

**INTERACTIVE LEARNING APPLICATION FOR COMPUTER  
PROGRAMMING**

By

CHONG NUI MEI

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEM ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2018

UNIVERSITI TUNKU ABDUL RAHMAN

**REPORT STATUS DECLARATION FORM**

**Title:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Academic Session:** \_\_\_\_\_

I \_\_\_\_\_

**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

\_\_\_\_\_  
(Author's signature)

\_\_\_\_\_  
(Supervisor's signature)

**Address:**  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
Supervisor's name

**Date:** \_\_\_\_\_

**Date:** \_\_\_\_\_

**INTERACTIVE LEARNING APPLICATION FOR COMPUTER  
PROGRAMMING**

By

CHONG NUI MEI

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEM ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2018

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**Interactive Learning Application for Computer Programming**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

ID No : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENT**

It is a genuine pleasure to express my deep sense of gratitude to my supervisors, Mr. Thomas Ooi Joo On for providing me an opportunity to engage in this project. His valuable guidance, encouragement, and scholarly advice has been solely and mainly responsible to guide me largely to accomplish this project.

Finally, it is my privilege to thank my parents for their constant encouragement, financial support and co-operation, which were the sustaining factors in carrying out the work successfully throughout the course.

## ABSTRACT

Computer programming, also known as coding, is the process that leads from an original formulation of a computing problem to executable computer programs, in short, creating software. Programming is a writing instruction for a computer to perform, which involves activities such as planning, testing, and debugging source code in target programming language. Source code is written in one or more programming languages. The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solving a given problem. However, the process of learning programming through books and online tutorial is less of interactivity.

This project titled as “**Interactive Learning Application for Computer Programming**” focuses on introducing the concept and knowledge of computer programming. The purpose of this project is to design a multimedia courseware to provide an interactive learning pathway to underage children (primary school kids). Multimedia courseware is including the multimedia elements inside the courseware to make the system become vivid and interactive. This courseware will include graphic, audio, animation and others multimedia elements.

Using multimedia to do a courseware is wish to increase concentration and focusing of target user during the process of learning. On the other hand, multimedia courseware provide interactivity between users, which reduce the difficulties of learning process.

As a consequent, the study has been concluded that the courseware is effectively in providing a new pathway of programming learning process for primary school kids.

## TABLE OF CONTENTS

REPORT STATUS DECLARATION FORM .....	i
INTERACTIVE LEARNING APPLICATION FOR COMPUTER PROGRAMMING .....	ii
DECLARATION OF ORIGINALITY .....	iii
ACKNOWLEDGEMENT .....	iv
ABSTRACT .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES .....	viii
LIST OF TABLES .....	x
CHAPTER 1 INTRODUCTION .....	1
1.1 Motivation .....	1
1.2 Problem Statement .....	2
1.3 Project Scope .....	3
1.3.1 Modules .....	3
1.4 Project Objectives .....	5
1.5 Impact, significance and contribution .....	5
1.6 Background Information .....	6
CHAPTER 2 LITERATURE REVIEW .....	8
2.1 Overview of Multimedia .....	8
2.1.1 Advantages of Multimedia .....	8
2.2 Definition of Multimedia Courseware .....	9
2.2.1 Advantages of Multimedia Courseware .....	10
2.3 Literature Review .....	11
2.3.1 Lightbot .....	11
2.3.2 CodeMonkey .....	12
2.3.3 Kodable .....	14
2.4 Critical Remarks of previous works .....	16
2.4.1 Strength and weaknesses .....	16
2.4.2 Comparison between Benchmark Review .....	18
2.5 Expected Innovation .....	18
CHAPTER 3 SYSTEM DESIGN .....	19
3.1 Use Case Diagram .....	19
3.2 System Design / Overview .....	20
3.2.1 Storyboard .....	20

3.3 Component of Multimedia .....	27
3.3.1 Images .....	27
3.3.2 Videos.....	31
3.3.3 Audios .....	32
CHAPTER 4 METHODOLOGY .....	34
4.1 Design Specifications .....	34
4.1.1 Development Methodology.....	34
4.2 Development Tools.....	36
4.2.1 Macromedia Director MX 2004 .....	36
4.2.2 MeituXiuXiu.....	36
4.2.3 Snipping tool .....	36
4.2.4 Video DownloadHelper Extension in Firefox .....	36
4.2.5 SoundOfText Website .....	36
4.2.6 YouTube to MP3 Website .....	36
4.2.7 Online MP3 Cutter Website .....	37
4.2.8 Scratch 2.0 Offline Editor .....	37
4.2.9 SWF to EXE Converter .....	37
4.3 Specification of Requirements.....	37
4.3.1 User Requirement.....	37
4.3.3 Non-Functional Requirement .....	38
4.4 Project Timeline .....	40
CHAPTER 5 SYSTEM IMPLEMENTATION .....	44
5.1 Screenshot .....	44
5.2 Issues and Challenges in Implementation .....	50
CHAPTER 6 TESTING.....	51
6.1 Alpha Testing.....	51
6.1.1 Unit Testing.....	51
6.1.2 Integration Testing.....	57
CHAPTER 7 CONCLUSION.....	58
7.1 Conclusion.....	58
7.2 Problem Faced .....	58
7.3 Future Work.....	59
BIBLIOGRAPHY .....	60



## LIST OF FIGURES

Figure 2-3-1-F1: Interface of Lightbot stage (a) .....	11
Figure 2-3-1-F2: Interface of Lightbot stage (b) .....	11
Figure 2-3-1-F3: Interface of Lightbot Menu .....	11
Figure 2-3-1-F4: Interface of Lightbot Guide .....	11
Figure 2-3-2-F1: Interface of CodeMonkey (a) .....	12
Figure 2-3-2-F2: Interface of CodeMonkey (b) .....	12
Figure 2-3-3-F1: Interface of Kodable .....	14
Figure 2-3-3-F3: Interface of Kodable planet.....	14
Figure 2-3-3-F4: Interface of Kodable stages.....	14
Figure 3-1-F1 Use Case Diagram .....	19
Figure 3-2-1-F1: Storyboard of Main Menu.....	20
Figure 3-2-1-F2: Storyboard of Tutorial, Quiz and Practical Stages Menu .....	21
Figure 3-2-1-F3: Storyboard of Game Page .....	22
Figure 3-2-1-F4: Storyboard of Lesson Page .....	23
Figure 3-2-1-F5: Storyboard of Lesson.....	24
Figure 3-2-1-F6: Storyboard of About Page .....	25
Figure 3-2-1-F7: Storyboard of Video Page.....	26
Figure 3-3-1-F1: Search image online .....	27
Figure 3-3-1-F2: Choose cropping tool in MeituXiuXiu.....	28
Figure 3-3-1-F3: Choose Auto Cropping.....	28
Figure 3-3-1-F4: Crop Images and Save in transparent background .....	29
Figure 3-3-1-F5: Rename image and save.....	29
Figure 3-3-1-F6: Select word tool in MeituXiuXiu.....	30
Figure 3-3-1-F7: Insert word and create button .....	30
Figure 3-3-2-F1: Download video.....	31
Figure 3-3-3-F1: Download audio .....	32
Figure 3-3-3-F1: YouTube to MP3 website .....	32
Figure 3-3-3-F1: Online MP3 cutter .....	33
Figure 4-1-1-F1: Process of Prototyping-Based Methodology.....	34
Figure 4-4-F1: Fyp 1 Gantt Chart.....	40
Figure 4-4-F2: Fyp 1 Gantt Chart.....	41
Figure 4-4-F3: Fyp 2 Gantt Chart.....	42
Figure 4-4-F4: Fyp 2 Gantt Chart.....	43
Figure 5-1-F1: Main Menu .....	44
Figure 5-1-F2: Quiz Module Mouse Over.....	44
Figure 5-1-F3: Tutorial Module Mouse Over .....	44
Figure 5-1-F4: Practical Module Mouse Over .....	45
Figure 5-1-F5: Lesson Module Mouse Over.....	45
Figure 5-1-F6: About Module Mouse Over .....	45
Figure 5-1-F7: Exit Mouse Over .....	45
Figure 5-1-F8: Video Module Mouse Over .....	45

Figure 5-1-F9: Game Stage.....	45
Figure 5-1-F10: Game Stage Full Screen .....	46
Figure 5-1-F11: Sub-Menu .....	47
Figure 5-1-F12: Sub-Menu Mouse Over .....	47
Figure 5-1-F13: Lesson Module.....	48
Figure 5-1-F14: Video Module .....	49
Figure 5-1-F15: About Module.....	50
Figure 5-1-F16: Exit Page .....	50

## LIST OF TABLES

Table 2-4-1-T1: Strength and weaknesses of Lightbot .....	16
Table 2-4-1-T2: Strength and weaknesses of CodeMonkey .....	16
Table 2-4-1-T3: Strength and weaknesses of Kodable.....	17
Table 2-4-2-T1: Comparison between Benchmark Review .....	18
Table 4-3-4-1-T1: Table of Software Requirements .....	39
Table 4-3-4-2-T1: Table of Hardware Requirement.....	39
Table 6-1-1-T1: Table of Main Menu Page Testing.....	51
Table 6-1-1-T2: Table of Tutorial Page Testing .....	52
Table 6-1-1-T3: Table of Quiz Page Testing .....	53
Table 6-1-1-T4: Table of Practical Page Testing.....	53
Table 6-1-1-T5: Table of Lesson Page Testing.....	53
Table 6-1-1-T6: Table of Video Page Testing .....	54
Table 6-1-1-T7: Table of About Page Testing.....	55
Table 6-1-1-T8: Table of Exit Page Testing.....	55
Table 6-1-1-T9: Table of Sub Topic Page Testing .....	55
Table 6-1-1-T10: Table of Scratch Stage Testing.....	56
Table 6-1-2-1-T1: Table of Image and Sound Testing .....	57
Table 6-1-2-1-T1: Table of SWF Testing .....	57

## CHAPTER 1 INTRODUCTION

### 1.1 Motivation

“In the 20<sup>th</sup> Century, meaningful education was all about learning your ABCs. Today, it’s centred on Alphas, Betas and C++”, Orsini’s words (2013) had showed how important about owing a programming knowledge is in this day. Computing involved in nearly all parts of our lives today. “Programming will be one of the fastest growing professions, and more programmer is need over the next 10 years” (Kodable, 2016). Missio (2015) spoke that, “Four- and five-year-olds can learn the foundations of coding and computer commands before they can even write and spell words.”, and in fact, kids may start to train their logic thinking skills and foster problem-solving techniques, through some relaxing way, before they can really start study in school.

Undeniable, children nowadays are living in the surrounding of digital, they may intelligence enough to enjoy on using the technologies, but they are not very interested in the logic behind, as adult will do. However, they should learn to understand the logic behind those technologies too. Our dependence on technology will never decrease. As programming had become a basic literacy in the digital age, children today must be able to grow into an active part of this giant digital shift (Tynker.com, 2016). In Kodable’s analysis (2016), diverse early learning will definitely benefit kids. Bill Gates stated, “Learning to write programs stretches mind, and helps you think better, produces a way to think about things that is helpful in all domains.” Start to learn programming since childhood may help children to become familiar with technology, as going from merely interact with technology, to create technology and convey oneself in new ways.

It seems like stressful for the kid to learn coding in proper way. However, an interactive multimedia is a good start for kids to contract with computer programming. In addition, they can learn without having stress.

## 1.2 Problem Statement

- i. Inefficiency of motivation in learning computer programming logic.

In traditional way of study computer programming, long paragraphs of concepts and theories, and how to apply the coding is very important in order to come out with a good application. However, not to mention a child, an adult might also feel difficult to understand a project coding or develop a project application. Sheets of words and programming code will always stress up a person in the process of learning, and many of them will ends up with give up and lost interest in learning computer programming. If anyone lost interest on learning anything, they cannot proceed; also, they cannot get good result on it.

- ii. Inefficiency of suitable method to learn computer programming logic.

When someone is eager to start on learning computer programming, they might prefer to buy a proper textbook; they might surf to any website, which provide programming tutorial; or even they might register to a tuition class for programming practices. Those are all correct ways, but they might not the suitable path for a young beginner, especially the underage children. If they proceed to proper learning on programming concepts and theories, or even coding directly, they might not able to catch up with the syllabus or really understanding about the concepts on it.

- iii. False impression of learning programming is difficult.

First say of learning programming, many people will have a false impression on it is a hard job. Because everyone knows programming is not just about learning, but the more important is how to apply it, and how to code it. Indeed, write a good program is bringing ourselves everlasting difficult life if we do not have good logic thinking skills and foster problem-solving techniques. Many people try to stop themselves from going to the difficult life just because they think they cannot do it, or do it well. That is why they face a barrier of grasping the knowledge and skills.

### **1.3 Project Scope**

This project aims to develop a prototype of interactive learning application for underage child (7 – 9 years old kids) to have some basic understanding on programming concepts. The main objective of this project is to improve problem solving and logic thinking skills of kids, which are both transferable and empowering, and may help a lot in their future when they wish to continue study on computer programming areas (Crawley, 2014).

#### **1.3.1 Modules**

##### **1.3.1.1 Tutorial Module**

Tutorial module allows target users to try on some simple stages on different topics designed. Each stage designed based on a particular topic, with guidance and hints to attract kids' attention to interact with it. Tutorial module as a game-like based module will contain multimedia elements include text, sound, graphics, and animations. Tutorial module should be attractive enough as it is important on a role of drawing interest of target users.

##### **1.3.1.2 Quiz Module**

Quiz module is the stage that is similar to tutorial module, with some additional challenges and no more hints will be given. Similar to Tutorial module, Quiz module is designed based on particular topic and to emphasize kids' understanding after Tutorial module. Quiz module will contain multimedia elements include text, sound, graphics, and animations. There will be marks rewarded based on users' performance in Quiz module, and Quiz module should have an important role on letting users to learn while they are playing.

### **1.3.1.3 Practical Module**

Practical module is the design of combination of topics to be a challenge game-like module. Design of Practical module will be similar like Tutorial and Quiz modules, which contain multimedia elements of text, graphics, sound, and animations. However, the main difference is Practical module targeted on application of techniques learnt from different topics of Tutorial and Quiz. Users are always encouraged to finish tutorial and quiz module first before they try on Practical module.

### **1.3.1.4 Lesson Module**

Lesson module is used to show the course materials about the Computer Programming. The course materials is prepared in simple C++ programming language to let the target users to further understand on what computer programming is, and act for realization of what they are actually learning while they are interacting in quiz, tutorial and practical module. Lesson module will contain text and graphics multimedia elements for kids to easily understand and learn the computer programming.

### **1.3.1.5 Video Module**

Video module contains video and audio multimedia elements, which include some interesting animated videos of explaining what computer programming is. Those animated videos can used to draw attentions of kids, or to emphasizing their realization on what they are learning about (what computer programming is).

### **1.3.1.6 About Module**

About module includes steps on trying out the Tutorial, Quiz, and Practical Module, and the guide on the rules of the game. In addition, About module will cover some appendixes, such as special word terms, keywords in computer programming etc.

### **1.4 Project Objectives**

- To invoke desire or interest of kids (age of 7 to 9) to learn programming skills, which will help a lot for their study in coding.
- To construct attractive method to motivate kids to pay attention on learning programming skills.
- To present a pleasurable method of learning programming skills to correct false impression of it is a hard job.

### **1.5 Impact, significance and contribution**

Nowadays, knowledge of programming and coding are important to the society and to an individual, even to an underage child (age of 7 to 9). Technologies are embedded in humans' daily lives, such as cell phones, computers, social media and games. Toys are digital and even a lot of them are programmable. Our dependence on technology will never decrease and it is the great opportunities for children to start to understand the logic behind the technologies and control them (Tynker.com, 2016). Thus, for a child to start on learn programming, an interactive multimedia will be a good start for them to understand about the programming logic and skills.

