Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Can ActionScript find a place in introductory programming language: a systematic literature review

By

Jingren Li

A Dissertation submitted in partial fulfilment of the requirements for the degree of MSc. Interactive Digital Media in Computer Science

Supervised by Glenn Strong

Submitted to the University of Dublin, Trinity College

June 2021

Declaration

I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar

I have also completed the Online Tutorial on avoiding plagiarism 'Ready, Steady, write', located at http://tcdie.libguides.com/plagiarism/ready-steady-write

I declare that the work described in this research Paper is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

I agree that Trinity College Library may lend or copy this research Paper upon request.

Jingren Li 04/06/2021

Abstract

The growing demand of programming results in an increase demand for introductory programming courses. To attract students' interest and enable a better learning outcome, several factors are being investigated to improve the introductory programming language. The choice of introductory programming language is one of the most important topics among the area of programming education. As ActionScript has some visual feature, it is to be investigated whether it can be part of the introductory programming language. Because of the end of life for Flash, Adobe has ceased the support of ActionScript. Will this result in a lost of a good introductory programming language? This paper investigates the characteristics of popular existing introductory programming language, in particular the Scratch, Python and Java through a systematic literature review. Then based on the findings, ActionScript will be analysed to find the place among the trending introductory programming language. Though the goal of this paper is positive, the final result is negative. The impact of death of Flash is the great decline in the popularity of ActionScript and the usage of it has been correspondingly reduced as well. Therefore, ActionScript is not to be considered as a good introductory programming language. However, the benefit of instant visual result can be synthesised with existing programmes like Scratch or Python to improve current introductory programming languages.

Acknowledgements

I would like to thank my supervisor, Glenn Strong, for his help and guidance during the preparation of this paper.

I would also like to thank my parents, Xiaofei and SingLee, my girlfriend Siqi for their continuous support throughout my university life.

Finally, thanks also to my friends for their support during my entire degree programme.

Table of Contents

Declaration ..... **i**

Abstract ..... **ii**

Acknowledgements ..... **iii**

Table of Figures ..... **v**

Table of Tables ..... **v**

Chapter 1 Introduction **1**

Chapter 2 Scope **3**

Chapter 3 Methodology **4**

3.1 Research question 4

3.2 Conducting searches 5

3.3 Selecting studies from the search 8

3.4 Filtering 8

3.5 Data Extraction 9

Chapter 4 Literature review of Scratch, Python and Java **10**

4.1 Overview of introductory programming language 10

4.1.1 Scratch 11

4.1.2 Java 12

4.1.3 Python 13

4.1.4 Summary 14

4.2 Input method 14

4.3 Syntax and Semantics 17

4.4 Acceptance 20

4.5 Summary 24

4.5.1 RQ1 What are the target groups for Scratch, Python and Java as the introductory programming language? 24

4.5.2 RQ2 What are the advantages and disadvantages of the introductory programming languages mentioned above to different target group? 25

Chapter 5 ActionScript 27

5.1 Properties 27

5.2 Impact of Flash EOL 29

5.3 Summary 30

5.3.1 RQ3 What can ActionScript provide to fit in the group of introductory programming languages? 30

Chapter 6 Conclusion 30

Chapter 7 Limitation and future work 31

Bibliography 34

Table of Figures

Figure 1 paper count for initial search 7

Figure 2 Distribution of 14 papers related with Scratch based on published year 12

Figure 3 Distribution of 12 papers related with Java based on published year 13

Figure 4 Distribution of 10 papers related with Python based on published year 14

Table of Tables

Table 1 Overview of papers that conduct experiments 20

Chapter 1 Introduction

The term programming has been defined as describing calculations, defining functions and defining and treating objects[1]. It is a useful skill and has great potential in career development. The demand for programmers has increased in recent years and because of that, more and more people in different age groups seek to learn programming. As a result, the demand for introductory programming courses has also increased tremendously.

However, programming is difficult to learn. It is suggested that ten years are needed for a novice programmer to become an expert programmer[2]. Because of the difficulty, the importance of programming education has grown as well. There are a lot of papers that discuss the difficulties of learning and teaching programming[3] from various perspectives such as programming tools and teaching methods. The trending topic among programming education would be a game approach[4] or a robot platform[5] teaching method which provides an easier and faster entry for novice programmers.

While most of the papers discuss the teaching method[6][7], there are few papers discussing the choice of first programming language. A recent review reveals that the most considered criteria of choosing introductory programming language in college is the ease of learning basic programming concepts[8]. It is stated that most of the students lose their interest in programming because of the amount of theoretical concepts and techniques that they need to learn[9]. But without such theoretical concepts, the learner will often make errors such as syntax error or logic error that causes bugs in their program and they could not understand the error because of the lack of theoretical knowledge for a certain programming language.

Different programming languages can have different programming paradigms which provide different functions and levels of difficulties to the learners. The most popular paradigm is object-oriented programming, it focuses on objects that contain attributes and methods[10].

Besides the paradigms, there are syntax and interface differences between each programming language and the most obvious difference would be block-based programming and text-based programming. Block-based programming uses a visual

programming approach to code the program such as Scratch, Scratchjr[11], Alice[12]. Text-based programming refers to using text as an interface to code the program such as Python, Java, C, Arduino[13]. The differences between paradigms, syntax and interface distinguish each of the programming from each other and provide different entry barriers for the novice programmer. Thus, it is crucial for the introductory programming course to use the suitable first programming language for the learner. There are some papers that have reviewed different programming languages as the first choice. The recent review by Andrew et al. [14] provided a comprehensive overview of the introductory programming from the aspects of student, teacher and programming language. Most of the reviews for the introductory programming language or first programming language choice are Java, Python, C++, Scratch[15]. The reviews from the papers were being performed under different student groups ranging from elementary students[13] to college students[16]. The common characteristic of such programming language is the low entry barrier.

While most of the papers are focussed on the newest and most recently developed programming language as their first choice, there is also a forgotten language that is suitable as the first programming language to teach as well: ActionScript. ActionScript as an old programming language also has a low entry barrier. However, only one paper has proposed it as the first programming language choice[17]. Because of the end of life of adobe flash player, the usage of ActionScript has been greatly reduced. Though Adobe Animate still has ActionScript as one of its programming language choices, Adobe itself has ceased the support of ActionScript. This means no more updated version of ActionScript will be available in the future. Even though ActionScript might be obsolete, it's friendly entry barrier still makes it a good entry level programming language. Thus, to investigate the viability of using ActionScript as the introductory programming language, the following research question has been posed:

1. What are the target groups for different entry level programming languages?
2. What are the advantages and disadvantages of each entry level programming language corresponding to its target group?
3. How can ActionScript find its position among the entry level of programming languages?

This paper focuses on the choices of first programming language for different student groups and proposes the position of ActionScript among those choices. In the next chapter (chapter 2), the scope of the review hopes to clear the boundary of the review. The chapter following will present the methodology behind the review. The fourth chapter will be divided into 5 sections in which each section, except for the first and last section, will review present findings with different topics from the review. Then, a summary of the findings as well as the comparison with existing reviews will be presented. In chapter 5, the characteristics of ActionScript, the current status of ActionScript and the common characteristics of ActionScript with other introductory programming languages will be investigated to present the possible place for ActionScript to be in the list of introductory programming languages. The last chapter will propose recommendations to the programming education sector, limitation of current review and possible future work.

Chapter 2 Scope

Previous systematic literature review that was done in 2018[14] provides a systematic review on the topic of the choice of introductory programming. This review will be an extension of the previous review to expand the process but the topic that was updated by this review will only cover the programming language section of the previous review. The papers that are reviewed were published between 2016 and 2020 inclusive to follow up with the previous review. Papers that are outside the range of 2016 and 2020 inclusive but have been reviewed in the previous systematic literature review will be included as a comparison between current finding and previous finding. Any other published paper that is outside this range and not as part of the previous review will not be included in the data selection process and will not be formally analysed. However, it may be included in discussion where appropriate.

During the planning phase of this review, there were no specific languages that had been selected for analysis. After an initial research on the trending introductory programming languages, a few programming languages have been selected as representative introductory programming languages to be reviewed. It has then been made clear that the focus of the review will only involve using Python, Scratch, Processing and HTML as the introductory programming language. Papers that

introduce ActionScript will also be reviewed to find the common characteristic between ActionScript and other popular introductory programming languages.

To limit the scope of this paper and distinguish the research question, the review will only be conducted on papers that fall into the scope above and focuses on the area of teaching introductory programming language. Other topics such as using introductory programming language to enhance computational thinking will not be considered and analysed. Also, any papers that fall into scope above but focused on education institutions other than primary school, secondary education or college education will also be ruled out to bring more focus on the analysis.

Chapter 3 Methodology

This review that will be conducted is a systematic literature review based on the guidelines proposed by Chitu [18]. The method was labelled as 'standalone literature review' which no primary data is analysed, and the process follows the structure below:

1. Clarify the research questions
2. Conduct searches for the papers
3. Selecting studies from the search
4. Filtering the papers by evaluating their relevancy
5. Data extraction
6. Synthesis result

3.1 Research question

This paper aims to explore the literatures of introductory programming including Scratch, Python, Java and HTML5. Because of the time limit, the paper will only review the literature that is interesting to programming education community. The specific research questions are as follow:

RQ1 What are the target groups for Scratch, Python, Java and HTML5 as the introductory programming language?

