למשתמש שליטה על אלמנטים שונים במסך באמצעות תנועות גוף ופקודות קוליות. המערכת כוללת, מלבד עריכה והקלטה, אפשרות להעלות ל-YouTube את הסרטונים המצולמים באמצעותה, מיד לאחר סיום הכנתם. כל התכונות הללו, במערכת אחת, מספקים אפליקציה בודדת הכוללת את כל הכלים הנחוצים לצילום, עריכה ושיתוף של סרטונים מקוריים עם אפקטים מיוחדים וכך חוסכת את הזמן והעבודה הדרושים בדרך כלל להכנת סרטונים כאלה.

המטרה של הפרויקט היא לבנות מערכת פשוטה הכוללת את הכלים הבסיסיים הדרושים לקיום מערכת כפי שתוארה. החלקים החשובים ביותר במערכת הם מערכת השליטה באלמנטים הגרפיים על המסך באמצעות זיהוי תנועות גוף ופקודות קוליות מהמשתמש בעזרת הקינקט, הקלטת סרטונים

באמצעות המערכת והעלאת הסרטונים לחשבון מוגדר ב-YouTube. חלק ניהול הפרויקטים והוספת קבצים אליהם הוא פחות קריטי לצורך הפרויקט משום שאין לו תרומה מדעית.

שנקר – בי"ס גבוה להנדסה ולעיצוב

הפקולטה להנדסה

<u>המחלקה להנדסת תוכנה</u>

Live Editing System

פרויקט גמר

מאת
נופר שקד

מוגש כחלק מהדרישות לקבלת תואר ראשון בוגר במדעים (.B.Sc).

2013.09.26