A multimedia courseware with some game-like stages is a great tool to draw attentions of kids, instead of asking them to “learn”, we start with “playing”. After they are willing to touch with it, different topics of game-like stages with challenges and rewards will assist in invoking their interesting into it. Step by step guidance here should be a good motivation for them to go on. Therefore, when they are rather “expert” in the gaming stage, it is the time to let them realize on what are they actually learnt.



Once the concepts of dealing with computer programming had planted in their mind, it will be a good starting line for them to learn programming languages.

The interactive application let the process of learning programming become fun and not boring. Unlike a programming tutorial website, the application contributes the pleasurable and interesting method in learning programming skills, and this is important to correct the false impression of learning programming is tough. In expect, the number of people learning programming will be increase in the future.

## **1.6 Background Information**

Learning computer programming aids students to improve critical thinking skills and problem solving skills and prepares them for a successful future. By learning computer programming, students are empowered to create their own projects and in the process, they are trained to be independent and firm.

Computer programming is a set of instructions that a computer will do something accordingly. Apps, digital toys, movie-making tools, and drawing tools are example of computer programs. Writing the systematic instructions that a computer follows is called coding. The code and written in clear and correct sequence to tell computer how exactly to complete a task. The programming logic could be imagine like writing instructions on how to make a peanut butter and jelly sandwich. Every steps should be covered, including opening the jar of peanut butter and using a knife to scoop it out and spread it on a slice of bread.

Different instructions will be involve in the coding process. Command is an instruction a computer can follow. For example, command a character to move one-step forward. A loop is a set of repeating command. Therefore, if you want a character to move one-step forward, then another step forward, and then another step forward, you might use a loop to show that a command is repeated three times. Mistakes are always exist in the process of coding. The process of finding and fixing errors are debugging. Patient and determination are always required to overcome the challenge of finding mistakes, addressing, and improving the code.

Algorithm is a set of instructions that is used to complete a task. Algorithms are made up of commands, loops, and other instructions. A computer program might be made up of different algorithms. It may be helpful to think of an algorithm as a recipe for a cake, anyone can follow the recipe to make the same cake.

Knowing how to use technology is essential, but in today's digital world, understanding how technology works is thrilling and also an important skill, particularly as our dependence on technology continues to grow.

## CHAPTER 2 LITERATURE REVIEW

### 2.1 Overview of Multimedia

As said by Hofstetter (2001), “Multimedia is the use of a computer to present and combine text, graphics, audio, video, and animations with links and tools that let the user to navigate, create, interact, and communicate.” In today’s information age, we include the term “multimedia” if we want to construct high quality environment of learning. The essentials of multiple media, user control over the delivery of information, and interactivity are advancing the process of learning through integrated learning environments.

#### 2.1.1 Advantages of Multimedia

Multimedia bring many benefits. The use of multimedia can critically affect attractive to the learners. Some of the various benefits are listed as below:

- i. Better understanding.

Multimedia able to make connections between verbal and visual representations of content by taking advantage of the brain’s ability, and hence lead to a deeper understanding for learning. It aids learners to picture out what they are learning and aids in constructing an association between the learner and the information. Wajid (2016) provided an example on this, our brain attempts to picture it by generating an image when we come across something strange. However, a ready-made picture come along with text definitely will aid learners to jump right to the point, helping learners to get familiar with the new concept faster.

- ii. Improve problem solving.

More than half of the brain undergoing visual processing by itself. Graphics, video and animations along with text inspires brain to rise human's attention and retention. Hence, under multimedia learning environment, it is easier for people to identify and solve problems than traditional teaching by textbook.

- iii. Increased positive emotions.

Psychologist Barbara Fredrickson indicated that, live through positive emotions aspire more potentials of people in their lives. Using multimedia while teaching effects learner's mood, in which they can tend to be more active and have better learning with a positive attitude. (Chioran, 2016)

## **2.2 Definition of Multimedia Courseware**

Courseware proposed as kits for educators or trainers, as well as lessons for students. It is normally packaged for use with a computer (Jing, 2005). Courseware cover many knowledge areas, which included sport, science, and languages, but the most usual is information technology. Most people understood multimedia courseware as communication mediums within a computer program to present information. As stated by Hick (1997), "communications mediums mean audio for music, sound effects, or voice-over narration, still photographs and / or graphics to aid the user to understand message, video for further explanation or ideas illustration."

### 2.2.1 Advantages of Multimedia Courseware

There are a list of benefits of interactive multimedia courseware, below are few of worth being nominated:

i. Time saving.

Content of course take hours to study, also, our surrounding is full of different interruptions. Time of learning are saved when those course contents explained in direct ways with the aid of videos or audio narrations in shorter time. Moreover, program is useful when situations arise; as some researches have shown that when people facing similar topic at the moment, they learn better,.

ii. Efficiency.

One of the most effective learning methods is learning by doing. Videos and animations aid in great explanation more efficiently than separate text for complex concepts. Video simulations and simple animations also help learners to develop practical skills and increase information retention.

iii. Flexible.

Multimedia courseware gives more flexibility compare to traditional teaching style. Learners can choose their own time and place to study in which they only need to have a computer.

iv. Interactive.

Abundant of studies found that strong optimistic result on learning comes from interactivity. For instance, Bosco (1986) reviewed 75 learning studies and found that, with interactive multimedia, learners have greater attitudes toward learning, and can learn in shorter time.

## 2.3 Literature Review

### 2.3.1 Lightbot

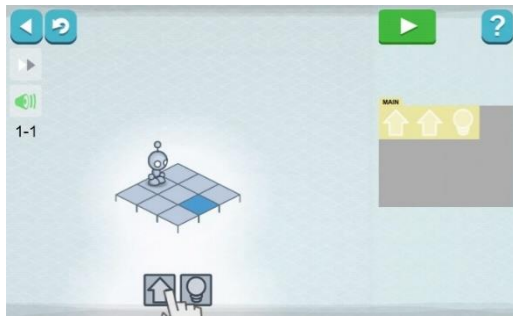


Figure 2-3-1-F1: Interface of Lightbot stage (a)

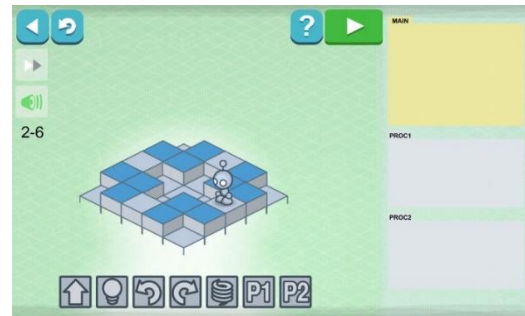


Figure 2-3-1-F2: Interface of Lightbot stage (b)



Figure 2-3-1-F3: Interface of Lightbot Menu



Figure 2-3-1-F4: Interface of Lightbot Guide

Lightbot has been there for few years, it started as an online Flash game in 2008, and newer version rebuilt as an application for Android and IOS mobile phones (Biggs, 2013). Lightbot is supported for several languages such as English, French, Traditional and Simplified Chinese and others.

Lightbot is a programming puzzles game, where players are required to use programming logic to solve levels and earn challenge stars. Based on review from deBoucher (2016), Lightbot offers 50 levels and 20 challenges stars to gather, where stars as a reward is an essential element for any age of puzzle enthusiasts.

Rules of Lightbot is to simply command a robot to navigate a maze, light up tiles, and solve levels. Players arrange buttons that are available on screen to command the robot to pace, turn (clockwise or anticlockwise), jump, and light up a tile and so on. The complication of the pattern of maze and the commands will be increasing as the



do in a text-based editor. The game-like quality makes them willing to take risks and persevere.” (Playcodemonkey.com, 2016).

In CodeMonkey, kids can learn the basics of programming in a JavaScript-like language which cover the key code concepts such as variable substitution, function definition, and basic flow control. Players gained experience and learned from every previous puzzle he or she solved, will apply the knowledge and solve the following one through a well-planned coding curriculum. The puzzle always keep attention on the solution to the problem rather than making themselves running into trouble in later levels, which could require the help of teachers or experts with programming knowledge to help them out. CodeMonkey provides a very easily reached set-up for potential programmers to learn complex technical concepts.

In conclusion, CodeMonkey always try their best to attract kids to learn programming with puzzles presented in a friendly style, and promotes idea that programming is easy and fun (Thomas, 2016).



### 2.3.3 Kodable

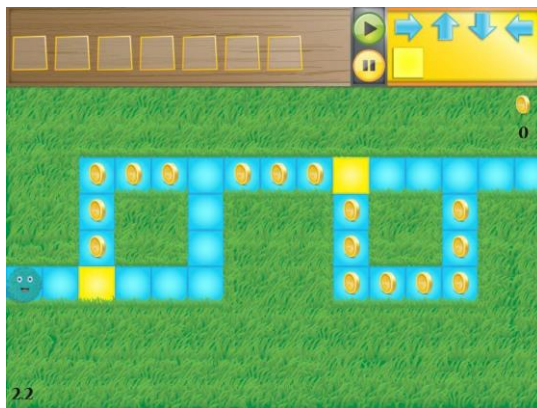


Figure 2-3-3-F1: Interface of Kodable

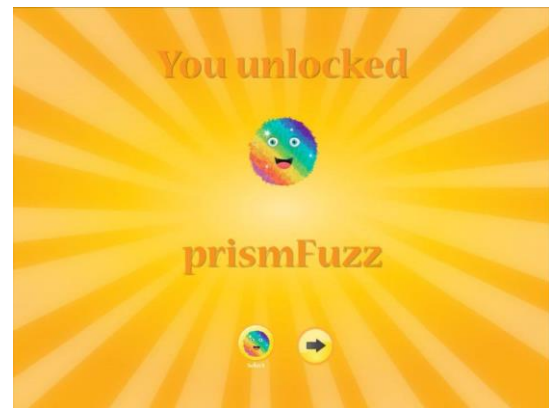


Figure 2-3-3-F2: Interface of Kodable reward



Figure 2-3-3-F3: Interface of Kodable planet

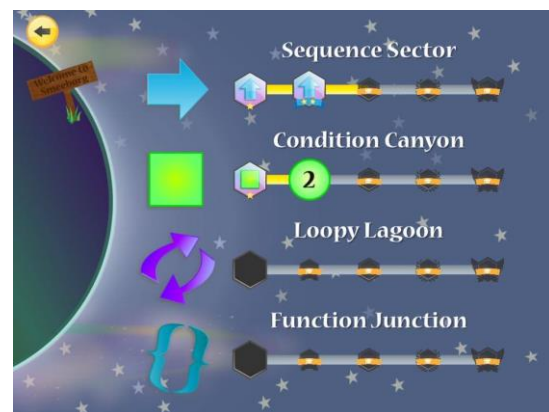


Figure 2-3-3-F4: Interface of Kodable stages

Kodable is a programming game which teaches pre-literate kids the basic concepts of programming. It was composed by Grechen Huebner and Jon Mattingly, and was first released into the AppStore in 2012 (En.wikipedia.org, 2016b).

The basic game mechanic is programming a fuzzy ball-like character by using a row of commands (arrows, paths, and boxes) to roll through mazes, collect coins and get the stars to complete maze levels, and unlock new coloured Fuzzes to program. Kodable features over 160 levels, 3 planets, 7 kind of ships and 16 characters to unlock.

Based on Gudmundsen (2014), kids may quickly catch up with the programming logic through playing this game, by dragging arrows to prepared spaces in the correct order, they command the Fuzzes to move through the mazes. Those arrows is the

direction the Fuzz should move in order to get through the maze. To earn the most stars at the end of the maze, player need to program the Fuzz to roll over the coins which spread throughout the maze. More, Kodable also introduce concepts of conditionals and loops in this puzzle game.

While the Fuzzes are rolling, it is together with upbeat music. The puzzle game will be gradually harder. Its progression is just right. When player finish one stage after few levels, they can unlock new Fuzzes, this motivate them to keep on playing.

Kodable is different from other programming application. Kodable provides great wholly voiced interactive tutorials. In addition, learning guide are provided to clarify concepts to guardians and teachers and offer solutions in all puzzles.

## 2.4 Critical Remarks of previous works

### 2.4.1 Strength and weaknesses

**Table 2-4-1-T1: Strength and weaknesses of Lightbot**

<b>Strength</b>	<b>Weaknesses</b>
Teaching kids programming concept	No user account, only one player at a time
Kids can get as many chances as they need to be successful.	Level of challenge ramp out very quickly, easy to get stuck for a while on some levels
No coding from any programming language will be use, look funny and easy.	No hints or clues are offered
No penalty for trying and failing. They may try, fail and try again if they want to find the best solution	

**Table 2-4-1-T2: Strength and weaknesses of CodeMonkey**

<b>Strength</b>	<b>Weaknesses</b>
Kids can get as many chances as they need to be successful.	Learning with real programming language, kids might lost interest on it. Suitable for kid above 8 or 9.
No penalty for trying and failing. They may try, fail and try again if they want to find the best solution	Level of challenge ramp out very quickly, easy to get stuck for a while on some levels
	No hints or clues are offered

**Table 2-4-1-T3: Strength and weaknesses of Kodable**

<b>Strength</b>	<b>Weaknesses</b>
Teaching kids programming concept, most likely engage kids who would be drawn to programming anyway	No hints or clues are offered
Kids can get as many chances as they need to be successful.	Only one language which is English is available
No coding from any programming language will be use, look funny and easy.	
No penalty for trying and failing. They may try, fail and try again if they want to find the best solution	
More than one user account can be created	

### 2.4.2 Comparison between Benchmark Review

Table 2-4-2-T1: Comparison between Benchmark Review

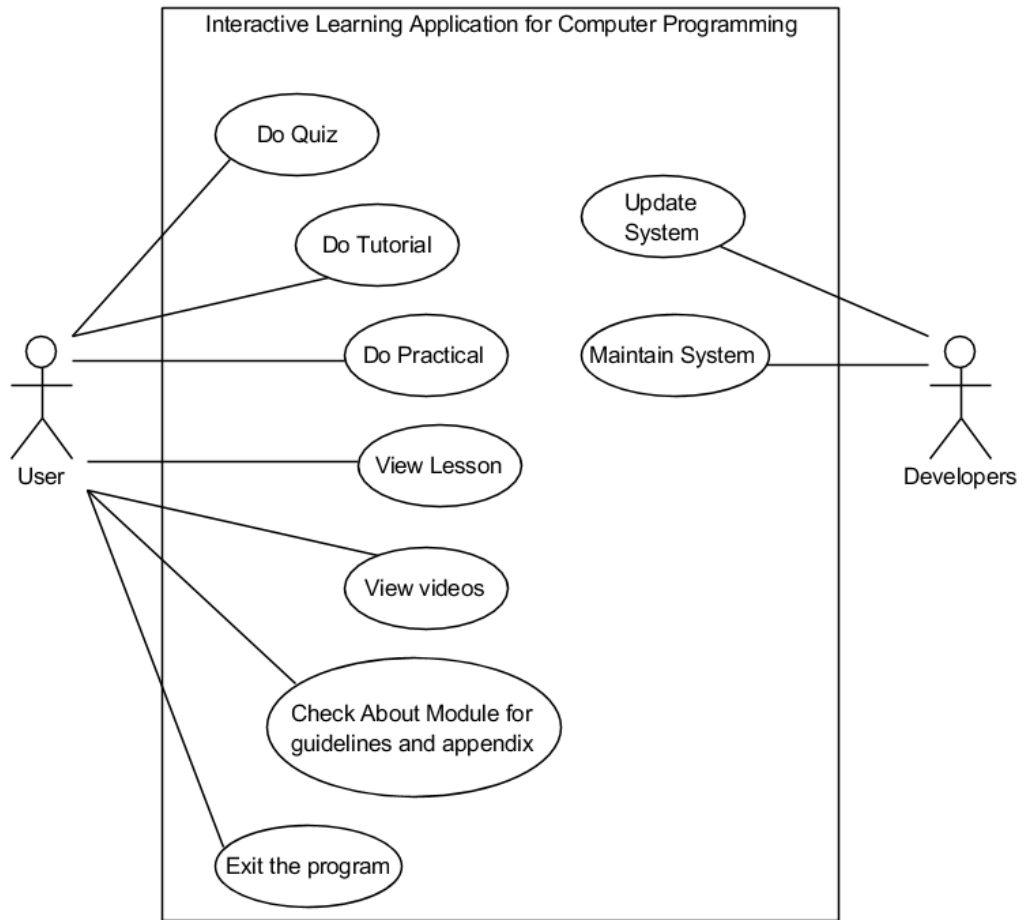
<b>System</b> <b>Lesson</b>	<b>LightBot</b>	<b>CodeMonkey</b>	<b>Kodable</b>	<b>Interactive learning application</b>
<b>Simple workflow</b>	✓	✓	✓	✓
<b>Procedure</b>	✓	✓	<b>X</b>	✓
<b>Looping</b>	✓	✓	✓	✓
<b>Variable Substitution</b>	<b>X</b>	✓	<b>X</b>	✓
<b>Conditional</b>	✓	<b>X</b>	✓	✓
<b>Array-pointer</b>	<b>X</b>	<b>X</b>	<b>X</b>	✓

### 2.5 Expected Innovation

Interactive Learning Application will offers a great interactivity and eye-catching interface for the users. Hence, the Interactive Learning Application will provide many practical exercises in several modules (tutorial, quiz, practical) for users to increase the interactivity and hands-on practical. In additional, multimedia courseware studied did not offer lesson module for the users to develop their realization on what they have learned. This interactive learning application for computer programming is not targeted to be a game but a tool to attract, teach and give lesson as what users had already learned in “game”.