RQ2 What are the advantages and disadvantages of the introductory programming languages mentioned above to different target groups?

RQ3 What can ActionScript provide to fit in the group of introductory programming languages?

3.2 Conducting searches

The selecting process for this review proved to be very challenging. In systematic literature review, searching papers is the key process of acquiring data. In this paper, the search will be done through google scholar. It is not an ideal search engine but the reason of using google scholar will be explained in the limitation section. Because of the nature of the google scholar, further quality assurance will also be done to make sure the quality of the paper can meet the requirement. The first step of the search is to define the search term. If the search term is too general, it will often result in a wide range of papers that are not related with the research question. However, if the search term is too specific, a lot of the relevant papers will be missed. To find the accurate and desired search term, the method of trial and error is used. The trial search took place through google scholar with keyword relevance search for each required search term with the range of 2016-2020. Using ‘ “introductory programming language” ”choice” ’ as an example, the initial search term was “introductory programming language” and there are 281 search results from the google scholar where some of them are related with logical thinking and promote computational thinking. To remove the unwanted result, another keyword is needed to narrow the search down. By reviewing the previous search result, it has been noticed that most of the relevant papers have the keyword “choice”. Thus, the search term has been updated with “introductory programming language” and “choice”. By adding the keyword “choice”, it has been narrowed down to 170 papers where most of the papers focus on the properties of the introductory level of programming language which fits with the topic of this paper. However, to avoid losing other related papers, some synonyms have been tested to replace “choice” such as “selection”, “option”. The new search results returned from those keywords are similar to the search result from keyword “choice” but more topics about computational thinking and mindset have also been returned in the search result. Thus, only the keyword “choice” has been selected as part of the search term. The same procedure has

been done for the rest of the search term to make sure the papers that are displayed are most relevant to the purpose of this paper and no other important paper has been missed.

After the trial section, the combination of search terms has been produced for each of the research questions.

Started from the basic of research question 1 which involves the basic of introductory programming language, the following keywords are being used to perform the search:

"low barrier programming language", "introductory programming language choice", "introductory programming course"

For the rest of research question 1 and research question 2, the search term has been divided into 5 parts based on the programming language. The first programming language is Scratch. The search term involves a prefix of "Scratch introductory programming language" following different education groups: "elementary school", "primary school", "secondary school", "college". Same goes to Python where the prefix has been changed to "python introductory programming language" and the suffix remains the same as scratch. Then Java which specifically targeted the processing library. The prefix search term is "processing java introductory programming language" and the suffix remains the same. The last search topic for research question 2 is HTML5, the search terms involve "HTML5 introductory programming language" with different suffixes like Java above.

The search terms for research question 3 are quite different from RQ1 and RQ2, as the question mainly asks about the characteristic of ActionScript, the search term becomes as follow:

"ActionScript", "ActionScript based application", "ActionScript programming language".

The search terms are then being applied to keyword fields of Google Scholar with the setting of range between 2016-2020 and sorted by relevance. The result of the search is as followed (figure 1):

- Scratch primary/elementary school: 28

- Scratch secondary school: 29
- Scratch college:72
- Python primary/elementary school: 18
- Python secondary school: 23
- Python college: 71
- Java primary/elementary school: 23
- Java secondary school: 27
- Java college: 81
- HTML5 primary/elementary school: 4
- HTML5 secondary school: 3
- HTML5 college: 10
- Total:389

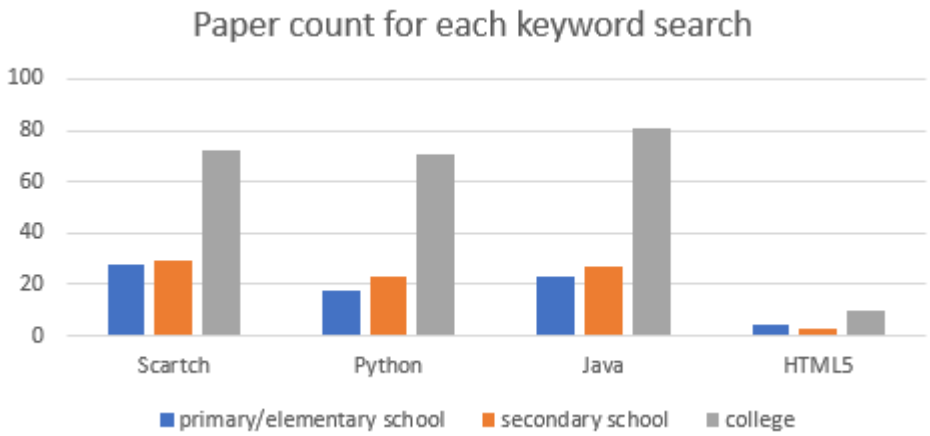

Figure 1 paper count for initial search

As the figure shows (Figure 1), the number of papers that are related with HTML5 is too small to conduct the proper review. Thus, another exploration is needed to expand the topic. It has been noticed that JavaScript is also returned from the search result of

HTML5. Thus, a test of changing the search term from HTML5 to JavaScript has been initiated. During the test, the search result from “introductory programming language” “JavaScript” has returned 80 papers. The initial result seems optimistic. After a closer look at 10 of the most relevant papers, it was clear that the content of these papers is not related to the topic. Thus, this search has been abandoned. Because of the failure of this topic, it has been then decided to remove HTML5 as part of the research question. The new research question 1 is amended to:

“What are the target groups for Scratch, Python and Java as the introductory programming language?”

3.3 Selecting studies from the search

The next stage of this review is to select papers that can be used to form the basis of this review. To select the correct paper, each of the search results is examined with title, abstract and, if necessary, the full paper. The selection process is done by eliminating papers that are irrelevant and making sure the papers that are selected are journal/article or book. Because of the nature of this topic, some of the papers are related to promoting computational thinking or teaching methods which has been ruled out and it is by far the biggest reduction to the number of viable papers.

3.4 Filtering

After the selection process of the papers, a set of potential topics has been created to devise the paper that has been covered. This process starts with a brainstorm of a list of topics that possibly exist within the papers. Then the topic will be merged into three sections: age group, advantages, disadvantages. The age group is the basic group and the advantages as well as disadvantages can be classified as a subgroup under ‘age group’ to classify the advantages and disadvantages of a certain programming language to a certain age group. After categorizing the papers, the focus of the review shifted to a particular topic or group and evaluate the pertinence of the paper to the topic and extract useful data. The process starts from reviewing the title, abstract and keyword in searching for the relevancy to the topic group. This was followed by a brief reading of the conclusion section. During the process, more papers were ruled out because of the content irrelevancy. It is stated that the reviewer should examine the paper more closely

to the quality of the paper to ensure the quality appraisal. Two of the main methods of ensuring the quality of the reviewed papers are "prioritize the papers according to their quality and exclude certain papers deemed not useful due to inferior methodological quality."[18]. It is recommended that a scoring system should be established to measure the quality of the paper and because of the qualitative research, the qualitative quality appraisal should be constructed based on the logical argument of the paper[18]. Because the number of papers available on this topic is limited, the scoring system is not being built but the quality of each paper that is being reviewed has been carefully examined to make sure the quality of this review.

The quality assurance process starts from examining the research question of the paper to verify the logic behind the questions. Only the papers with solid research questions that related with the topic of this paper, that is using Scratch, Python, Java as an introductory programming language, will then proceed to the next stage. Then, the methodology section will be reviewed for each of the papers. For any paper that has a small sample number, for example, the number of participants is less than 10 students, or total number of references is less than 8 or total number of pages is less than 6, it will be ruled out. The number here is based on the Springer's instruction for authors where it requires the author to have a word count of 1500-3000 and that can be converted to at least 6 pages. This is to make sure there will be enough data to analyse while maintaining the quality of the data. After the collection process, the data will then be analysed to arrive at the result for each of the research questions.

And the final number of papers that has been accepted is 36. This turns out to be 9.2% of the initial search returned result. The acceptance rate is unexpectedly low, and part of the reason will be explained in the limitation section. However, a review from 2016 with the similar topic that focus only on Scratch has returned a rate of 14%[19]. A review that compares visual programming with textual programming in 2017 has a filter rate of 18.75%[20]. Another related literature review from 2018 has the acceptance rate of 43% with a broader topic[14]. Though the acceptance rate in this study is less than the reviews above, with the limitation added, it can be considered as a normal search result. Therefore, the review continues with this number of papers.

3.5 Data Extraction

At this stage, the data extraction will be performed from the filtered papers. Due to the nature of this review, the data extracted from the paper will be qualitative content such as characteristics, advantages rather than quantitative content such as the number or percentage of a certain group or usage. Based on the note form of the data extracted from the filter section, a formal version of the data will be created and documented for further analysis.

Chapter 4 Literature review of Scratch, Python and Java

In this chapter, a systematic review will be performed that synthesises findings of studies about the target group, advantages and disadvantages on Scratch, Python, Java and the characteristics of ActionScript. The review starts with an overview of the current situation with introductory programming language to provide a general background. Then, each of the introductory programming languages in the research question will be introduced. The following section will list out the findings from the literature based on the different topic group that summarized during the process. Lastly, a summary of the findings will be presented to lay the foundation for the next chapter.