### CHAPTER 3 SYSTEM DESIGN

#### 3.1 Use Case Diagram



**Figure 3-1-F1 Use Case Diagram**

## 3.2 System Design / Overview

### 3.2.1 Storyboard

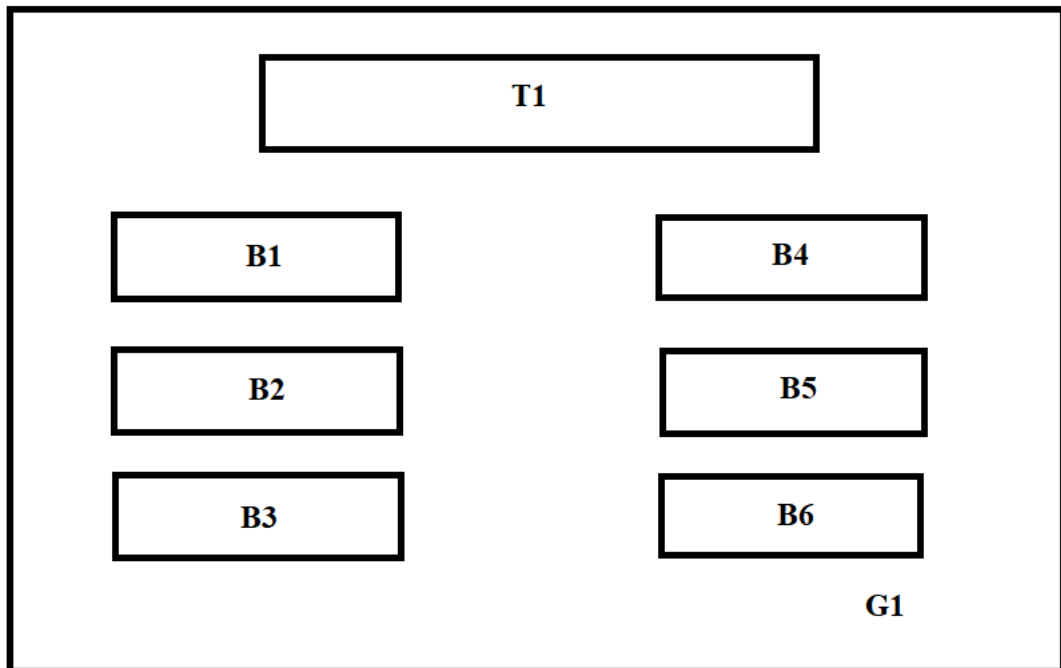


Figure 3-2-1-F1: Storyboard of Main Menu

**T1:** Text element showing “Interactive Learning Application for Computer Programming”.

**B1:** Graphic element showing “Lesson” and will link to Lesson Page.

**B2:** Graphic element showing “Quiz” and will link to Quiz Page.

**B3:** Graphic element showing “Practical” and will link to Practical Page.

**B4:** Graphic element showing “Tutorial” and will link to Tutorial Page.

**B5:** Graphic element showing “About” and will link to About Page.

**B6:** Graphic element showing “Video” and will link to Video Page.

**G1:** Graphic element showing background image.

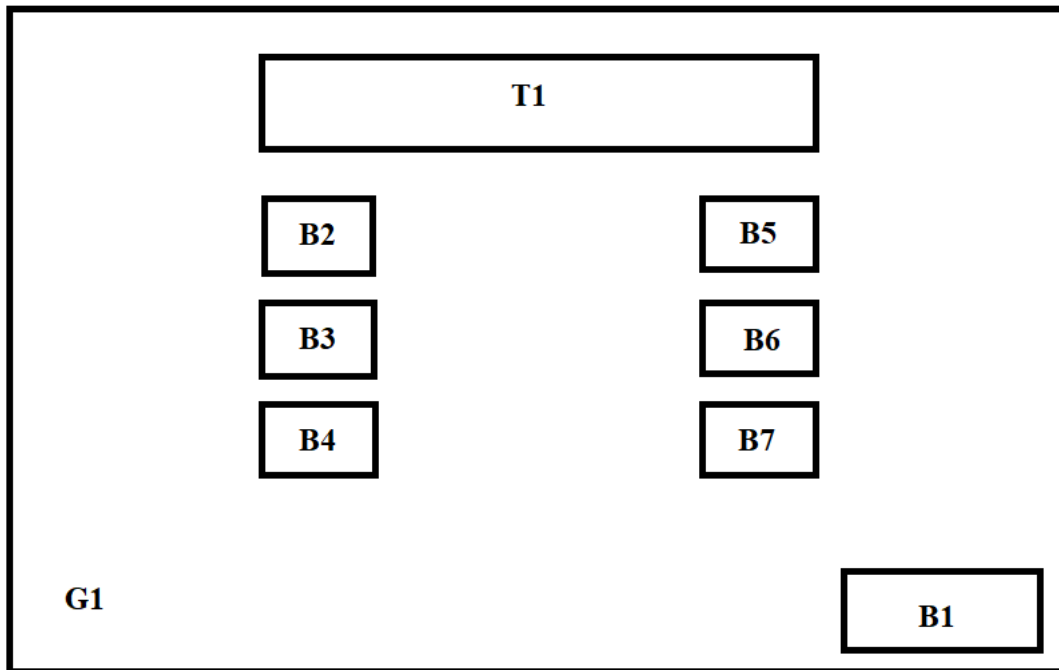


Figure 3-2-1-F2: Storyboard of Tutorial, Quiz and Practical Stages Menu

**T1:** This is a text element, which shows “Tutorial” or “Quiz” or “Practical”.

**G1:** Graphic element showing background image.

**B1:** Graphic element showing “Back” and will link to Main Menu.

**B2-B7:** Graphic elements showing name of each stages and will link to those game stage.



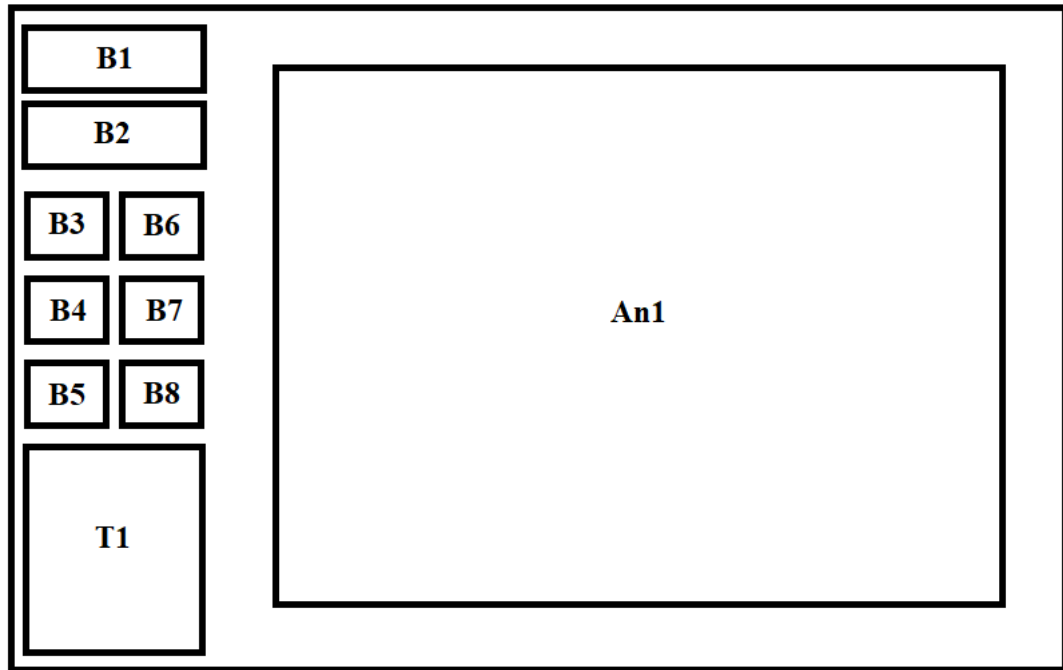


Figure 3-2-1-F3: Storyboard of Game Page

**An1:** Animation element showing the platform of the game.

**B1:** Button element showing “Start” and start the actions.

**B2:** Button element showing “Clear” and clear the actions.

**B3-B8:** Actions available to do, included “up”, “down”, “left”, “right”, “take”, and “procedure”.

**T1:** Text element showing list of actions selected.

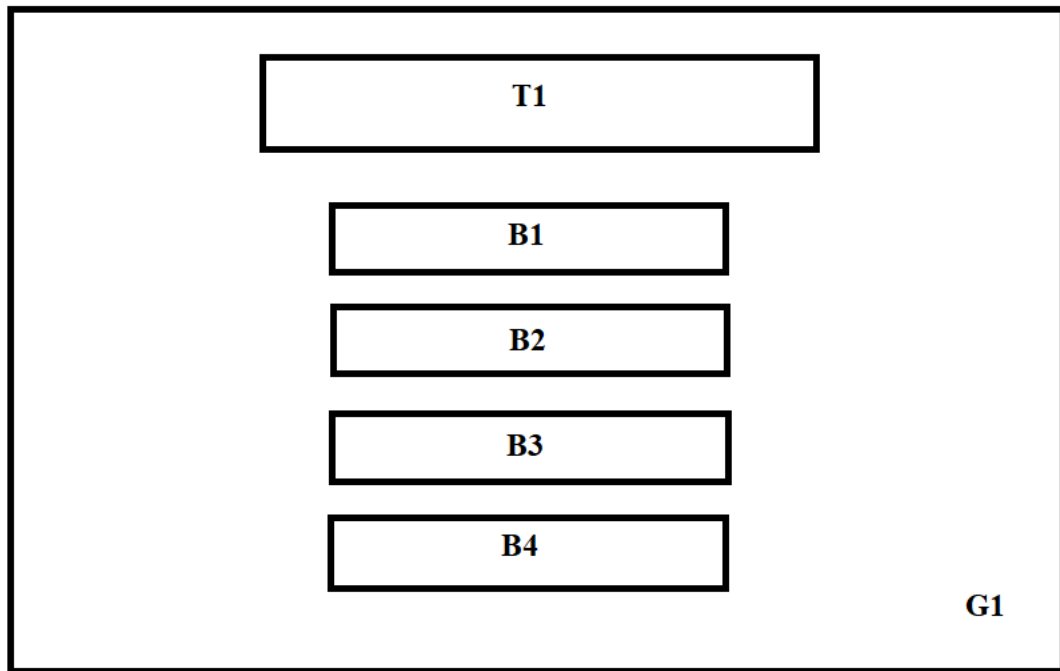


Figure 3-2-1-F4: Storyboard of Lesson Page

**T1:** Text element showing “Lesson”.

**B1 – B4:** Graphic elements showing name of certain topics and link to those topics.

**G1:** Graphic element showing background image.

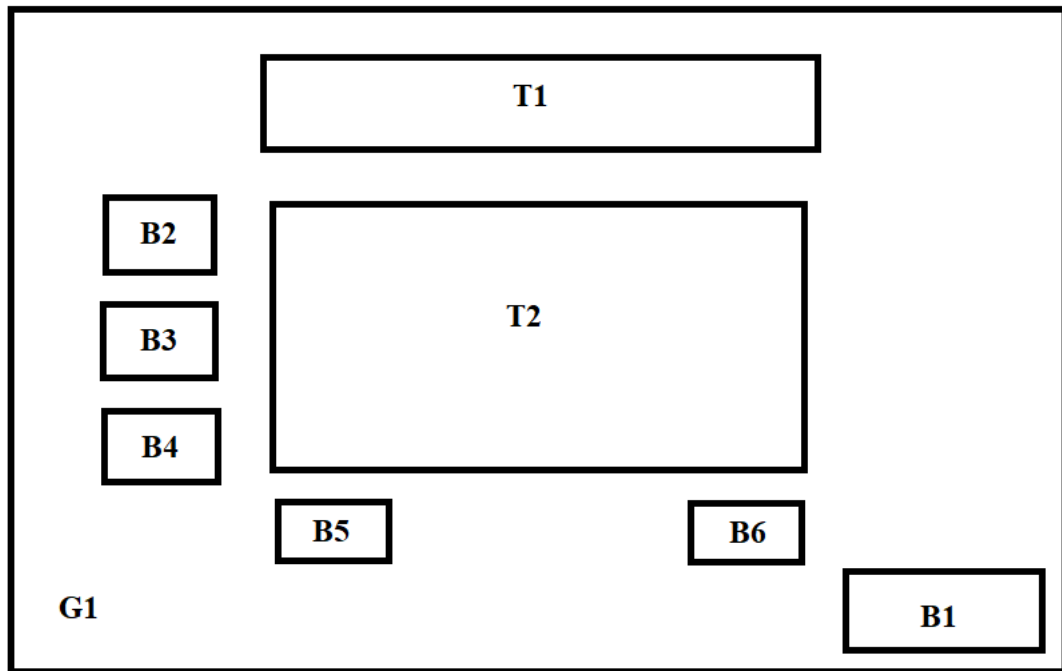


Figure 3-2-1-F5: Storyboard of Lesson

**T1:** Text element showing title of the particular topic.

**G1:** Graphic element showing background image.

**B1:** Graphic element showing “Back” and will link to Lesson Menu.

**B2-B4:** Graphic elements showing title of subtopic and will link to those subtopics.

**B5:** Graphic element showing “Previous” and will link to previous lesson.

**B6:** Graphic element showing “Next” and will link to next lesson.

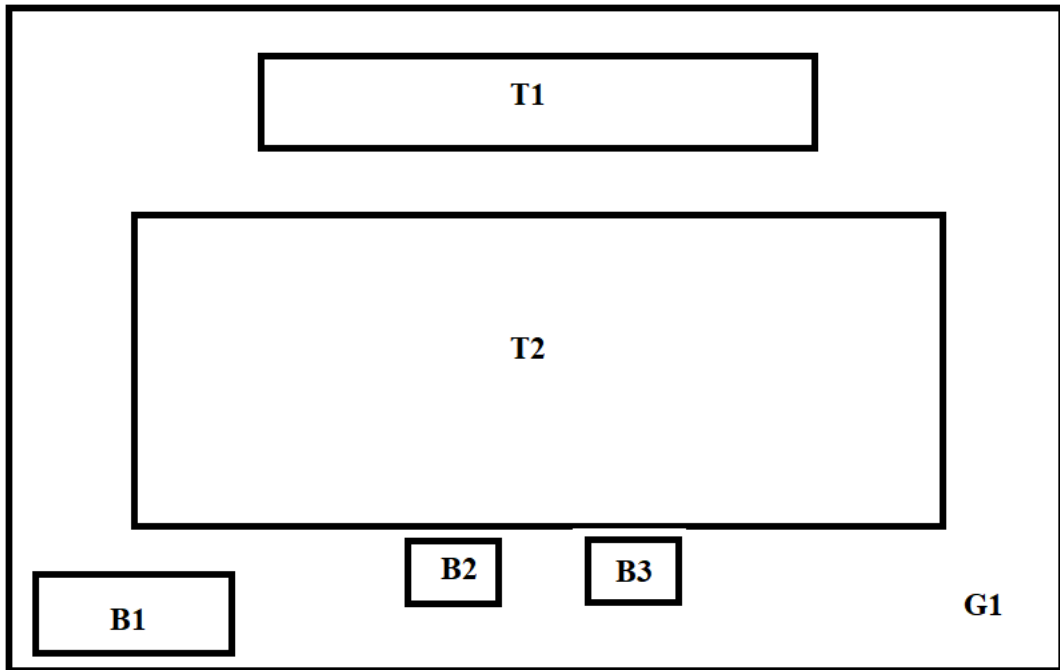


Figure 3-2-1-F6: Storyboard of About Page

- T1:** Text element showing “About”.
- T2:** Text element showing related information of programming.
- G1:** Graphic element showing background image.
- B1:** Graphic element showing “Back” and will link to Main Menu.
- B2:** Graphic element showing “Previous” and will link to previous about page.
- B3:** Graphic element showing “Next” and will link to next about page.

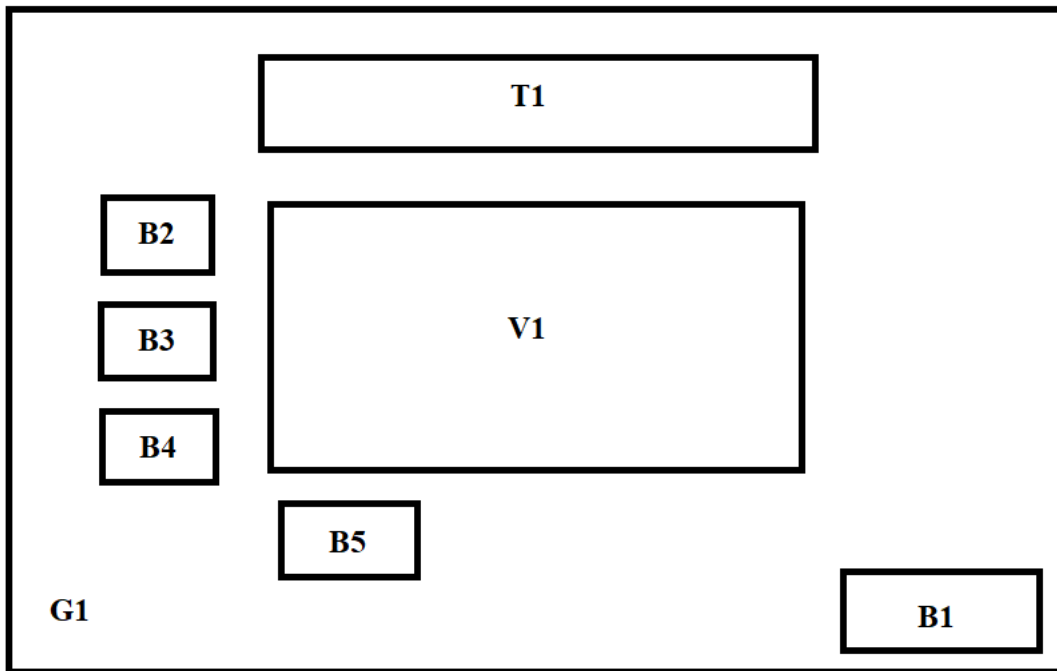


Figure 3-2-1-F7: Storyboard of Video Page

**T1:** Text element showing “Video”.

**G1:** Graphic element showing background image.

**B1:** Graphic element showing “Back” and will link to Main Menu.

**B2-B4:** Graphic elements showing thumbnails of videos and will play the video.

**B5:** Graphic element showing “Pause” and pause or play the video.

**V1:** Video element, which the video will play based on user’s selection.

### 3.3 Component of Multimedia

Several applications used to generate, edit and combine all the essential multimedia elements.

#### 3.3.1 Images

Suitable images searched and downloaded online. Before imported those images to the proposed multimedia courseware, images need to be edited.

First, search for suitable images from Google Image. Screenshot below shows searching on a clipart of board.

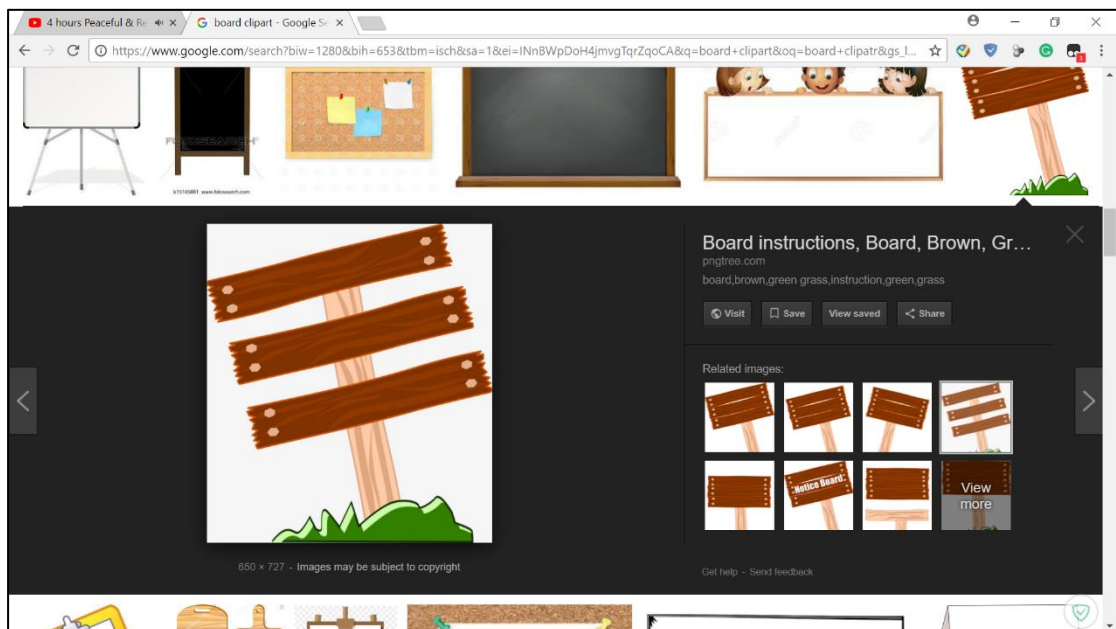


Figure 3-3-1-F1: Search image online

Next, open the image in MeituXiuXiu, a simple photo editor software to crop out the image from white background into transparent background. MeituXiuXiu offers a simple cropping tool, which brings a great help in this case.