4.1 Overview of introductory programming language

A programming language is a system that allows a person to interact with machines and be able to be "understood" by both of them. There are many levels in terms of programming language such as "Entry level", "Intermediate level" etc. The English dictionary defines "Entry level" as "Suitable for a beginner or first-time user; Basic". APA defines that an introductory level of programming course should be designed for learners who may have little to no background in programming skill. This level of programming course should get the learner familiar with theoretical underpinnings, principles, methods, and perspectives of programming language. The content within the course should be focused on breadth, enrichment and or general knowledge[21].

There are a variety of factors that affect the novice programmer to learn programming. For example, student motivation[14][22][23]. The study indicates that lack of motivation is one of the main factors that causes the student to drop-off from introductory programming courses[24]. It is also stated that students get frustrated by

not being able to find the error of their code, payoff is imbalanced which results in motivation dropped[24].

To solve these issues, the root of the issue must be identified. In the previous review, serval topics has been identified in relation to teaching introductory programming language. Example includes theory of learning, teaching delivery, tools and programming languages. Though the analysis is thorough, most of the findings are regarding teaching method[25] and delivery. In terms of the characteristic of programming language itself, though there aren't many findings on it but it is still considered to be an important topic[14].

In order to better facilitate the entry level programming course, the choice of the first programming language based on the characteristic of the programming language for an introductory programming course is important. There is no such thing as 'perfect' inductor programming language but the choice of it has been argued through many papers by comparing different languages as the first programming language. Examples include comparison of Scratch with Python[26] Python with Java[27], C++ with Java[28] and C++ with Python[29]. The following subsection will provide general introduction regarding Scratch, Java and Python.

4.1.1 Scratch

Scratch is a block-based visual programming language. Visual programming language refers to the syntax of the programming language including visual expression such as diagram, sketch or icons[30]. Students can drag blocks and snap them into scripts to run the program[31]. It is free and available in nearly 50 languages such as English, Chinese etc. In addition to that, Scratch is designed to introduce programming to those without any previous experience. Because of this goal, Scratch uses visual blocks language, single-window user interface layout and minimal command set[32].

In this study, the published papers that relate with scratch within the review scope including any comparative paper that compares scratch with other programming languages and demonstrates the benefits or drawbacks of scratch will be explored. Figure 2 gives an overview of the number of papers that have been selected for this section ordered by the year of publication with a total of 14 papers.

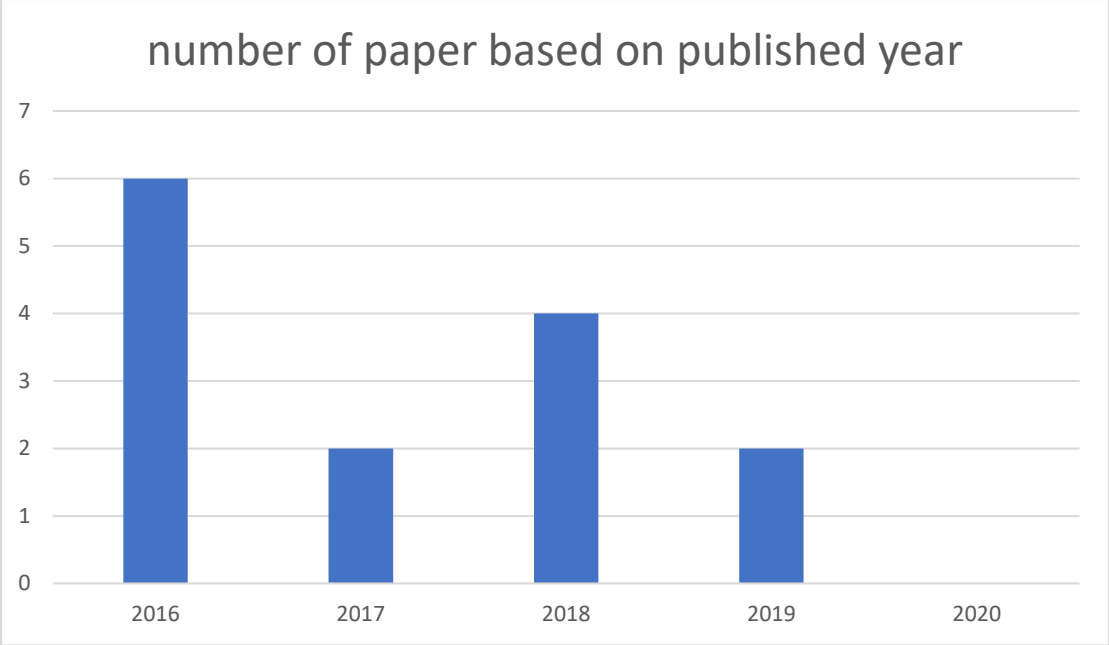

Figure 2 Distribution of 14 papers related with Scratch based on published year

During the review section, it has been found that 2 papers[33] [26] have topic that related with input method and emphasising the benefit of having block-based programming language such as Scratch. 9 papers mentioned or focused on the syntax of scratch which promote the syntax error free of Scratch. Lastly, almost all of the papers are related to the acceptance within the student group ranging from primary to college level of education

4.1.2 Java

Another popular introductory programming language is Java, an object-oriented text-based programming language. It was developed around 1995 and it is an C based programming language which makes it one of the traditional programming languages and has been stated as one of the most popular introductory programming languages[34]. And it is also the most commonly used programming language in the world[20]. Because of its popularity, most of the college would use Java as the introductory programming language to freshman students. The papers selected to review for the Java section includes all papers that conduct experiments using Java as an introductory programming language or comparing Java with other programming languages to identify the characteristics of it. The number of the papers that qualified are shown in figure 3.

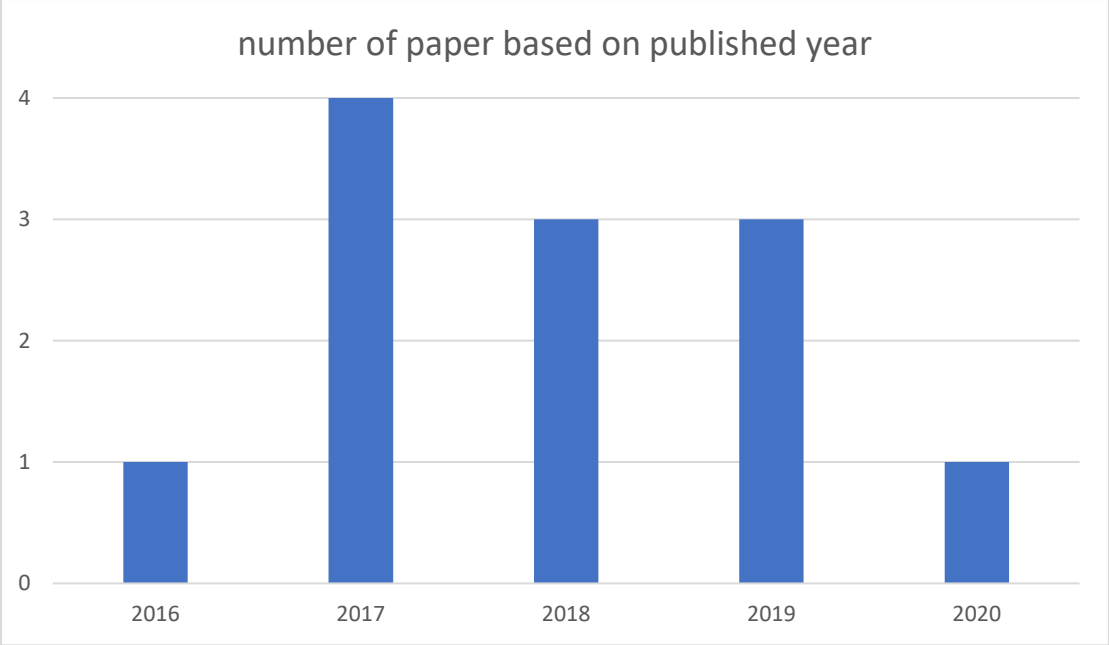

Figure 3 Distribution of 12 papers related with Java based on published year

The actual number of papers that can be used in the Java section is lower than expected number. But it is still sufficient to conduct the research. There is a total of 3 papers that have a target experiment group and 7 of the papers discussed the syntax difficulties and 3 papers discussed the perceptions of students studying Java as an introductory programming language.

4.1.3 Python

Python is another object-oriented text-based programming language. It is considered as one of the most popular first-to-learn programming languages with almost the same popularity as Java[29]. Though Python is older than Java, it is actually simpler than Java in terms of syntax and grammar. The syntax within Python is nearly English which made it easier for students to study[35]. Python also been described as “a good programming language for initial learning” because of the simplicity of the syntax as well as the popularities in the programming education area[36]. The number of schools that use Python as introductory programming language also increases for the past few years[37]. In Figure 4, the number of papers that lies within the scope of this review after the filtering process is being illustrated.