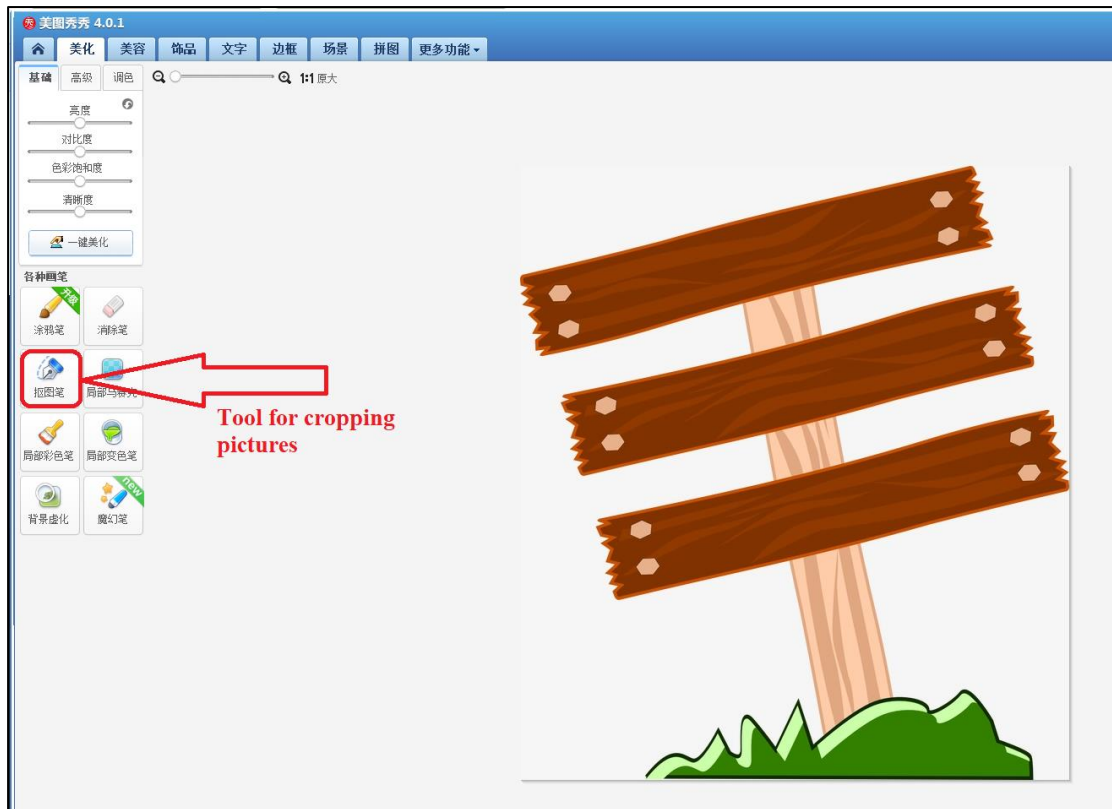


Figure 3-3-1-F2: Choose cropping tool in MeituXiuXiu

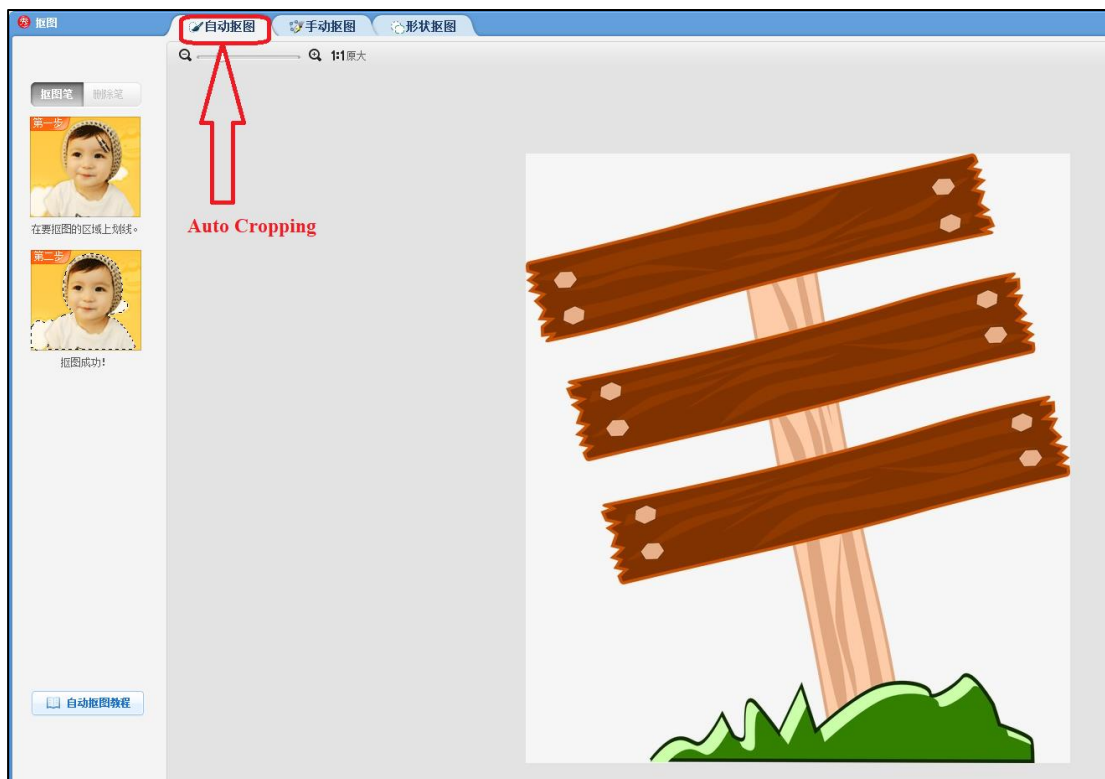


Figure 3-3-1-F3: Choose Auto Cropping

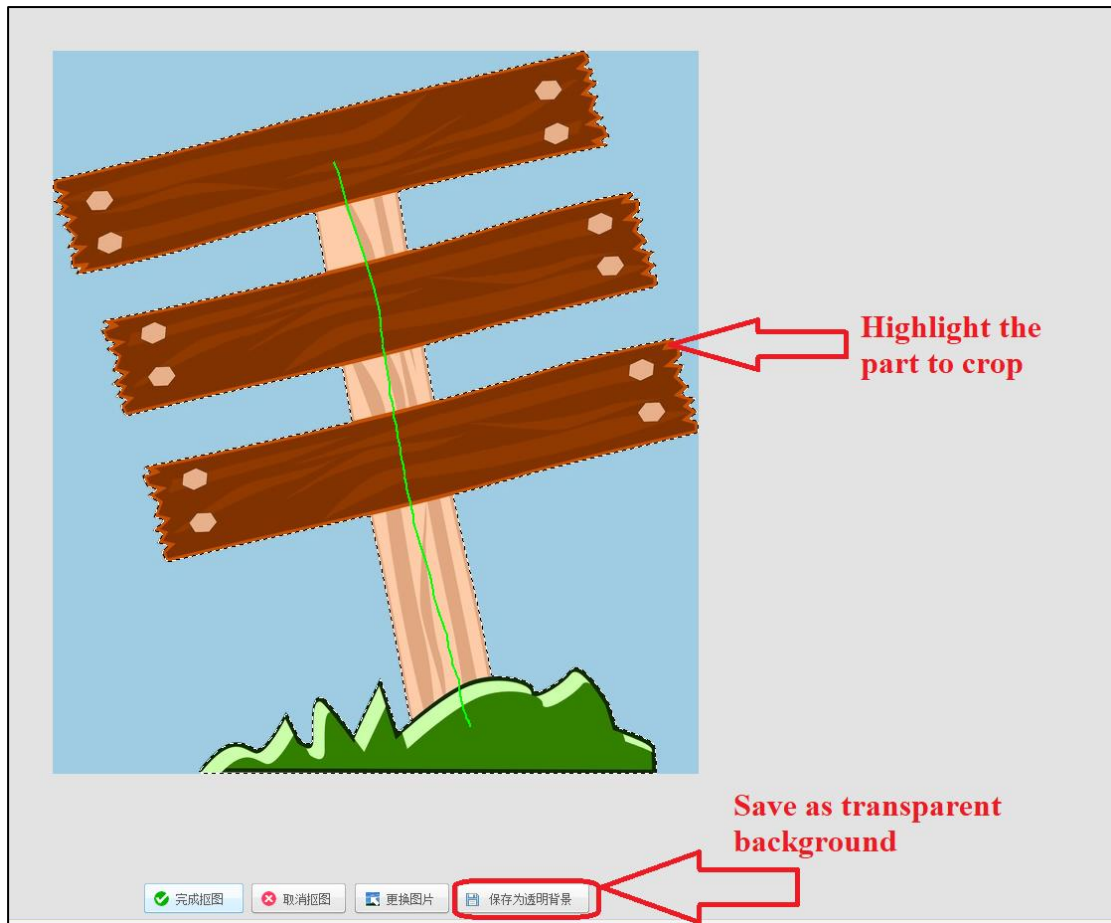


Figure 3-3-1-F4: Crop Images and Save in transparent background

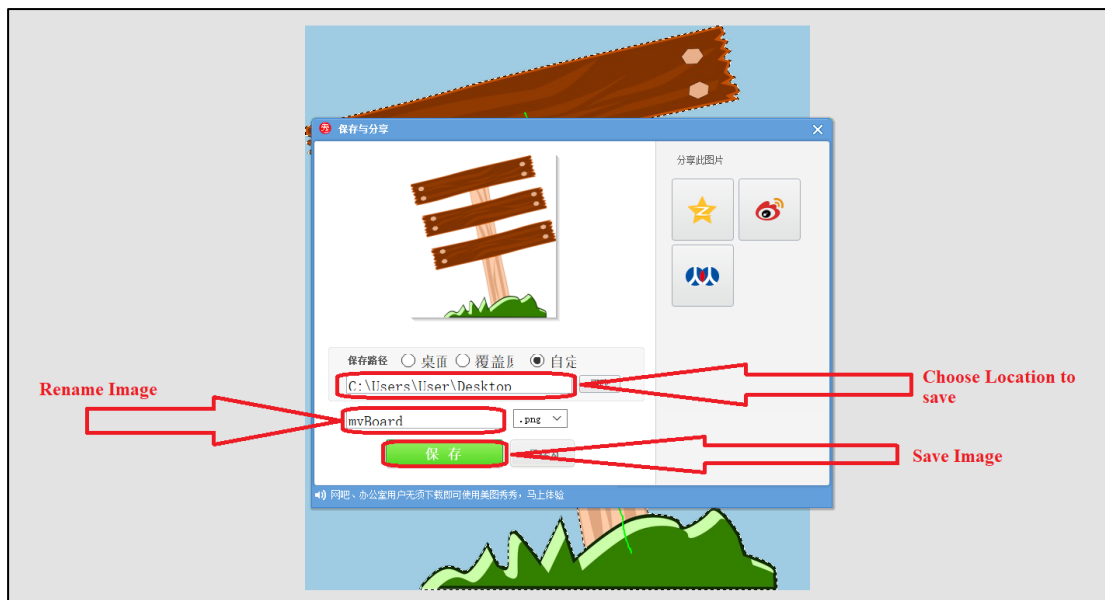


Figure 3-3-1-F5: Rename image and save

Image will be saved in PNG format, as it is an image of transparent background.



Apart from edit image into transparent background, MeituXiuXiu can used to create button-like graphics element as well.



Figure 3-3-1-F6: Select word tool in MeituXiuXiu

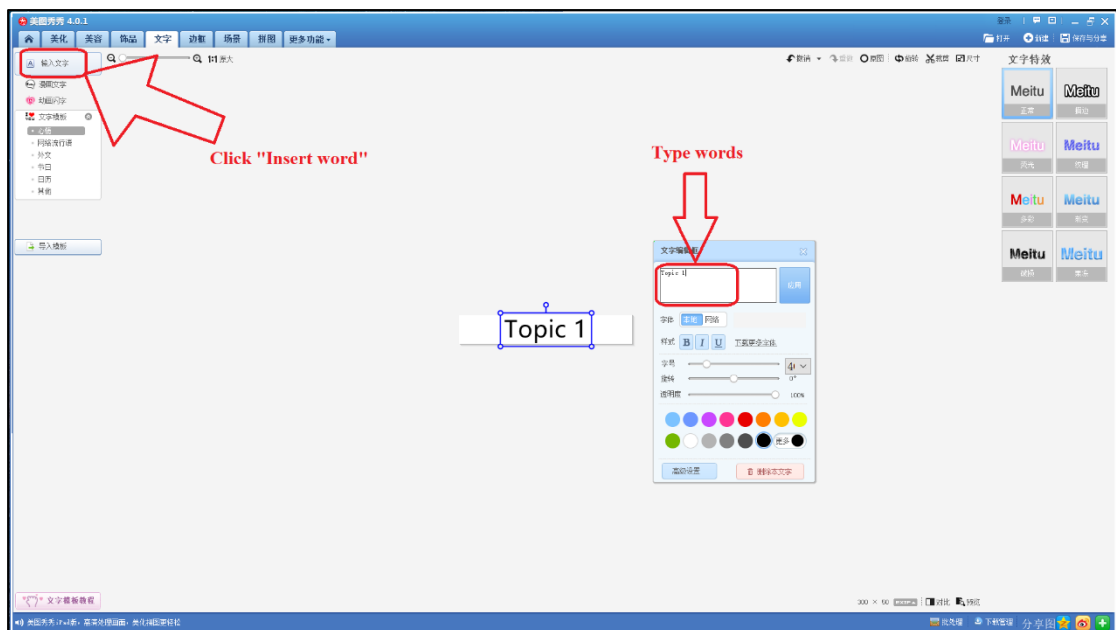
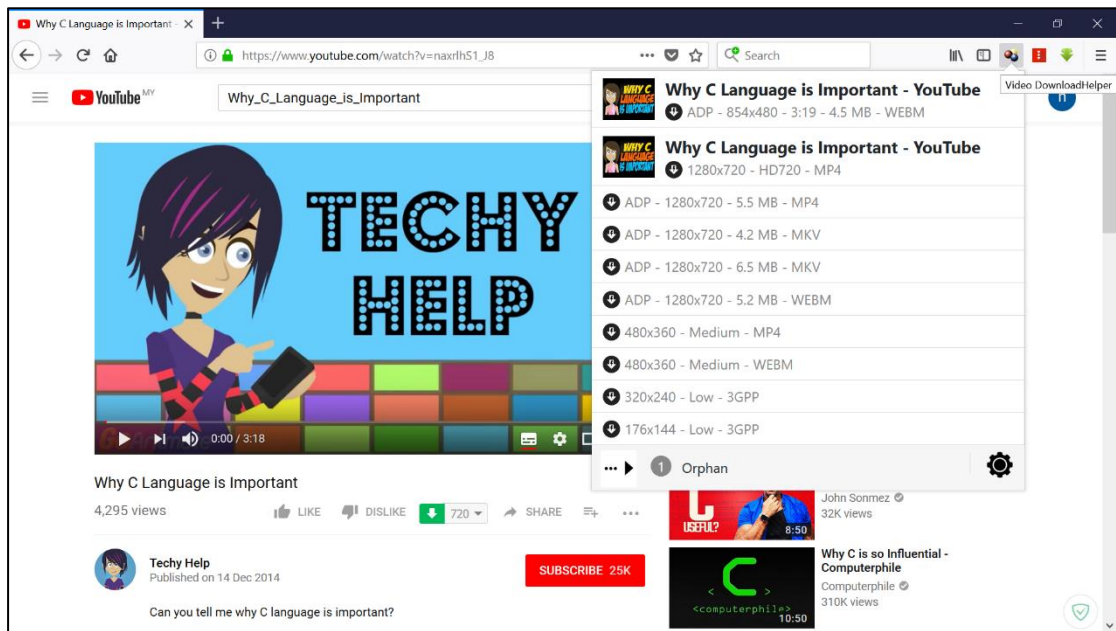


Figure 3-3-1-F7: Insert word and create button

### 3.3.2 Videos

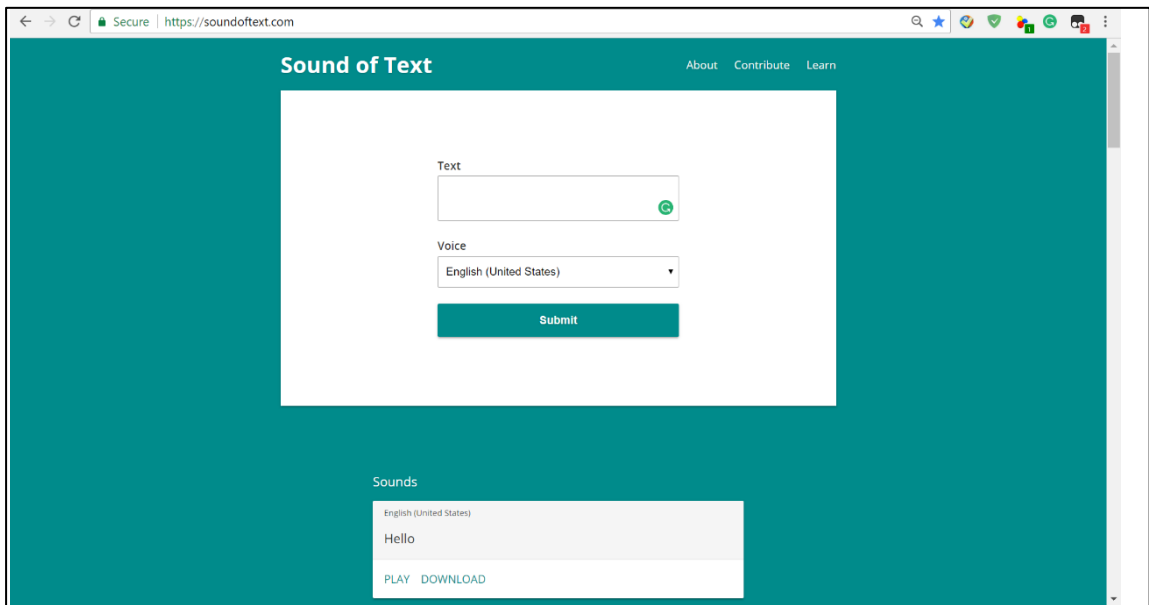


**Figure 3-3-2-F1: Download video**

As shown in figure above, videos used in proposed Interactive Learning Application is mostly downloaded from YouTube. With the Firefox Extension (Video DownloadHelper), as the video in YouTube is playing, the video can be downloaded in different format.

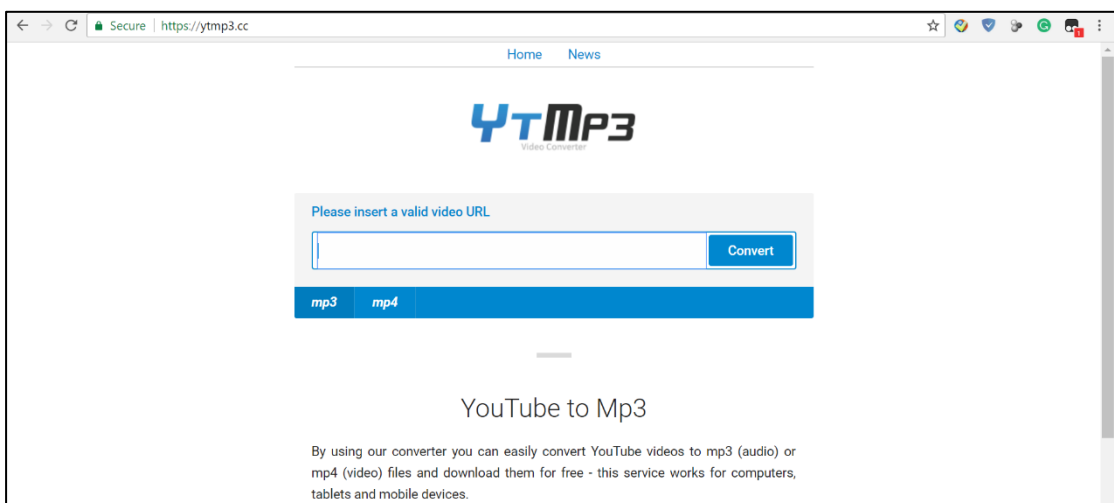
### 3.3.3 Audios

The speaking audio file is generated from the website SoundOfText, by typing in line of words needed, choose the voice preferred (in this case, English (United States) as the default google translate sound), submit and sound will be generated and free for download.



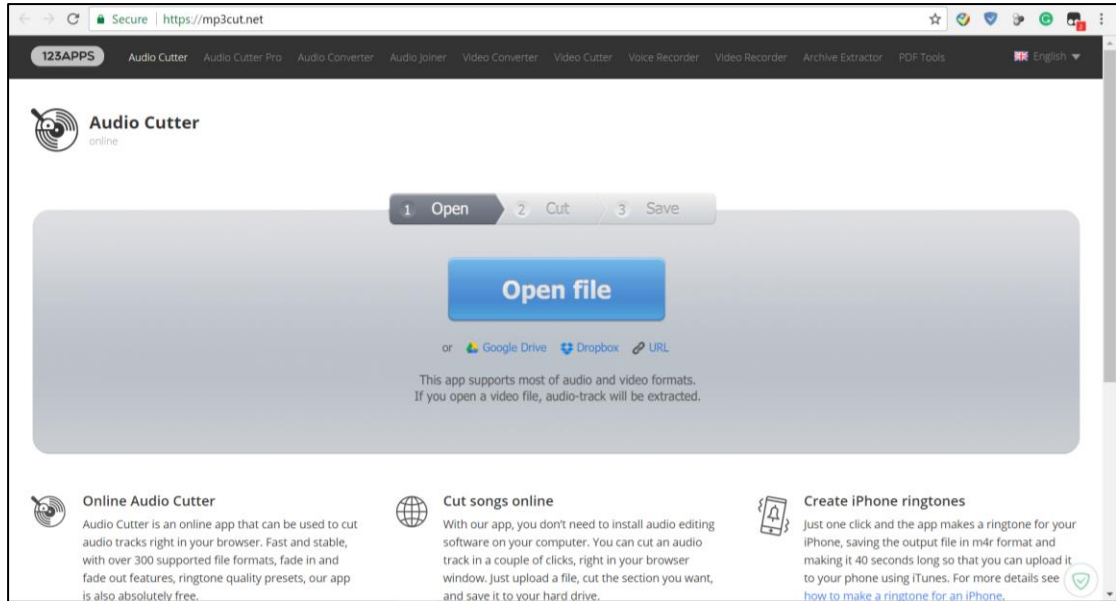
**Figure 3-3-3-F1: Download audio**

Besides, some audio are converted from a video by the online converter, YouTube to MP3 Converter.



**Figure 3-3-3-F1: YouTube to MP3 website**

Sometimes we might only need part of the audio, an online MP3 cutter also give a big hand on cutting audio file.



**Figure 3-3-3-F1: Online MP3 cutter**

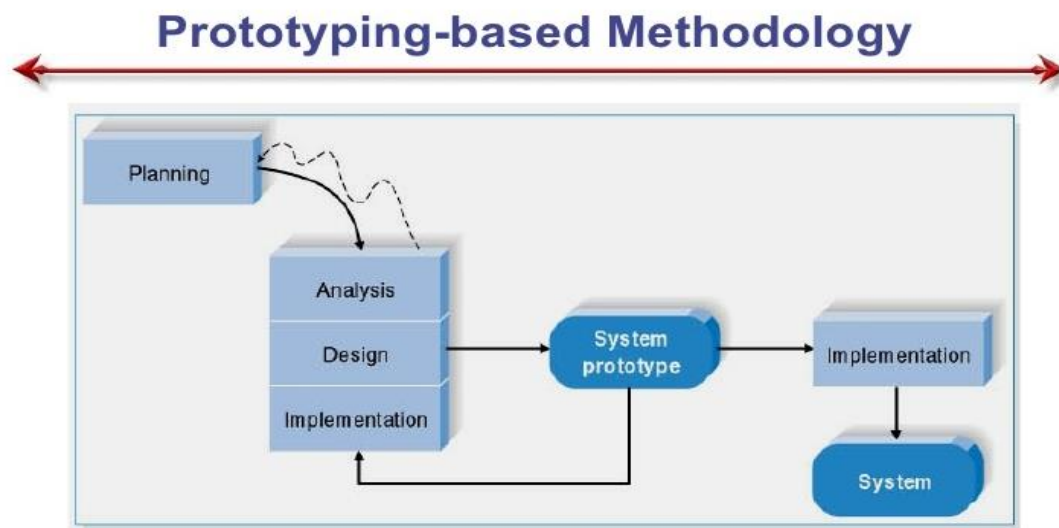
## CHAPTER 4 METHODOLOGY

## 4.1 Design Specifications

## 4.1.1 Development Methodology

Prototyping Methodology was implemented as software development methodology, which a prototype was built, tested, reworked and modified according to feedback from user. The idea of prototyping methodology is to build a prototype before proceeding of design or coding. It is letting client involve in the whole development process, to ensure the function and design is meeting with user requirements. Prototype bringing many advantages, one of them is detecting errors earlier, so that can solve the errors as soon as possible before deliver to client. Also, gain user feedback in sort time and lead to better solutions to cut off the misunderstanding of requirements of the desired system.

With this methodology, a project is planned first, and then analysis, design, and implementation phases will work simultaneous after planning phase. Step is possible to move backward if there are any problem, or changing in requirement, as the process is iterative.



**Figure 4-1-1-F1: Process of Prototyping-Based Methodology**

In planning phase, the very first step is to identify the current problems of the topic and check out effective solutions to overcome these problems. Next, come out with a proposal to present the proposed system and to outline the problem statements and objectives of the proposed system. Additionally, some similar system or project was chosen and shown as benchmarks of proposed system to recognize features and performance of benchmarked system.

Then, in analysis phase, an analysis is to describe the system requirements and specifications that needed to be developed. Unified Modeling Language (UML) is the key to show the event flow, storyboard, and the use case of the system.

After completing analysis phase, continue with design phase and implement phase. First, the development of Graphical User Interface (GUI), which is to draft out the interface and layout of the system in storyboard to have a clearer look on how the project is going to be designed. For a multimedia courseware, GUI is no doubt the most important part, as it decided how much user interest in it, and how easy can it be used. During the process of implementation, testing will be continuously carried out to ensure the system is designed in a correct way and to fix bugs immediately. Additionally, Adobe Director MX 2004 and Scratch2 are the main software to develop the whole system.

During the implementation phase after system prototype, the system is going to be published into executable file format (EXE), and implemented into user's computer.

## **4.2 Development Tools**

### **4.2.1 Macromedia Director MX 2004**

Macromedia Director MX 2004 is a multimedia application authoring platform to make text, graphics, sound, and animation for the multimedia courseware. It is to organize and associate single modules into courseware. More, Macromedia Director supports various format of graphics, audio and video. Moreover, it support Lingo, a scripting language, which allow for advanced developing options.

### **4.2.2 MeituXiuXiu**

MeituXiuXiu is a simple image editing software for editing photos. Features suggested include applying filters, retouching of photos, Collage of photos, framing, and photo decorations etc.

### **4.2.3 Snipping tool**

Snipping tool is a build-in software in the Windows, which assist to snip, crop image, and edit it.

### **4.2.4 Video DownloadHelper Extension in Firefox**

Video DownloadHelper is a complete tool to extract videos and image files from website and save them into hard drive.

### **4.2.5 SoundOfText Website**

SoundOfText is a website, which is to customize text and download it in MP3 format for free.

### **4.2.6 YouTube to MP3 Website**

YouTube to MP3 website is an online converter, which is to convert YouTube video into audio file online, and download it.

#### **4.2.7 Online MP3 Cutter Website**

Online MP3 Cutter website is used to cut unnecessary part of audio file online and download it.

#### **4.2.8 Scratch 2.0 Offline Editor**

A free programming language that allow users to create their own interactive stories, games and animations, in which they can publish and discuss their creations with one another on the official website. Scratch 2 is used to build game stages in proposed Interactive Learning Application.

#### **4.2.9 SWF to EXE Converter**

A converter to convert SWF file to EXE format. As Macromedia Director supported to open EXE file only directly, it is easier to convert the Scratch 2 file to SWF file online, and use the converter to convert it into EXE file.

### **4.3 Specification of Requirements**

#### **4.3.1 User Requirement**

To determine and specify what the user expects from the system. Method used is by researching on similar interactive learning applications, go through journal, forum and blog to determine proposed user requirement in previous work.

#### **4.3.2 Functional Requirement**

This is to outline the function the system should execute in certain circumstances. Functional requirement of Interactive Learning Application are listed as:

- i. User can do game-like tutorial based on specific topic with guidelines and hints in Tutorial module.



- ii. User can do game-like Quiz based on specific topic with increasing challenges in Quiz module.
- iii. User can do game-like Practical based on combinations of topics with increasing challenged in Practical module.
- iv. User can have realization on learning Computer Programming and study concepts in Lesson module.
- v. User can watch videos to know more about what computer programming is, at the same moment of relaxing in Video module.
- vi. User can learn and review the appendixes of Lesson as well as the guideline of the game-like stages in About module.

#### **4.3.3 Non-Functional Requirement**

This is to outline the criteria and measurement the system need to meet. Non-functional requirement of Interactive Learning Application are listed as below.

- i. Easy to use

It should be simple and easy enough for user to use.

- ii. Reliability

When users using the system, no error should be arise.

- iii. Good performance

There should be no delay and quick performance for the response period of the system.

### 4.3.4 System Requirement

#### 4.3.4.1 Software Requirement

**Table 4-3-4-1-T1: Table of Software Requirements**

<b>Software</b>	<b>Requirement</b>
<b>Operating System</b>	<ul style="list-style-type: none"> <li>- Windows XP, 7 (or higher)</li> <li>- 32-bit Operating System (or higher)</li> </ul>
<b>Adobe Flash Player</b>	<ul style="list-style-type: none"> <li>- Adobe Flash Player 10 (or higher)</li> </ul>

#### 4.3.4.2 Hardware Requirement

**Table 4-3-4-2-T1: Table of Hardware Requirement**

<b>Hardware</b>	<b>Requirement</b>
<b>Processor</b>	<ul style="list-style-type: none"> <li>- 1 GHz Processor or higher</li> </ul>
<b>RAM</b>	<ul style="list-style-type: none"> <li>- 1 GB of RAM above</li> </ul>
<b>Free Disk Space</b>	<ul style="list-style-type: none"> <li>- 300 MB of free disk space above</li> </ul>
<b>Graphics Card</b>	<ul style="list-style-type: none"> <li>- 32-bit color (or higher)</li> </ul>
<b>Sound Card</b>	<ul style="list-style-type: none"> <li>- 32-bit sound card (or higher)</li> </ul>

## 4.4 Project Timeline

	Task Name	% Complete	Duration	Start	Finish
1	1 Review Project Proposal	100%	1 day	Thu 1/6/17	Thu 1/6/17
2	2 Meeting Supervisor	100%	1 day	Fri 2/6/17	Fri 2/6/17
3	<b>3 Project Planning</b>	<b>100%</b>	<b>3.5 days</b>	<b>Tue 6/6/17</b>	<b>Fri 9/6/17</b>
4	3.1 Define Problem Statement	100%	0.5 days	Tue 6/6/17	Tue 6/6/17
5	3.2 Define Objective	100%	0.25 days	Tue 6/6/17	Tue 6/6/17
6	3.3 Define Scope	100%	0.25 days	Tue 6/6/17	Tue 6/6/17
7	3.4 Define Background	100%	2 days	Wed 7/6/17	Thu 8/6/17
8	3.5 Meet Supervisor	100%	0.5 days	Fri 9/6/17	Fri 9/6/17
9	3.6 Project Planning Correction and Modification	100%	0 days	Fri 9/6/17	Fri 9/6/17
10	<b>4 Literature Review</b>	<b>100%</b>	<b>8 days</b>	<b>Mon 12/6/17</b>	<b>Wed 21/6/17</b>
11	4.1 Review LightBot	100%	2 days	Mon 12/6/17	Tue 13/6/17
12	4.2 Review CodeMonkey	100%	1 day	Wed 14/6/17	Wed 14/6/17
13	4.3 Review Kodable	100%	2 days	Wed 14/6/17	Thu 15/6/17
14	4.4 Summarize Literature Review	100%	1 day	Fri 16/6/17	Fri 16/6/17
15	4.5 Meet Supervisor	100%	0.5 days	Mon 19/6/17	Mon 19/6/17
16	4.6 Literature Review Correction and Modification	100%	2 days	Tue 20/6/17	Wed 21/6/17
17	<b>5 Design Specification</b>	<b>100%</b>	<b>15 days</b>	<b>Mon 26/6/17</b>	<b>Mon 17/7/17</b>
18	5.1 Define Development Methodology	100%	5 days	Mon 26/6/17	Fri 30/6/17
19	5.2 System Design	100%	5 days	Mon 3/7/17	Fri 7/7/17
20	5.3 Meet Supervisor	100%	0.5 days	Mon 10/7/17	Mon 10/7/17
21	5.4 Data Specification Correction and Modification	100%	5 days	Mon 10/7/17	Fri 14/7/17
22	5.5 FYP 1 Submission	100%	0 days	Mon 17/7/17	Mon 17/7/17
23	<b>6 System Prototype Development</b>	<b>100%</b>	<b>27.25 days</b>	<b>Mon 24/7/17</b>	<b>Wed 30/8/17</b>
24	6.1 Design Main Menu	100%	3 days	Mon 24/7/17	Wed 26/7/17
25	6.2 Design Lesson Page	100%	2 days	Thu 27/7/17	Fri 28/7/17
26	6.3 Design Quiz Page	100%	3 days	Mon 31/7/17	Wed 2/8/17
27	6.4 Design Practical Page	100%	2 days	Thu 3/8/17	Fri 4/8/17
28	6.5 Design Tutorial Page	100%	3 days	Mon 7/8/17	Wed 9/8/17
29	6.6 Design About Page	100%	2 days	Thu 10/8/17	Fri 11/8/17
30	6.7 Design Video Page	100%	2 days	Thu 24/8/17	Fri 25/8/17
31	6.8 Perform Integration Test	100%	1 day	Mon 28/8/17	Mon 28/8/17
32	6.9 Prototype Demonstration	100%	0.25 days	Wed 30/8/17	Wed 30/8/17

Figure 4-4-F1: Fyp 1 Gantt Chart

## CHAPTER 4 METHODOLOGY

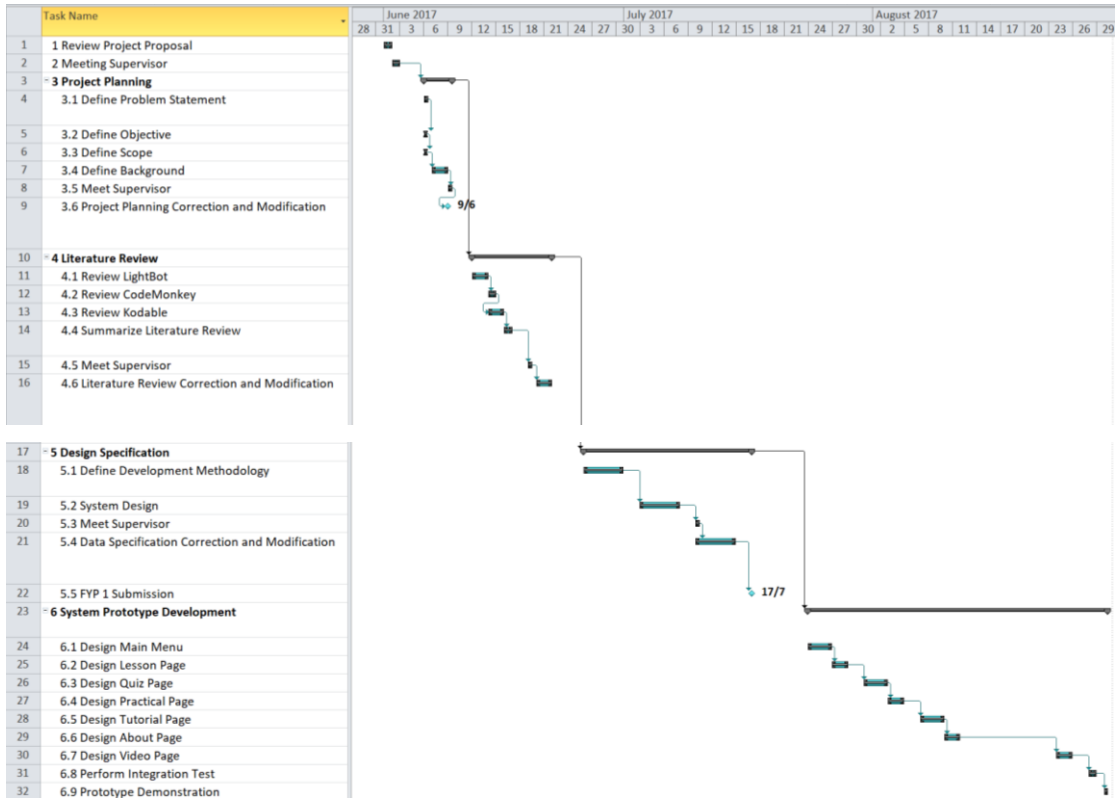


Figure 4-4-F2: Fyp 1 Gantt Chart

## CHAPTER 4 METHODOLOGY

	Task Name	% Complete	Duration	Start	Finish
1	1 Review FYP1	100%	1 day	Mon 22/1/18	Mon 22/1/18
2	2 Meeting Supervisor	100%	0.25 days	Tue 23/1/18	Tue 23/1/18
3	<b>3 Documentation</b>	<b>100%</b>	<b>52 days</b>	<b>Wed 24/1/18</b>	<b>Thu 5/4/18</b>
4	<b>3.1 Literature Review</b>	<b>100%</b>	<b>1.25 days</b>	<b>Wed 24/1/18</b>	<b>Thu 25/1/18</b>
5	3.1.1 Define Multimedia	100%	1 day	Wed 24/1/18	Wed 24/1/18
6	3.1.2 Meeting Supervisor	100%	0.25 days	Thu 25/1/18	Thu 25/1/18
7	<b>3.2 System Design</b>	<b>100%</b>	<b>5 days</b>	<b>Mon 29/1/18</b>	<b>Fri 2/2/18</b>
8	3.2.1 Use Case Diagram	100%	1 day	Mon 29/1/18	Mon 29/1/18
9	3.2.2 Correct Storyboard	100%	1 day	Tue 30/1/18	Tue 30/1/18
10	3.2.3 Multimedia Component	100%	1 day	Wed 31/1/18	Wed 31/1/18
11	3.2.4 Meeting Supervisor	100%	1 day	Fri 2/2/18	Fri 2/2/18
12	<b>3.3 Methodology</b>	<b>100%</b>	<b>5 days</b>	<b>Mon 5/2/18</b>	<b>Fri 9/2/18</b>
13	3.3.1 Development Tools	100%	1 day	Mon 5/2/18	Mon 5/2/18
14	3.3.2 Requirement Specifications	100%	3 days	Tue 6/2/18	Thu 8/2/18
15	3.3.3 Meeting Supervisor	100%	1 day	Fri 9/2/18	Fri 9/2/18
16	<b>3.4 System Implementation</b>	<b>100%</b>	<b>4 days</b>	<b>Mon 2/4/18</b>	<b>Thu 5/4/18</b>
17	3.4.1 Screenshot of System	100%	1 day	Mon 2/4/18	Mon 2/4/18
18	3.4.2 Explain Screenshot of System	100%	1 day	Tue 3/4/18	Tue 3/4/18
19	3.4.3 Implementation Issues and Challenges	100%	1 day	Wed 4/4/18	Wed 4/4/18
20	3.4.4 Meeting Supervisor	100%	1 day	Thu 5/4/18	Thu 5/4/18
21	4 Submission of FYP 2	100%	0 days	Mon 9/4/18	Mon 9/4/18
22	<b>5 System Development</b>	<b>100%</b>	<b>40.25 days</b>	<b>Wed 21/2/18</b>	<b>Wed 18/4/18</b>
23	5.1 Enhance Main Menu	100%	3 days	Wed 21/2/18	Fri 23/2/18
24	5.2 Enhance Lesson Page	100%	2 days	Mon 26/2/18	Tue 27/2/18
25	5.3 Enhance Quiz Page	100%	3 days	Mon 5/3/18	Wed 7/3/18
26	5.4 Enhance Practical Page	100%	2 days	Mon 12/3/18	Tue 13/3/18
27	5.5 Enhance Tutorial Page	100%	3 days	Mon 19/3/18	Wed 21/3/18
28	5.6 Enhance About Page	100%	2 days	Mon 26/3/18	Tue 27/3/18
29	5.7 Enhance Video Page	100%	2 days	Thu 29/3/18	Fri 30/3/18
30	5.8 Meeting Supervisor	100%	1 day	Fri 30/3/18	Fri 30/3/18
31	5.9 Perform Integration Test	100%	1 day	Mon 2/4/18	Mon 2/4/18
32	5.10 System Demonstration	100%	0.25 days	Wed 18/4/18	Wed 18/4/18