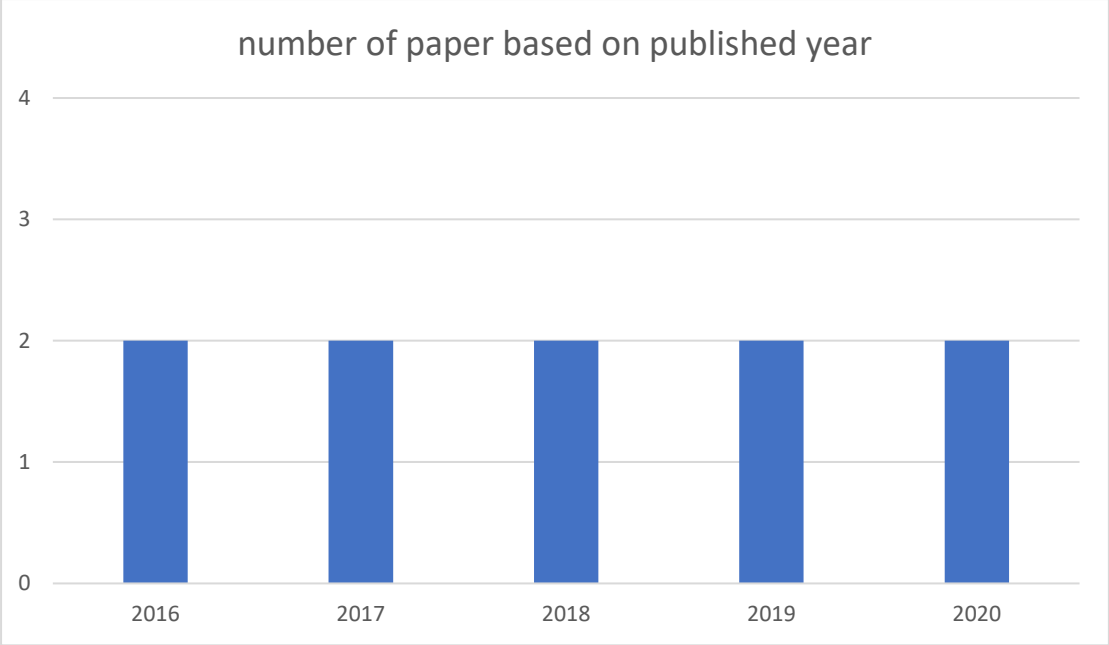

Figure 4 Distribution of 10 papers related with Python based on published year

The result is as unexpected as Java, there are less than expected number of papers that related with the properties of Python programming languages. Of the 10 papers, 3 of them have controlled group experiments with either teaching Python alone to students or comparing teaching Python with teaching other programming languages. Another 3 papers within the 10 papers focus on the syntax difference including the comparison between syntax of Python with Java. 8 out of 10 papers have either included the recommended age group for teaching python as introductory programming languages or included the feedback from target experiment group about the perception of studying Python as their introductory programming language.

4.1.4 Summary

Throughout the 36 papers, it has been noticed that the most noticeable topics among the reviewed papers are input method, syntax, semantics and acceptance. Based on this finding, the following three topics have been created: input method, syntax and semantics and acceptance. In the following section, data for each language within the research question are being detailly analysed based on the topic mentioned above.

4.2 Input method

This section focuses on the different input methods between Scratch, Python and Java. The input method in programming language is defined as how the user can interact with the programming language to create a program. There are many input methods for different programming languages such as text, visual and vocal[38].

However, in most introductory programming languages, the two preferred types of input methods are text and block. As introduced in the introduction chapter, block-based input method refers to allowing students to drag blocks to form the program. Text-based programming language on the other hand, requires students to type the actual syntax to construct the program. In another recent review which focused on the input method of different types of the programming language, a summary of 4 comparison studies has been given and suggested that visual based programming language can provide a better result for the novice programmer to understand programming concepts[20].

Another study states that one of the benefits of using Scratch as an introductory programming language is that the difficult concepts can be easily understood by novice programmers because Scratch has hidden the text code behind the blocks[39][40]. Using variables as an example, it takes the student a click of a button to create variables. In the process, the users get to select the scope of the variable: global or local. There is no such thing as naming conventions for Scratch as the students are allowed to give any name to any variables. The operations for variables in Scratch are also pre-defined which means the students can simply drag the block with set, change, show, hide variables to achieve such operation. It gives the student more sense of how to and what can be done to manipulate such variables.

An experiment that includes teaching primary school students with Scratch as an introductory programming language indicates that the process of creating their first programming project with a visual programming language such as Scratch is proven to be fun. Students are highly motivated by the block input method which improves the student's commitment towards the project. The overall result of the experiment indicates the success of block-based programming language and also demonstrates that with block based programming language, students can better understand the concept such as loop[33].

Another experiment uses Scratch as an introductory programming language as well to deliver the introductory programming course to primary school students in 2017. The participants are 5th grade primary school students without prior programming experience. The content of the experiment includes using Scratch as the introductory programming language to deliver introductory programming courses to the participants. Findings suggest that students show an increasing positive attitude towards programming courses because of the visual programming method and have concluded that block-based programming language would greatly increase the motivation for novice programmers[26].

During a comparative study, the two control groups are set to explore the result of using block-based or text-based programming language as the first programming language. The findings show that students who start with visual programming(Scratch) are proven to have higher level confidence in their understanding of programming techniques than those who start with text-based programming language[41].

Text based programming language on the other hand, will require students to manually type the syntax in the editor to form the program. In a paper that compares the input method for programming languages, it is stated that the text is a form of representative of programming which cause more difficulties for the student [42]

Another review also compares block-based programming language with text-based programming language by conducting a class experiment with 312 students ranging from 5th grade to 6th grade. Each group of students is being taught by experienced CS teachers with either Scratch as block-based programming language or Python/Logo as text-based programming language and student's performance is evaluated through a series of tasks. The findings indicate that at the beginning of a simple task, both groups of the students performed at the same level. As the difficulties of task grow, the requirement for student to understand the programming concept and rule increases and the group with Scratch is demonstrating a significantly better performance than those with Python.[43]

Though there is literature that investigates the effect of input methods towards the novice programmer, most of the related papers are proposing a visualization tool/method of the text-based programming language to enhance student's performance

while maintaining the benefit of transitioning to text-based programming language. Examples include ViLLE, which runs on a subset of Python, and allows for visualisation[20], Alice which is a Java based visual programming language and tool that can convert the project back to plain Java code. A frame-based editor that combines the block-based programming language with text-based programming language[44] and a means that combines Python with visualization tools to demonstrate the programming process[45]. There are few papers that focus on the effect of input methods towards the choice of introductory programming language. However, the topic has been widely discussed informally as stated in the paper[42] but in terms of academic study, there is still a gap to fill.

4.3 Syntax and Semantics

The most mentioned factor in the input method section is the syntax. It is clearly another important topic that needs to be addressed for introductory programming language based on the analysis of the difficulties of learning programming[46]. Syntax in programming language refers to rules of the combination of symbols to form a program[47]. It exists in any programming language no matter the input type. In text-based programming language, the syntax refers to the combination of symbols and keywords or the variable name in text-based programming language whereas in block-based programming language, it refers to the block that has been created for the programmer to drag. One of the basic errors that novice programming could have is the syntax error which occurs by violating the rule of the symbol combination which causes the program to fail. It is often described as syntax noise that would negatively affect a student's learning experience.

In block-based programming languages, syntax has been pre-defined which makes it nearly impossible to have a syntax error. The study from 2017 also states that Scratch offers a syntax free programming experience to the novice student. The syntax-free programming language will enable students to focus more on problem-solving and design algorithms rather than memorising syntax which increases their learning efficiency[26].

However, in text-based programming languages, syntax errors are a common mistake. It is stated that the novice programmers often face syntax barriers; they face a general-

purpose programming language such as Python and Java. Compared with block based programming, it is found that even the Python syntax is not considered to be simple to a novice programmer and the syntax difficulties vary based on the programming language they faced[41]. The syntax within Python has been described as "elegant, simple and practical" and easy to read[20] but Java is more complex compared to Python. For example, in Python, the keyword 'def' can make it unambiguously clear to the students that they defined a function rather than variable whereas in Java, the keywords for different variables and functions such as 'int', 'String' would confuse students. Another example includes the scope of the variable. The attributes in Python are all public and all of the exceptions are prefixed with one or two underscores so that the learner can understand it is not a public variable whereas in Java, students need to specify the scope of the variable[48]. It has also been stated in the previous literature review that in order for novice students to master the Java programming language, one must master the syntax before they can move to master the problem solving skill[39].

A study comparing the syntax difference between Python and Java suggested that Java as a C type programming language has a traditional syntax rule. The syntax approach of Java is described as 'imperative' and requires additional mental focus on complex issues to fully understand the syntax rule within. Only then can the student fully understand the mechanism of syntax in Java. Python, on the other hand, is a recently developed programming language. Though it is another object-oriented text-based programming language, the syntax rules are quite different from Java. For example, brace-based syntax has been applied to Java which is similar to C or C++. This means if the student couldn't have the curly brace correctly to define a conditional statement, loop, function or class, it would cause a syntax error. While in Python, the functions are being defined with code blocks. Thus, the paper conclude that as Python uses a more declarative approach, it is easier for the students to understand the programming concepts without suffering from syntax noise and also help the students to avoid syntax error compare with Java[48].