**Figure 4-4-F3: Fyp 2 Gantt Chart**

## CHAPTER 4 METHODOLOGY

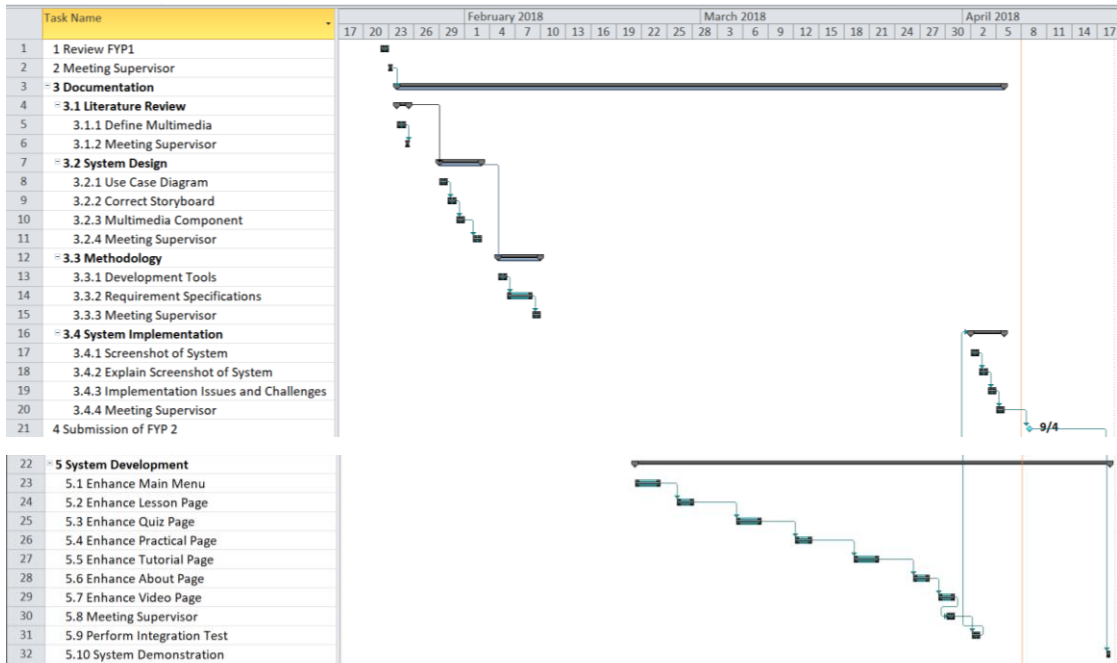


Figure 4-4-F4: Fyp 2 Gantt Chart

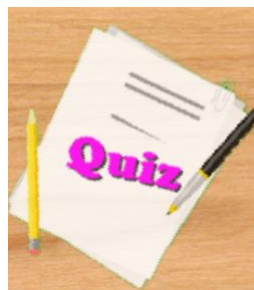
## CHAPTER 5 SYSTEM IMPLEMENTATION

### 5.1 Screenshot



**Figure 5-1-F1: Main Menu**

Figure-5-1-F1 shows the main menu of the Interactive Learning Application, it is an animated splash screen, which may assist in attract attention of users. When mouse is over some of the icon, words indicating modules will show as figures below.



**Figure 5-1-F2: Quiz Module Mouse Over**



**Figure 5-1-F3: Tutorial Module Mouse Over**



Figure 5-1-F4: Practical Module Mouse Over



Figure 5-1-F5: Lesson Module Mouse Over



Figure 5-1-F6: About Module Mouse Over



Figure 5-1-F7: Exit Mouse Over



Figure 5-1-F8: Video Module Mouse Over

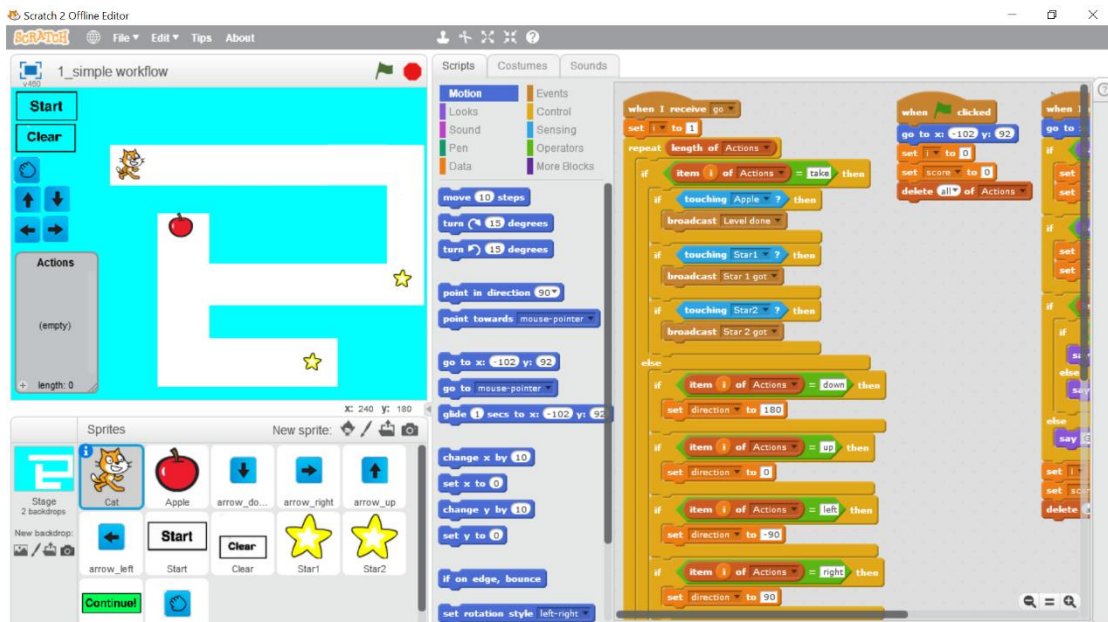
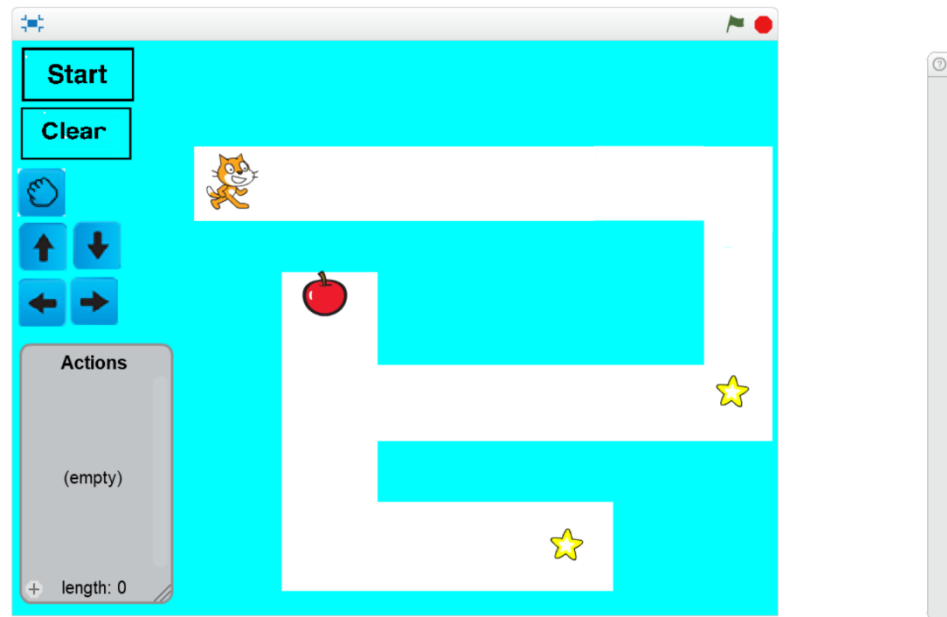


Figure 5-1-F9: Game Stage





**Figure 5-1-F10: Game Stage Full Screen**

Game Stages is linked from Quiz Module, Tutorial Module, and Practical Module. It will be link to Scratch 2 application. Despite of playing with the game and learning basic concepts of computer programming, code block in right side of Figure-5-1-F9 showing how the game is built, user also may go through it if they familiarize with the code blocks, or they may try to modify it.

In Tutorial Module, guides will be given while users go through the game. In Quiz Module, stages are similar to Tutorial Module but no hints are given. In Practical Module, there are increasing challenges where few topics will be combined into one stage.

Figure-5-1-F10 shows the full screen of the game, in which when users are doing the practical, they will not drag those sprites (images/icons). As sprites can be easily dragging in edit mode (Figure-5-1-F9).

For the game to start, user need to click on the green flag on the top-right of Figure-5-1-F10. Then, user may click on any dark blue icon (up, down, right, left and take) to add the actions into action list. After finish the action list, click Start, the “cat” will perform actions based on the action list.



Figure 5-1-F11: Sub-Menu

Figure above shows the sub-menu for Lesson Module, Quiz Module, and Tutorial Module, which will be separated into six topics. When mouse over the title, the title will change colour to white to indicate the pointing title as shown below.



Figure 5-1-F12: Sub-Menu Mouse Over

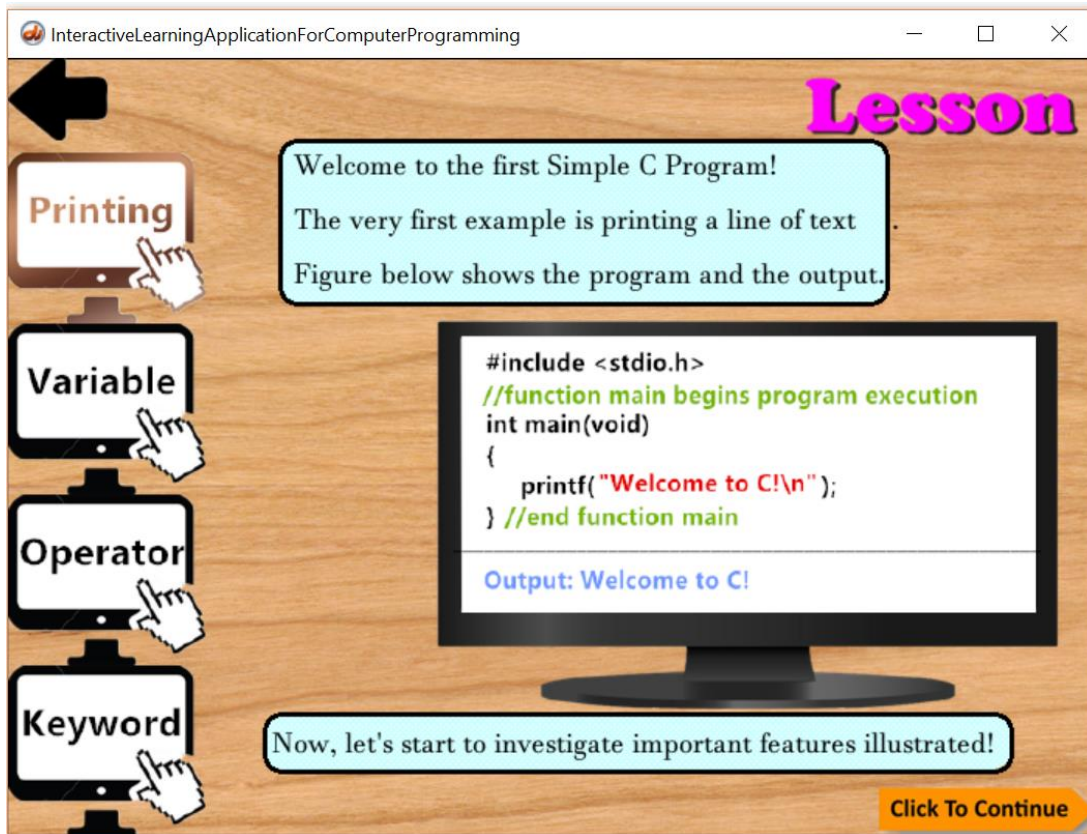


Figure 5-1-F13: Lesson Module

Figure above shows one of the lesson. The lesson are expressed in the way as simple as possible, and the example used was in C programming for entering level. However, the lesson is not mainly targeted on teaching computer programming like what a textbook did, but used for realization on what user have already learnt while they are playing the game.



Figure 5-1-F14: Video Module

The Video Module have five videos available to play, where clicking on the thumbnails and the video shall play on the right side. A pause/play button also available while the video is playing.

Two figures below show the about module page as well as the exit page.



**Figure 5-1-F15: About Module**



**Figure 5-1-F16: Exit Page**

## 5.2 Issues and Challenges in Implementation

Few issues and challenges were met in the implementation stage. First, lack of experience in Macromedia Director, as author is not familiar with the functionality, especially Lingo script, a lot of research on tutorial online are needed. Next, for this learning application, Macromedia Director was used for Menu implementation, and Scratch 2 is for Practical implementation. Scratch 2 is easier for practically development, as well it brings along programming concepts. However, Scratch 2 is not easy for large development, hence, practical part is separate into several file for easy implementation. Furthermore, when trying to link up Menu in Macromedia and hands-on practical in Scratch 2, some problem met when Macromedia Director allows for open application in project, but only EXE file is allowed to open, whereas other kind of application will not be responding. Hence, an extra action of converting Scratch 2 file into EXE file need to be carry out. Additionally, navigating back to Menu by button from Scratch practical is not allowed. The only action to perform is close the practical and return. Lastly, target user need guidance of guardian or teacher, as no limitation on which stage of practical should play out first, or which stage of lesson should study first.

## CHAPTER 6 TESTING

### 6.1 Alpha Testing

Alpha testing is in-house testing carried out by developers. Thus, alpha testing is conducted to test the software development for the interactive learning application. Alpha testing aim to detect errors or bugs in the system as well as assist in reduce inaccuracy arise when implementing the user evaluation of the Interactive Learning Application.

#### 6.1.1 Unit Testing

To test each unit of source code. For Interactive Learning Application, unit testing was conducted to test on each module and to ensure each module functioning as expected. If any error found during the unit testing, actions will be carried out to solve the error immediately to ensure the system can perform in correct way.