A comparative analysis in 2020 also compares the difference between Java and Python for novice programmers. The study conducted through exploring the syntax difference without any experiment that requires the participants. Comparison includes the topic of syntax and implementation where a major amount of effort is spent towards the syntax

area. It analysed the syntax between Java and Python including variables, castings, comments, print, separators, structure style, operators, strings, input, loops, conditional statement, classes, objects, scope, file handling and exception handling. The finding shows novice programmers can easily read Python programs because of the simplicity of its syntax while Java programs could cause major compilation errors for them because of the complexity of Java syntax[35].

Another paper from the previous review compares Java with Python in the aspect of syntax as well. The result of that paper also indicates that python can reduce the syntax error because of the simplicity of its syntax compared with Java. It is described in the experiments that students would normally have syntax errors of 'missing semicolons', 'missing brackets' whereas in Python, these errors won't exist[49].

Findings by another recent study revealed that teaching Python which is a syntactically simple language as the first programming language can enhance student's understanding towards the programming concepts when comparing it with the syntactically complex language such as Java. As a result of such simplicity, the teaching priority can then shift from programming concepts to problem-solving skills which greatly enhance students' performance[50].

Apart from syntax, the semantics is another major concern. Semantics in programming language describes the execution and result of the program from the input code. The issue of semantics exists in both block-based programming language and text-based programming language. It is an issue with the students' problem-solving skills rather than the understanding of the programming language. Many papers have mentioned the issue of semantic error in the introduction or background section[26][51][52] but none of the reviewed papers has indicated any direct impact of semantic error towards the choice of first programming language. However, in the comparison of Java and Python, it is stated that the student would feel the reference semantics of Java are "illogical"[49]. This perception from students indicate that the semantics error does affect student's attitudes towards the programming language. Thus, more research is needed to this topic to deeply explore the effect of semantic error towards the choice of first programming language.

4.4 Acceptance

Besides the characteristic of the programming language itself, the acceptance of the programming language is also an important factor to consider. This section will present the review on Scratch, Python and Java in terms of the age group and some student's perception regarding it. There are three different age groups that have been identified: primary school, high school and college. The following table (table 1) shows an overview of the paper regarding the corresponding group.

Paper	Age/Student group	Programming Language
[19]	Primary school student	Scratch
[26]	Primary school student	Scratch
[33]	Primary school student	Scratch
[41]	College student	Scratch, Java
[48]	College student	Python, Java
[51]	Primary school student	Scratch
[52]	Primary school student	Scratch, Python
[53]	Primary school student	Scratch
[54]	Primary school student	Scratch
[55]	College student	Scratch, Python
[56]	Primary school student	Scratch
[57]	College student	Scratch

Table 1 Overview of papers that conduct experiments

As shown in the table, over the 11 papers that perform target group experiments, 10 of them are investigating Scratch as an introductory programming language either a sole scratch programming course or a control group comparison between scratch and other

programming languages. Only three papers have the experiments on Java and three papers conduct experiments using Python. This demonstrates the difference between block-based programming language and text-based programming language.

It is obvious from the table that most of the target group from Scratch are primary school students. In one of the experiments, it has concluded that compared with traditional programming language, Scratch may be a better approach to teach introductory programming language because it can significantly increase student's motivation and increase the content absorption[19]. A study that conducted an experiment with primary school children also finds "at least every second child in grade 4 is able to learn basic programming with Scratch in three days"[53]. Another experiment concludes that Scratch is so simple that primary school students without prior programming knowledge can also easily capture the basic programming concepts such as variables and loops[56]. According to the control group experiment, Scratch can also reduce the impact of low problem solving abilities and reduce the give-up rate[52]. It has also been suggested that students learning Scratch have better understanding of logic and a stronger problem-solving skill[51]. Another research concludes that Scratch is designed for children aged 8 and above with the desire of learning basic programming skills[36].

Apart from primary school, a recent review also suggests that the high school students also express a satisfactory attitude towards the use of Scratch as an introductory programming language[12]. The same paper also suggests that applying Scratch to students from 6th grade classes is a better option for the student's learning curve because of the level of cognitive maturation. Even for the college level education, the research also shows that good experience is identified for college students to use Scratch as an introductory programming language because the students without prior programming experience can also understand the programming process[55].

However, the paper also indicates that Scratch does not provide enough assistance towards the student in preparing them to advance to higher-level programming languages such as Java[51]. As described in another paper, it has been deliberately avoided to overload the Scratch with anything too complex that might make a child find it threatening. The lack of procedures also leads to the failure to convey the core idea of computer science (CS): recursion to the students. There is also no such concept as class or object which the paper then concludes Scratch is not useful to teach object-

oriented programming language which is so called “real” programming language. Thus, it is not considered to be a desirable introductory programming language for college students studying CS[39]. It has been suggested that Scratch should be a tool or a platform that serves the purpose of teaching simple programming concepts for the students[26].

Furthermore, in an experiment conducted with 93 college students using Scratch as introductory programming language, they expressed negative attitudes towards the usage of Scratch as the introductory programming language. Typical commons include students referring Scratch as “childish” and “non-serious” programming language. The paper then concludes the informal introduction of Scratch in college course would fail to motivate student from studying it[57].

Python has a high acceptance rate in high school. The study reflected that python is a common choice for an introductory programming language for high school students who desire to get ahead of others in study programming[48]. It is also stated that the acceptance of python as the first programming language is affected by the factor of ‘ease of learning’. Students who desire to get an easy entrance to the ‘real’ programming language would normally choose Python as the first programming language[34].

However, when it comes to primary school education, the nature of text-based programming language results in less acceptance than block-based programming such as Scratch. Research also finds that students get better performance when taking python as an introductory programming language if the student possesses higher problem-solving abilities[52]. This means the primary school students are normally not mature enough to have the required skill to perform well in a Python introductory programming course. Beside primary school education, it is also stated that students with Python as an introductory programming language often have difficulties transitioning to Java. This is mainly because of the nature of ‘object first’ in Java and the lack of the concept of array in Python[20].

The acceptance for Java is mainly at the college level as described in the table, the only three experiments that involve Java have the target group of college students. This is mainly because of the complexity of the structure and syntax of Java as discussed in

section 4.3. It is considered as a difficult programming language to learn for novice programmers[41]. According to a survey from college students, Java is the most popular with 41.94% of respondents picking it as their teaching language[34]. Another study that research the acceptance rate for different programming language as introductory programming language in UK and Australasia. Finding shows that Java hits the highest selection rate among other 21 programming languages[58].

The factor that affects the acceptance of the programming language for different group are varied based on the review above.

Students at the stage of primary school often lack the problem-solving ability and cognitive levels are low as well. Thus, more focus has been paid towards the simplicity of the programming language such as syntax free. Moreover, the teaching goal is more towards the area of introducing basic programming concept. Thus, languages that Scratch is the most considered programming language as the first programming language to learn.

As the students' age grow, the problem-solving ability increases, and it would be possible for students to handle other issues such as syntax that they may encounter during the learning process. The aim of the introductory programming language is shifted from introducing basic programming concept to preparing students for higher level programming[58]. The review indicates that block-based programming such as Scratch cannot help the student to advance to a higher level. To satisfy the learning goal and learning ability, text-based programming language starts to get more popular. However, the student conation is still at a limited level which means complex programming language would still be too difficult to teach as introductory programming language. Thus, the acceptance for Python lies in the high school education and college education[36].

When the students reach college level, the maturity of conation allows the student to process more complex logic and understand more difficult rules and grammar. In this case, the goal of introductory programming language would be more towards the industry area which prepare students to get their hands on some difficult programming project. The teaching focus has then shifted again from basic programming concept, foundation to industry programming and enhance problem solving skill. Java as

compared with Python has a complex syntax but has a more general implementation. It is also suggested that even though java is difficult to learn, with good knowledge it is possible to apply it for development of different types of applications[36]. This leads to the popularities of Java among colleges.

4.5 Summary

The previous section has analysed in detail the input method, syntax & semantics and the acceptance between Scratch, Python and Java. Based on the review above, the answer for the research question 1 and research question 2 can be concluded.

4.5.1 RQ1 What are the target groups for Scratch, Python and Java as the introductory programming language?

The review on acceptance has already gives some basic conclusion towards the research question. To summarise the findings from above section in relation to this research question, the following conclusion has been made.

The target group for Scratch are more for primary school education than high school or college education[26][33][53][54][51][19][56]. Though the usage of Scratch cannot bring student to a higher level of programming, it gives the student a basic knowledge of programming concepts such as loop and conditional statement and the block-based nature makes it better for student in primary school to understand those concepts.

Python is more suitable for high school education than primary school because of the different cognitive level among students or the level of understanding difficult content for students[20]. As for college level, it is still argued by many papers of its usage as an introductory programming language in college education as shown by the multiple comparative papers[34]. The review from the papers suggests that Python is suitable as introductory programming language to college student because it allows the students to get a better understanding of the programming concept, thus, fulfil the purpose of the introductory programming language: introduce programming concept to students and a better performance can be achieved through the syntactically simple programming language such as Python. But drawbacks have also be pointed out- such as the simplicity of Python cannot satisfy the needs to transitioning to a complex

programming language such as Java. Therefore, it is still under debate whether Python should be the choice of introductory programming language in college education[29].

Java, however, has been mostly mentioned in college education, specifically for CS students or students that take CS1 courses. The implementation and complexity of Java makes it a hard programming language for beginners and requires enough cognitive level to understand the syntax and concepts within Java. Thus, it is not suitable for primary school student because of the complex rules and programming concepts presented by Java to them. The findings from the review points out that Java is the most popular introductory programming language in college education. Because of the popularities, many comparisons between Java with other language has been made to point out a better introductory programming language other than Java. Though many papers have promoted other language to replace the place of Java in college education[36], it is still widely accepted by college to use Java as introductory programming language.

4.5.2 RQ2 What are the advantages and disadvantages of the introductory programming languages mentioned above to different target group?

In the above sections, the three introductory programming languages in terms of input method, syntax and semantics have been examined in detail and the findings in relation to research question 2 can be concluded as follows:

The advantages for Scratch are mainly based on the input method[20] [26] [33] [39]. The block-based input method allows the students who study Scratch to be free from syntax error. This is particularly beneficial to primary school student. The review suggests that without syntax error, student have a higher motivation and better performance. However, semantics error is still unavoidable but because this is the only difficulty that student will face, it can be managed. As for the disadvantages, this study shows that the nature of block-based programming language has limited the concepts that can be introduced to student. For example, as mentioned in the review, teaching with Scratch cannot introduce the concept of recursion and object to the students. These seems to be acceptable in the primary education as the purpose of introductory programming language is only to introduce the programming concept, but it is not acceptable in college level education especially for students who study computer

science. Another finding from the review indicates Scratch makes no effort to prepare students for higher level of programming.

When facing primary school students, the text-based nature makes python not as competitive as Scratch for students. But the advantages of providing student the foundation towards higher level programming language makes it lies within the choice of being one of the introductory programming languages in primary school education. As stated in the conclusion for research question 1, python is more popular in high school education and college education. The findings from reviewing multiple comparison papers indicates such. It is stated in the literature that Python has the advantages of being syntactically simple compared with other programming language such as Java[48]. The dynamic typed nature of the programming language makes it easier for the students to define variable. These become the main reasons why it is popular in college level as well as the high school level. It makes the Python a good choice to introduce a low barrier entrance for students while maintain the fundamental programming concepts and forge the foundation towards other high-level programming language. However, the disadvantages are also caused by one of its advantages: dynamic typed. The nature of dynamic typed makes the student ignore the issue with variable type that would cause significant error when the students move to other high-level programming language.

Java on the other hand, undoubtably has the disadvantage of being syntactically difficult for primary school students. As stated in the acceptance section, the cognitive level for primary school students is normally not enough to understand the mechanism of Java which result in the conclusion that Java is not suitable for introductory programming language in primary school level education. The popularities of Java stay within the college as shown in the acceptance section. The advantages of Java are also the disadvantages of Java. Because of the complexity of Java such as strongly typed, strict expectations, the students would normally be able to get the hangs of the fundamental programming concepts within the high-level language. One of the reasons that college level education chose Java over Python is the implementation. As Python is only getting popular in recent years, Java is already in every part of the software we use. It has also been pointed in the study that Java compare with C, C++ or C#, has easier way to write,

compile and debug. Thus, these advantages make Java an idea introductory programming choice in college level.

Chapter 5 ActionScript

ActionScript is an object-oriented script programming language that is used to create content for website animation, flash player. There are three generations of ActionScript: 1.0, 2.0 and 3.0. The latest version is ActionScript 3.0 which was released in 2006. The production of ActionScript at the beginning was flash player (.swf) file only. With the release of Adobe Animate, it now also produces the flash animation (.fla) file. This chapter focus on the introduction of ActionScript from early literatures and the characteristic of the ActionScript. The first section introduces the properties of ActionScript. Then the following section will discuss the issue of adobe flash player end of life. Lastly, ActionScript will be analysed based on three topics that are summarized from chapter 4 in the second section to summarize a conclusion that answers the research question 3.

5.1 Properties

ActionScript as described above is an object-oriented programming language. In the handbook from Adobe, it is stated that ActionScript 3.0 is designed for "facilitate the creation of highly complex applications with large data sets and object-oriented, reusable code bases"[59].

The input method for ActionScript is by default a text input. However, ActionScript is merely a script for the action of the object. The object creation process can be done through visual method such as drawing or dragging default image. This can save the students from struggling with creating object or variables using plain code and provide a better programming experience.

A study in 2006 promotes ActionScript as a "gentle" introductory programming language. In the paper, it is said that the syntax and semantic is similar with other C like programming language such as Java. But, unlike other object-oriented programming languages such as C++, Java, ActionScript provides instant visualized

results for the learner with less scaffolding and frustration. This will not only help students over the initial barriers of a first course in programming, but also enhance the engagement. The paper also investigates the code block based on the example. It has shown that for implementing I/O and array processing, ActionScript has 14 less code statement than Java[17]. The comparison result is similar with the recent study of comparing Python with Java[35]. The conclusion demonstrate that ActionScript can be a potential introductory programming language to provide clear and concise programming exercise. It can also facilitate further transition to programming language like Java or C++.

Another study in 2006 also investigates the usage of ActionScript as the introductory programming language. The targeted participants were 30 undergraduate students with little or no programming experience. Experiment divided the participants into three group where first group was using ActionScript as the introductory programming language and other two groups were using C++ as the introductory programming language. The findings suggested that ActionScript is not recommended as introductory programming language. Issues that identified are as followed: The ActionScript is not specifically designed for novice programmers or for the purpose of education. The biggest drawback identified in the research is that declaration of variables is not required in ActionScript which results in difficulty of detecting typographic error. When it comes to progressing students to high-level programming language such as C++, it does not build solid foundation for students to understand the fundamental programming concept[60].

The acceptance of flash is relatively high as many papers has proposed the flash based application for education purpose[61][62]. However, the distinction must be made in relation to the difference between the acceptance of ActionScript and the acceptance of Flash. The flash-based program will not be counted as the acceptance of ActionScript as it is an application that is produced by ActionScript but no fundamental of ActionScript is presented to student. This is similar to the relation of Scratch 3.0 and JavaScript. Though Scratch3.0 is written in JavaScript, the acceptance for Scratch is not equivalent to the acceptance of JavaScript.

A research in 2014 investigates the usage of ActionScript as the introductory programming language to non-computing background students. 62 postgraduate

student participates the experiments. Students express the ActionScript is interesting and can attract the students' attention. The motivation of studying programming language is also increased from studying ActionScript[63].

Other than the factor of students' perception and acceptance of ActionScript, the recent end of life for adobe flash player is another key impact towards the acceptance of ActionScript.

5.2 Impact of Flash EOL

The original file that ActionScript produced was a flash player (.swf) file which is an animation that can be played using a website plugin. However, as adobe announced in July 2017, adobe flash player will no longer be supported after 31 December 2020 and all the existing flash player content will be blocked from 12 January 2021. This leads to massive destruction to the website that's dependent on flash content. The reason being the complete deprecation of .swf file which will no longer be available on any browser or even local copy. Google reports that flash usage declines from 80% in 2014 to under 8% in 2018[64]. However, in China, most of the education websites are still using flash as the basic animation content for students. Because of that, flash is still supported in China[65].

The main reason of the end of life is because of the security issue with adobe flash player. The review in 2015 detailly listed the trends and threats of ActionScript web security. Findings suggest the vast range of security issues exists because of the flexibility and power of the Adobe Flash platform[66]. To cope with this situation, several solutions has been investigated.

A study in 2017 investigate the method of merging flash files to HTML5/JavaScript. The study compares the code difference between ActionScript 3.0 and JavaScript. It has been found that in order to merge the flash file, it has to be rewritten in JavaScript[67]. The result of the study indicates that it is difficult for the current flash application to be transformed into HTML5/JavaScript.

It can be seen from the papers that though efforts have been made to convert the flash-based application to HTML5/JavaScript, it is still a difficult process which result in the usage of ActionScript being greatly reduced. As the option of HTML5 canvas appears

in Adobe Animate, more people are choosing to use HTML5 canvas rather than ActionScript as the programming language to use in Adobe Animate. Thus, the usage of ActionScript is relatively low.

5.3 Summary

5.3.1 RQ3 What can ActionScript provide to fit in the group of introductory programming languages?

Owing to the limited number of papers available for the study on using ActionScript as introductory programming language, it is thus hard to conclude if ActionScript can be considered as introductory programming language based on the syntax and semantic. When evaluating the factor of input method, none of the literatures has compared ActionScript with block-based programming language but the nature of text-based programming language makes ActionScript not suitable for the primary school students. Though the instant visual result from ActionScript has been proven to be beneficial to student's motivation towards learning programming, the complexity of ActionScript makes it difficult to use as an introductory programming language in high school. Because of the end of life, the industry demand of flash has been greatly decreased which result in the decline in the usage of ActionScript as well. Thus, it is not to be considered as a good introductory programming language compare with other available programming language.