#### A) Main Menu Page

**Table 6-1-1-T1: Table of Main Menu Page Testing**

No	Test Field	Estimated Result	Remark
1.	Mouse over the “Quiz”	A text of “Quiz” displayed	✓
2.	Mouse over the “Tutorial”	A text of “Tutorial” displayed	✓
3.	Mouse over the “Practical”	A text of “Practical” displayed	✓
4.	Mouse over the “Lesson”	A text of “Lesson” displayed	✓
5.	Mouse over the “Video”	A text of “Video” displayed	✓
6.	Mouse over the “About”	A text of “About” displayed	✓
7.	Mouse over the “Exit”	A text of “Exit” displayed	✓
8.	Click on “Quiz”	Go to Quiz Module Page	✓
9.	Click on “Tutorial”	Go to Tutorial Module Page	✓
10.	Click on “Practical”	Go to Practical Module Page	✓
11.	Click on “Lesson”	Go to Lesson Module Page	✓

12.	Click on “Video”	Go to Video Module Page	✓
13.	Click on “About”	Go to About Module Page	✓
14.	Click on “Exit”	Go to Exit Page	✓

### B) Tutorial Module Page

**Table 6-1-1-T2: Table of Tutorial Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on “Simple Workflow”	Go to Scratch stage for Simple Workflow	✓
3.	Click on “Procedure”	Go to Scratch stage for Procedure	✓
4.	Click on “Looping”	Go to Scratch stage for Looping	✓
5.	Click on “Variable Substitution”	Go to Scratch stage for Variable Substitution	✓
6.	Click on “Conditional”	Go to Scratch stage for Conditional	✓
7.	Click on “Array-pointer”	Go to Scratch stage for Array-pointer	✓

**C) Quiz Module Page****Table 6-1-1-T3: Table of Quiz Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on “Simple Workflow”	Go to Scratch stage for Simple Workflow	✓
3.	Click on “Procedure”	Go to Scratch stage for Procedure	✓
4.	Click on “Looping”	Go to Scratch stage for Looping	✓
5.	Click on “Variable Substitution”	Go to Scratch stage for Variable Substitution	✓
6.	Click on “Conditional”	Go to Scratch stage for Conditional	✓
7.	Click on “Array-pointer”	Go to Scratch stage for Array-pointer	✓

**D) Practical Module Page****Table 6-1-1-T4: Table of Practical Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on “Start Practical”	Go to Scratch stage for Practical	✓

**E) Lesson Module Page****Table 6-1-1-T5: Table of Lesson Page Testing**

No	Test Field	Estimated Result	Remark
----	------------	------------------	--------



1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on “Simple Workflow”	Go to Simple Workflow lesson	✓
3.	Click on “Procedure”	Go to Procedure lesson	✓
4.	Click on “Looping”	Go to Looping lesson	✓
5.	Click on “Variable Substitution”	Go to Variable Substitution lesson	✓
6.	Click on “Conditional”	Go to Conditional lesson	✓
7.	Click on “Array-pointer”	Go to Array-pointer lesson	✓

### F) Video Module Page

**Table 6-1-1-T6: Table of Video Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on Thumbnail for Video 1	Play Video 1	✓
3.	Click on Thumbnail for Video 2	Play Video 2	✓
4.	Click on Thumbnail for Video 3	Play Video 3	✓
5.	Click on Thumbnail for Video 4	Play Video 4	✓
6.	Click on Thumbnail for Video 5	Play Video 5	✓
7.	Click on Pause button for Video 1	Pause (or Play if pausing) Video 1	✓
8.	Click on Pause button for Video 2	Pause (or Play if pausing) Video 2	✓
9.	Click on Pause button for Video 3	Pause (or Play if pausing) Video 3	✓
10.	Click on Pause button for Video 4	Pause (or Play if pausing) Video 4	✓
11.	Click on Pause button for Video 5	Pause (or Play if pausing) Video 5	✓

**G) About Module Page****Table 6-1-1-T7: Table of About Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Go to Main Menu Page	✓
2.	Click on “Previous”	Go to previous About Page	✓
3.	Click on “Next”	Go to next About Page	✓

**H) Exit Page****Table 6-1-1-T8: Table of Exit Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Yes”	Exit the program	✓
2.	Click on “No”	Go to Main Menu Page	✓

**I) Lesson Sub-topic Page****Table 6-1-1-T9: Table of Sub Topic Page Testing**

No	Test Field	Estimated Result	Remark
1.	Click on “Back”	Link to Lesson Module Page	✓
2.	Click on Title button for subtopics	Go to particular subtopic	✓
3.	Click on “Previous”	Go to previous page of particular subtopic	✓
4.	Click on “Next”	Go to next page of particular subtopic	✓

**J) Scratch Stage****Table 6-1-1-T10: Table of Scratch Stage Testing**

<b>No</b>	<b>Test Field</b>	<b>Estimated Result</b>	<b>Remark</b>
<b>1.</b>	Click on “Back”	Go to Main Menu Page	✓
<b>2.</b>	Click on “Actions”	Focus on “Actions” (to add steps into Action list)	✓
<b>3.</b>	Click on “Procedure”	Focus on “Procedure” (to add steps into Procedure list)	✓
<b>4.</b>	Click on “Up” arrow	Add “Up” into focused list	✓
<b>5.</b>	Click on “Down” arrow	Add “Down” into focused list	✓
<b>6.</b>	Click on “Left” arrow	Add “Left” into focused list	✓
<b>7.</b>	Click on “Right” arrow	Add “Right” into focused list	✓
<b>8.</b>	Click on “Take” figure	Add “Take” into focused list	✓
<b>9.</b>	Click on “P” figure	Add “P” into focused list	✓
<b>10.</b>	Click on “Start” button	Start the actions of Action list	✓
<b>11.</b>	Click on “Clear” button	Clear all actions in all list	✓

### 6.1.2 Integration Testing

To examine integration or interfaces between fragments of the system. Integration testing is a group testing, which is significant to discover faults before any issues with the interfaces amongst system component.

#### 6.1.2.1 Images and Sound

**Table 6-1-2-1-T1: Table of Image and Sound Testing**

No	Test Field	Estimated Result	Remark
1.	Import image	Images imported	✓
2.	Import sound	Sounds imported	✓
3.	Integrate sound into frame	Background music integrated	✓
4.	Integrate sound into button	Sound effect integrated	✓

#### 6.1.2.2 SWF of Each Module

**Table 6-1-2-1-T1: Table of SWF Testing**

No	Test Field	Estimated Result	Remark
1.	Integration between different module	Each module integrated	✓

## CHAPTER 7 CONCLUSION

### 7.1 Conclusion

Programming skills are essential in current digital world. Primary school kids are encouraged to start on learning those knowledge and skills, to attract them on learning the basic knowledge of computer programming, such as simple workflow, looping and procedure, an interactive multimedia will be a good choice.

In fact, multimedia courseware is more interesting than the hardcopy material, and it can provide the interactivity between user and the system. Moreover, using multimedia may help on easier understanding.

As a consequent, user can gain the knowledge of computer programming when they use the “**Interactive Learning Application for Computer Programming**”.

### 7.2 Problem Faced

By developing this interactive learning application for computer programming, some problem faced as follows:

- Simplifying the courses

As the target user are range from 7 to 9 years old, terms used in textbook referred are not suitable, and courses covered might be too hard for target user of this learning application. Thus, some time were spent on simplifying the courses into simple English, as well as pick up simple example for explanation.

- Design hands-on practical on Macromedia Director

At very first of the development, hands-on practical were planned to be designed on Macromedia Director itself, however, some problem encountered such as actions performed not as expecting. After discussing with supervisor, Scratch 2 application was suggested to develop game stages as hands-on practical. More,

some time are wasted on researching for how to link the Scratch 2 application to Macromedia Director (where menu is build).

### **7.3 Future Work**

Since the Interactive Learning Application for Computer Programming is targeted on lower primary school students, course provided are lesser and introduced in simple ways. Therefore, future lesson such as proper coding and enhanced knowledge will not provide in the current learning application. For the future project, those lessons should be added which target to higher education level of students.

Also, the design of the learning application will be improved by increasing interesting image, animation, and audio to increase attraction of kids to use the application.

Besides, more stages of hands-on practical will be add in future project to enhance user's understanding on concept on computer programming.

## BIBLIOGRAPHY

- Biggs, J. (2013). *Light-Bot Teaches Computer Science With A Cute Little Robot And Some Symbol-Based Programming*. [online] TechCrunch. Available at: <https://techcrunch.com/2013/06/26/light-bot-teaches-computer-science-with-a-cute-little-robot-and-some-symbol-based-programming/> [Accessed 21 Nov. 2016].
- Bindel, A. (2016). *Lightbot - Programming Puzzles - App Review*. [online] Commonsensemedia.org. Available at: <https://www.commonsensemedia.org/app-reviews/lightbot-programming-puzzles> [Accessed 21 Nov. 2016].
- BrainPOP Educators. (2017). *Computer Programming Background Information for Teachers, Parents, and Caregivers | BrainPOP Educators*. [online] Available at: <https://educators.brainpop.com/teaching-tip/computer-programming-background-information-teachers-parents-caregivers/> [Accessed 7 Jun. 2017].
- Chioran, A. (2016). *5 benefits of multimedia learning*. [online] Nuiteq.com. Available at: <https://www.nuiteq.com/company/blog/5-benefits-of-multimedia-learning> [Accessed 21 Mar. 2018].
- Crawley, D. (2014). *12 games that teach kids to code — and are even fun, too*. [online] VentureBeat. Available at: <http://venturebeat.com/2014/06/03/12-games-that-teach-kids-to-code/> [Accessed 18 Nov. 2016].
- deBoucher, A. (2016). *Lightbot Reviews | edshelf*. [online] edshelf. Available at: <https://edshelf.com/tool/lightbot/> [Accessed 21 Nov. 2016].
- En.wikipedia.org. (2016). *Lightbot*. [online] Available at: <https://en.wikipedia.org/wiki/Lightbot> [Accessed 21 Nov. 2016].
- En.wikipedia.org. (2016). *List of educational programming languages*. [online] Available at: [https://en.wikipedia.org/wiki/List\\_of\\_educational\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_educational_programming_languages) [Accessed 21 Nov. 2016].

- En.wikipedia.org. (n.d.). *Adobe Director*. [online] Available at: [https://en.wikipedia.org/wiki/Adobe\\_Director](https://en.wikipedia.org/wiki/Adobe_Director) [Accessed 20 Mar. 2018].
- En.wikipedia.org. (n.d.). *MeituPic*. [online] Available at: <https://en.wikipedia.org/wiki/MeituPic> [Accessed 20 Mar. 2018].
- Englishlearningcanbefun.blogspot.my. (2011). *Benefits of interactive multimedia courseware | English learning can be fun*. [online] Available at: <http://englishlearningcanbefun.blogspot.my/2011/01/benefits-of-interactive-multimedia.html> [Accessed 20 Mar. 2018].
- Gudmundsen, J. (2014). *Kodable Review - TechWithKids.com*. [online] Techwithkids.com. Available at: [http://www.techwithkids.com/Review\\_SR01117S\\_kodable](http://www.techwithkids.com/Review_SR01117S_kodable) [Accessed 21 Nov. 2016].
- Hick, S. (1997). *Benefits of Interactive Multimedia Courseware*. Available at: <http://http-server.carleton.ca/~shick/mypage/benifit.html> [Accessed 12 March 2018]
- Hofstetter, F. (2001). *Multimedia Literacy*, 3rd edn. McGraw Hill, New York. [Accessed 12 March 2018]
- Jing, Y. (2005). *Courseware*. Available at: <http://whatis.techtarget.com/definition/courseware> [Accessed 12 March 2018]
- Kodable, (2016). *5 Reasons to Teach Kids to Code*. [image] Available at: <http://resources.kodable.com/kodableInfographic.png> [Accessed 18 Nov. 2016].
- Mann, J. (2013). *Kodable Reviews | edshelf*. [online] edshelf. Available at: <https://edshelf.com/tool/kodable/> [Accessed 21 Nov. 2016].
- MISSIO, E. (2015). *Why Kids Should Learn To Code (And How To Get Them Started) | Learning*. [online] CBC Parents. Available at: <http://www.cbc.ca/parents/learning/view/why-kids-should-learn-to-code-and-how-to-get-them-started> [Accessed 18 Nov. 2016].



- Orsini, L. (2013). *Why Programming Is The Core Skill Of The 21st Century - ReadWrite*. [online] ReadWrite. Available at: <http://readwrite.com/2013/05/31/programming-core-skill-21st-century/> [Accessed 18 Nov. 2016].
- Playcodemonkey.com. (2016). *CodeMonkey*. [online] Available at: <https://www.playcodemonkey.com/> [Accessed 21 Nov. 2016].
- Thomas, D. (2016). *CodeMonkey - Website Review*. [online] Commonsensemedia.org. Available at: <https://www.commonsensemedia.org/website-reviews/codemonkey#> [Accessed 21 Nov. 2016].
- Tynker.com. (2016). *Four Reasons Why Kids Should Learn to Program | Tynker Blog*. [online] Available at: <http://www.tynker.com/blog/articles/ideas-and-tips/four-reasons-why-kids-should-learn-programming/> [Accessed 18 Nov. 2016].
- Wajid, S. (2016). *Benefits of Multimedia in eLearning | 360 Authoring Program-Blog*. [online] 360training.com. Available at: <http://www.360training.com/authoring-program/author-blog/benefits-of-multimedia-in-elearning> [Accessed 21 Mar. 2018].

## Summary of Plagiarism (Turnitin) checking result

### Interactive Learning Application for Computer Programming

#### ORIGINALITY REPORT

<b>5%</b>	%	%	<b>5%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

#### PRIMARY SOURCES

<b>1</b>	<b>Submitted to South Bank University</b> Student Paper	<b>1%</b>
<b>2</b>	<b>Submitted to Nottingham Trent University</b> Student Paper	<b>1%</b>
<b>3</b>	<b>Submitted to University of East London</b> Student Paper	<b>&lt;1%</b>
<b>4</b>	<b>Submitted to Guangzhou Institute of Technology (NCC)</b> Student Paper	<b>&lt;1%</b>
<b>5</b>	<b>Submitted to University of Leicester</b> Student Paper	<b>&lt;1%</b>
<b>6</b>	<b>Submitted to Manchester Metropolitan University</b> Student Paper	<b>&lt;1%</b>
<b>7</b>	<b>Submitted to University of Northampton</b> Student Paper	<b>&lt;1%</b>
<b>8</b>	<b>Submitted to University of West London</b> Student Paper	<b>&lt;1%</b>

9	Submitted to University of Reading Student Paper	<1%
10	Submitted to Barking College Student Paper	<1%
11	Submitted to Glasgow Caledonian University Student Paper	<1%
12	Submitted to THE HONG KONG MANAGEMENT ASSOCIATION (HKMA) Student Paper	<1%
13	Submitted to University of Surrey Roehampton Student Paper	<1%
14	Submitted to 9419 Student Paper	<1%
15	Submitted to University of Teesside Student Paper	<1%
16	Submitted to University of Essex Student Paper	<1%
17	Submitted to University of Glamorgan Student Paper	<1%
18	Entregado a University of Essex el 2012-12-07 Student Paper	<1%
19	Submitted to Maastricht School of Management Student Paper	<1%

<b>Universiti Tunku Abdul Rahman</b>			
<b>Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF** \_\_\_\_\_

<b>Full Name(s) of Candidate(s)</b>	
<b>ID Number(s)</b>	
<b>Programme / Course</b>	
<b>Title of Final Year Project</b>	

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
<b>Overall similarity index:</b> _____ % <b>Similarity by source</b> Internet Sources: _____ % Publications: _____ % Student Papers: _____ %	
<b>Number of individual sources listed of more than 3% similarity:</b> _____	
<b>Parameters of originality required and limits approved by UTAR are as follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words</b> <b>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8</b>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

\_\_\_\_\_  
 Signature of Supervisor  
 Name: \_\_\_\_\_

\_\_\_\_\_  
 Signature of Co-Supervisor  
 Name: \_\_\_\_\_

Date: \_\_\_\_\_

Date: \_\_\_\_\_