Chapter 6 Conclusion

This study extended the previous literature review related to introductory programming language. Specifically, this study investigated how the characteristic of the programming language will affect the choice of introductory programming language by instructors and students. The findings from recent literature shows similar result as the previous reviews. The role of Scratch and Java remains the same. Scratch is still the most popular choice among primary education because of the block-based input method. Java, because of its complexity of syntax, semantics and industry usage, is still the choice for college education. But a minor difference has been found in this review. The

usage of Python as introductory programming language is gradually increasing. The target level ranging from high school to college and it is challenging the dominate position of Java. This may have been caused by the growing industry demand for Python combine with the simplicity of its syntax. This review has set to find if ActionScript has a position among other introductory programming language and the result turns out to be an unfortunate no. Because of the input method, it is not to be considered as an introductory programming language to primary education and because of the complexity of the syntax, it is also not recommended for high school students. The end of life also reduced the industry usage of ActionScript which makes it nearly impossible to compete with other introductory programming language. However, one thing is sure, the instant visual result from ActionScript has definitely made the programming easier for the students to comprehend. Thus, it could be possible to amalgamate the feature of the ActionScript into other popular introductory programming language to further reduce the entry barrier for novice programmers.

Chapter 7 Limitation and future work

Many limitations can be identified throughout this research. The limitation of this research is quite serious. During the initial searching phase, the search result returned by Google scholar was satisfying and the trial of the search term turns out to be successful. However, as the filter process goes on, a serious issue rises. Because of limited access, over 31.6% of the papers are not available to download or view. Thus, only the title and abstract are available. Even though the abstract of the paper might seem to be useful, the lack of deeper analysis of the content within forced the review to rule out from the analysis phase. Another issue is the search result from Google Scholar. At the beginning of this review, a systematic search from credible website such as Springer, IEEE, ACM was considered. But the time limit as well as the access limitation forces the search to be conducted on google scholar. And as stated in the methodology section, the quality of the papers that returned from Google Scholar is not ideal and requires massive amount of validation. Besides the quality, the accuracy of Google Scholar is another obstacle. Google Scholar does not support the keyword search. This results in a massive amount of false positive from the search result. Even though the

search term has been tested to make sure the search result is narrowed down, there are still huge number of papers that only have the search term in a sentence or in abstract. This greatly reduced the search efficiency. Thus, the conclusion from this paper to the three research questions is though credible from the existing literature available, it still needs further validation. Besides the limitation of the search method, several new search terms have been found that can be used for future research. During the analysis process, related keywords are being found in the literatures and because of the time constrain, it is not possible for this review to apply new search term and perform more literature review. Thus, in relation to the topics that summarised in chapter 4, search terms be added on top of the original search term. For 'input method', "block-based programming" or "text-based programming" can be added to complete the related search. A search term "syntax" can be added to the search term to get more literatures that relates to the topic of 'syntax'. As for the 'acceptance', there are no new search term has been found during the process. The search terms suggested above has been partially tested and the result returned are promising. But further validation is still required.

Furthermore, the limitation has restricted the review and a research gap has been noted. The two of the previous literature reviews to this related topic could only return approximately 10%-15% of which only 9% is useful paper for this review. It is stated in the literatures that there are few literature reviews that explore the characteristic of introductory programming language and its impact to the choice of first programming language. From this study, only three of the related literatures has been found for the past 4 years and even so the topic which the paper focused on only being mentioned in a section and more emphasis are put on the teaching method and student's perception. It has also been found during the review section in this study that more papers have been published in relation to the topic mentioned above, but they are yet to be systematically analyzed. It is important for the teacher or the course organizer to know the difference between different programming language and their characteristics so that a proper choice of first programming language can be used for different student group to raise the success rate in programming course. Therefore, more systematic analysis is needed to explore the mechanism and nature of the programming language in order to choose the appropriate introductory programming language for students at different education level.

Moreover, another factor that affect the choice of introductory programming language has been identified as well. It has been noted that the factors that determine the target group can be divided into two sections. This study has explored one of the sections which is related with the programming language itself. However, it is also related to the purpose of the introductory programming language. The different goal of the introductory programming language often results in different application of different programming language as introductory programming language for different education level. Thus, further research is needed to explore the general purpose of introductory programming at different education level.

Bibliography

[1] J.-M. Hoc, *Psychology of programming*. Academic Press, 2014.

[2] L. E. Winslow, "Programming Pedagogy—a Psychological Overview," *SIGCSE Bull.*, vol. 28, no. 3, pp. 17–22, 1996.

[3] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Int. J. Phytoremediation*, vol. 21, no. 1, pp. 137–172, 2003.

[4] K. Kanaki and M. Kalogiannakis, "Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses," *Educ. Inf. Technol.*, vol. 23, no. 6, pp. 2673–2698, 2018.

[5] A. Merkouris, K. Chorianopoulos, and A. Kameas, "Teaching programming in secondary education through embodied computing platforms: Robotics and wearables," *ACM Trans. Comput. Educ.*, vol. 17, no. 2, 2017.

[6] P. Brusilovsky and others, "Teaching Programming to Novices: A Review of Approaches and Tools.," in *Proceedings of ED-MEDIA 94--World Conference on Educational Multimedia and Hypermedia (Vancouver, British Columbia, Canada, June 25-30, 1994)*, 1994.

[7] N. Van Es and J. Jeuring, "Designing and comparing two scratch-based teaching approaches for students aged 10-12 years," *ACM Int. Conf. Proceeding Ser.*, no. November, pp. 178–182, 2017.

[8] Y. Prokop, E. Trofimenko, N. Severin, and L. Bukata, "An Analysis of Criteria for Choosing a First Programming Language in Universities.," in *ICTERI*, 2019, pp. 420–425.

[9] G. Nel and L. Nel, "Motivational value of code.org's code studio tutorials in an undergraduate programming course," *Commun. Comput. Inf. Sci.*, vol. 963, pp. 173–188, 2019.

[10] P. Wegner, "Concepts and Paradigms of Object-Oriented Programming," *ACM*

OOPS Messenger, vol. 1, no. 1, pp. 7–87, 1990.

[11] M. Pinto and A. J. Osório, "Scratchjr App in Portuguese Schools: Kids Media Lab Project," *ICERI2020 Proc.*, vol. 1, no. November 2020, pp. 5709–5718, 2020.

[12] M. D. Do Nascimento *et al.*, "Which visual programming language best suits each school level? A look at Alice, iVProg, and Scratch," *EDUNINE 2019 - 3rd IEEE World Eng. Educ. Conf. Mod. Educ. Paradig. Comput. Eng. Career, Proc.*, no. May, 2019.

[13] C. C. Lu, J. C. Hong, F. F. Chen, and S. Y. Ma, "Elementary school students learn arduino programming to assemble sensory-controlled works," *Int. J. Inf. Educ. Technol.*, vol. 10, no. 4, pp. 265–270, 2020.

[14] A. Luxton-Reilly *et al.*, "Introductory Programming: A Systematic Literature Review ITiCSE working group; CS1; introductory programming; novice programming; systematic literature review; systematic review; lit-erature review; review; SLR; overview," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 55–106.

[15] B. Kaučič and T. Asič, "Improving introductory programming with Scratch?," *MIPRO 2011 - 34th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. - Proc.*, no. January, pp. 1095–1100, 2011.

[16] S. R. Sobral, "The First Programming Language and Freshman Year in Computer Science: Characterization and Tips for Better Decision Making," *Adv. Intell. Syst. Comput.*, vol. 1161 AISC, pp. 162–174, 2020.

[17] S. Crawford and E. Boese, "ActionScript: a gentle introduction to programming," *J. Comput. Sci. Coll.*, vol. 21, no. 3, pp. 156–168, 2006.

[18] C. Okoli, "A guide to conducting a standalone systematic literature review," *Commun. Assoc. Inf. Syst.*, vol. 37, no. 1, pp. 879–910, 2015.

[19] J. Moreno-León and G. Robles, "Code to learn with Scratch? A systematic

literature review," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 150–156.

[20] M. Noone and A. Mooney, "Visual and textual programming languages: A systematic review of the literature," *J. Comput. Educ.*, vol. 5, no. 2, pp. 149–174, 2017.

[21] R. T. Sataloff, M. M. Johns, and K. M. Kost, "Levels of Programming."

[22] T. J, "The motivation of students of programming," 2001.

[23] S. Bergin and R. Reilly, "The influence of motivation and comfort-level on learning to program," in *17th Workshop of the Psychology of Programming Interest Group*, 2005, no. June, pp. 293–304.

[24] P. Kinnunen and L. Malmi, "Why students drop out CS1 course?," *ICER 2006 - Proc. 2nd Int. Comput. Educ. Res. Work.*, vol. 2006, pp. 97–108, 2006.

[25] W. M. Kunkle and R. B. Allen, "The impact of different teaching approaches and languages on student learning of introductory programming concepts," *ACM Trans. Comput. Educ.*, vol. 16, no. 1, pp. 1–26, 2016.

[26] M. Mladenović, D. Krpan, and S. Mladenovi, "Learning programming from scratch," *Turkish Online J. Educ. Technol.*, vol. 2017, no. November Special Issue INTE, pp. 419–427, 2017.

[27] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching introductory programming: A quantitative evaluation of different approaches," *ACM Trans. Comput. Educ.*, vol. 14, no. 4, 2014.

[28] and D. D. Waleed Farag, Sanwar Ali, "Does language choice influence the effectiveness of online introductory programming courses?," *Proc. 14th Annu. ACM SIGITE Conf. Inf. Technol. Educ. (SIGITE '13).*, 2013.

[29] R. J. Enbody, W. F. Punch, and M. McCullen, "Python CS1 as preparation for C++ CS2," *ACM SIGCSE Bull.*, vol. 41, no. 1, pp. 116–120, 2009.

[30] T. B. Dinesh, "Visual programming," *Sci. Comput. Program.*, vol. 32, no. 1–3,

pp. 218–220, 1998.

[31] D. Weintrop and U. Wilensky, “Comparing block-based and text-based programming in high school computer science classrooms,” *ACM Trans. Comput. Educ.*, vol. 18, no. 1, pp. 1–25, 2017.

[32] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, “The scratch programming language and environment,” *ACM Trans. Comput. Educ.*, vol. 10, no. 4, pp. 1–15, 2010.

[33] J. M. Sáez-López, M. Román-González, and E. Vázquez-Cano, “Visual programming languages integrated across the curriculum in elementary school: A two year case study using ‘scratch’ in five schools,” *Comput. Educ.*, vol. 97, pp. 129–141, 2016.

[34] O. Ezenwoye, “What language? - The choice of an introductory programming language,” *Proc. - Front. Educ. Conf. FIE*, vol. 2018-Octob, 2018.

[35] S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and T. D. Singh, “Comparative Analysis of Python and Java for Beginners,” *Int. Res. J. Eng. Technol.*, vol. 7, no. 8, pp. 4384–4407, 2020.

[36] O. Ristić, D. Milošević, and V. Urošević, “The importance of programming languages in education,” *6th Int. Conf. Fac. Tech. Sci.*, no. May, pp. 243–249, 2016.

[37] D. Smith and A. Ali, “A welcomed reversal in computer science enrollments: Analysis of contributing factors and recommendations to sustain the growth,” *Online J. Appl. Knowl. Manag.*, vol. 6, no. 1, pp. 119–137, 2018.

[38] J. P. Bigham, L. Rosenblatt, P. Carrington, and K. Hara, “Vocal programming for people with upper-body motor impairments,” in *15th Web for All Conference W4A 2018: April 23-25, Proceedings*, 2018, pp. 1–10.

[39] N. Zaranis, V. Orfanakis, S. Papadakis, and M. Kalogiannakis, “Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study.,” *Int. J. Technol. Enhanc. Learn.*, vol. 1, no. 1, p. 1,

2016.

[40] A. Funke and K. Geldreich, “Measurement and visualization of programming processes of primary school students in scratch,” *ACM Int. Conf. Proceeding Ser.*, no. October 2018, pp. 101–102, 2017.

[41] H. Chetwynd, Frances; Aiken, Fiona and Jefferis, “Breaking the coding barrier: transition from Level 1 to Level 2 programming,” 2018.

[42] D. Saito, H. Washizaki, and Y. Fukazawa, “Comparison of text-based and visual-based programming input methods for first-time learners,” *J. Inf. Technol. Educ. Res.*, vol. 16, no. 1, pp. 209–226, 2017.

[43] M. Mladenović, I. Boljat, and Ž. Žanko, “Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level,” *Educ. Inf. Technol.*, vol. 23, no. 4, pp. 1483–1500, 2018.

[44] N. C. C. Brown, A. Altadmri, and M. Kolling, “Frame-based editing: Combining the best of blocks and text programming,” *Proc. - 2016 Int. Conf. Learn. Teach. Comput. Eng. LaTiCE 2016*, pp. 47–53, 2016.

[45] M. Rahman and R. Paudel, “Preliminary Experience and Learning Outcomes by Infusing Interactive and Active Learning to Teach an Introductory Programming Course in Python,” *Proc. Int. Conf. Front. Educ. Comput. Sci. Comput. Eng.*, pp. 51–57, 2018.

[46] Y. Bosse and M. A. Gerosa, “Why is programming so difficult to learn?,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 41, no. 6, pp. 1–6, 2017.

[47] K. Slonneger and B. L. Kurtz, *Formal Syntax and Semantics of Programming Languages*. 1995.

[48] J.-P. Pellet, A. Dame, and G. Parriaux, “How beginner-friendly is a programming language ? A short analysis based on Java and Python examples,” *12th Int. Conf. informatics Sch. Situation, Eval. Perspect.*, no. November, 2019.

[49] L. Mannila, M. Peltomäki, and T. Salakoski, “What about a simple language? Analyzing the difficulties in learning to program,” *Comput. Sci. Educ.*, vol. 16,

no. 3, pp. 211–227, 2006.

[50] E. Mehmood, A. Abid, M. S. Farooq, and N. A. Nawaz, “Curriculum, Teaching and Learning, and Assessments for Introductory Programming Course,” *IEEE Access*, vol. 8, pp. 125961–125981, 2020.

[51] M. Marimuthu and P. Govender, “Perceptions of Scratch Programming among Secondary School Students in KwaZulu-Natal, South Africa,” *African J. Inf. Commun.*, no. 21, pp. 51–80, 2018.

[52] M. Mladenović, D. Krpan, and S. Mladenović, “Introducing Programming To Elementary Students Novices By Using Game Development in Python and Scratch,” *EDULEARN16 Proc.*, vol. 1, no. July, pp. 1622–1629, 2016.

[53] A. Funke, K. Geldreich, and P. Hubwieser, “Analysis of scratch projects of an introductory programming course for primary school students,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, no. April, pp. 1229–1236, 2017.

[54] H. Y. Durak and T. Guyer, “Programming with Scratch in primary school, indicators related to effectiveness of education process and analysis of these indicators in terms of various variables,” *Gift. Educ. Int.*, vol. 35, no. 3, pp. 237–258, 2019.

[55] O. Mironova, I. Amitan, J. Vendelin, J. Vilipold, and M. Saar, “Teaching programming basics for first year non-IT students,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. 10-13-Apri, pp. 15–19, 2016.

[56] N. Zamin, H. Ab Rahim, K. S. Savita, E. Bhattacharyya, M. Zaffar, and S. N. Katijah Mohd Jamil, “Learning Block Programming using Scratch among School Children in Malaysia and Australia: An Exploratory Study,” in *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*, 2018, pp. 1–6.

[57] J. A. Martínéz-Valdés, J. Angél Vélazquéz-Iturbidé, and R. Hijon-Néira, “A (relatively) unsatisfactory experience of use of Scratch in CS1,” *ACM Int. Conf. Proceeding Ser.*, vol. Part F1322, 2017.

[58] Simon, R. Mason, T. Crick, J. H. Davenport, and E. Murphy, “Language choice in introductory programming courses at australasian and UK universities,” *SIGCSE 2018 - Proc. 49th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 2018-Janua, pp. 852–857, 2018.

[59] Adobe, *Programming ActionScript 3.0*. California, 2007.

[60] S. Leutenegger and J. Edgington, “A games first approach to teaching introductory programming,” *ACM SIGCSE Bull.*, vol. 39, no. 1, pp. 115–118, 2007.

[61] D. P. Astuti, Leonard, Y. B. Bhakti, and I. A. D. Astuti, “Developing Adobe Flash-based mathematics learning media for 7th-grade students of junior high school,” *J. Phys. Conf. Ser.*, vol. 1188, no. 1, 2019.

[62] H. Mustafidah, M. G. A. Imani, Sriyanto, and Suwarsito, “Development of natural science learning media in primary school using flash applications to increase student’s achievement,” *MATEC Web Conf.*, vol. 205, pp. 1–8, 2018.

[63] E. M. Saari, “Motivation for Trainee Teachers : Non-Computing Background Learn Action Script,” *Int. J. Comput. Inf. Technol.*, vol. 03, no. 01, pp. 133–138, 2014.

[64] The Chromium Projects, “Flash Usage Trends,” 2020. [Online]. Available: https://www.chromium.org/flash-roadmap/flash-usage-trends. [Accessed: 02-Jun-2021].

[65] Catalin Cimpanu, “Flash version distributed in China after EOL is installing adware,” *Zero Day*, 2021. [Online]. Available: https://www.zdnet.com/article/flash-version-distributed-in-china-after-eol-is-installing-adware/. [Accessed: 02-Jun-2021].

[66] and D. K. Meera Sridhar, Mounica Chirva, Benjamin Ferrell, Kevin W. Hamlen, “FLASH IN THE DARK: ILLUMINATING THE LANDSCAPE OF ACTIONSCRIPT WEB SECURITY TRENDS AND THREATS,” *J. Inf. Syst. Secur.*, vol. 13, no. 2, pp. 57–98, 2017.

[67] Y. Maheshwari and Y. Raghu Reddy, “A study on migrating flash files to HTML5/JavaScript,” *ACM Int. Conf. Proceeding Ser.*, no. February, pp. 112–116, 2017.