**METHODOLOGY**

This chapter shows the materials and methods that the researchers used for the de- velopment of CodeNect: Visual Programming Software for Learning Fundamentals of Pro- gramming.

**Materials**

For the development of CodeNect: Visual Programming Software for Learning Fun- damentals of Programming, the researchers used the following specifications:

For the software requirements, the following materials were used in the develop- ment of the software: Linux 5.4 Kernel with Manjaro distrubution as Operating System, Terminal for running commands, Vim for text and code editor, OpenGL for rendering, GLFW for windowing and input, DearImGUI for immediate mode graphical user interface (GUI), TinyC Compiler for running transpiled code at runtime, and C++ as programming language. The following are used for the deployment of the software: Microsoft Windows 7 and above, C++ Runtime libraries, and the C++ programming language.

For the hardware requirements, the following materials are used in the develop- ment of the software: Laptop with 2 GB of RAM (Random Access Memory), processor of Intel Core 2 Duo (1.4 GHz), and storage of 80 GB HDD (Hard Disk Drive). The following materials are used for the deployment of the software: at least Intel Core 2 Duo at 1.4 GHz,

2 GB of RAM, and 1 GB HDD of storage.

**Method**

The researchers used the V-Model methodology of Software Development Life Cycle (SDLC) for the proposed software to be developed. The V-Model methodology is a linear development methodology that focuses and follows a strict and incremental steps of stages. The initial phases are generally focused on planning and designing the system, the next phases are focused on implementation and actual programming. After that, the model will go in upwards direction for testing and verification of the project. The development of the software follows the timeline (See Appendix Figure [32).](#_bookmark0)

The researchers conducted a survey to gather data from students, instructors, and learners in the field of technology and under the course with programming subjects such as

Bachelor of Science in Information Technology, Bachelor of Science in Computer Science,

32

and Bachelor of Science in Computer Engineering as respondents. The data gathered (See Appendix Figure [1)](#_bookmark3) are evaluated and assessed to determine the knowledge and un- derstanding of the respondents in regards to the fundamentals of programming, experience and feedback on traditional text-based tools and software. Problems were identified and the Ishikawa Diagrams were constructed (See Appendix Figure [15,](#_bookmark6) [16,](#_bookmark5) and [17).](#_bookmark5)

The researchers gathered twelve students with programming subjects and have conducted pre-assessment through Google Form to evaluate the current knowledge and understanding on programming fundamentals of the respondents who use traditional text-

based programming. The profile of the respondents are shown in Appendix E.

**Pre-Assessment Test Results**

For the pre-assessment test, the researchers composed ten questions pertaining to various programming concepts in traditional text-based code (See Appendix A) and gath- ered twelve respondents with programming subjects through the use of online survey and assessment form. The respondents were not given any time limitation.

Table [1](#_bookmark7) shows the result of the pre-assessment test.

Table 1. Pre-assessment Result

**CORRECT**

**INCORRECT**

**TOTAL**

**TOPICS**

**n**

**%**

**n**

**%**

**n**

**%**

1. Data Types
2. Unary Comparison
3. Binary Comparison
4. Ternary Comparison
5. Expression Evaluation
6. Array
7. Branching
8. For-Loop

5

12

8

4

4

6

7

3

41.67%

100%

66.67

33.33%

33.33%

50%

58.33%

25%

7

0

4

8

8

6

5

9

58.33%

0%

33.33%

66.67

66.67

50%

41.67%

75%

12

12

12

12

12

12

12

12

100%

100%

100%

100%

100%

100%

100%

100%

Table [2](#_bookmark8) shows the result of the pre-assessment test (mean = 2.55, SD = 2.17) in

the Likert scale.

33

Table 2. Pre-assessment Result Evaluation

**STANDARD**

**DEVIATION**

**TOPICS**

**MEAN**

**INTERPRETATION**

1. Data Types
2. Unary Comparison
3. Binary Comparison
4. Ternary Comparison
5. Expression Evaluation
6. Array
7. Branching
8. For-Loop

**AVERAGE**

2.08

5.00

3.33

1.67

1.67

2.50

2.92

1.25

**2.55**

2.57

0.00

2.46

2.46

2.46

2.61

2.57

2.26

**2.17**

FAIR EXCELLENT GOOD POOR POOR

FAIR GOOD POOR

**FAIR**

A post-assessment test is

planned

to be conducted during software evaluation

to assess the effect of the study on the knowledge and understanding on programming fundamentals of the respondents.

After data gathering, the researchers assessed the gathered data and studied the information in order to construct a context diagram representing the manual way that the study will solve (See Appendix Figure [30).](#_bookmark1) The schedule of the development and the allotted time for each is task is planned through the Gantt chart (See Appendix Figure [32).](#_bookmark0)

After which, the researchers designed and developed specifications that served as the blueprint of the software. The libraries, packages, tools, and more are finalized and prepared for later use (See Appendix Figure [31).](#_bookmark4)

Further, the researchers identified and defined the scopes and specific features of each module and how each is integrated along the system to work with other modules and components to ensure that each module is decoupled and can be tested without depen- dency in other module (See Appendix Figure [33).](#_bookmark2)

The researchers started to program each module in the C++ programming lan- guage using Vim as the primary text editor. Compiling and running the software will be done by running a command in the terminal. The rendering backend is the OpenGL and the GLFW for the windowing and input framework. Each functionality of each module was

tested using unit tests by running the test every functionality that is implemented. After

34

each module passes the associated test, all is coupled and integrated to a single system. The end product of this phase is the CodeNext: Visual Programming Software for Learning Fundamentals of Programming.

After the software was developed, the researchers applied tests in the software that was developed. Further testings are done. Bugs, errors, and misbehaviors were han- dled and fixed. This test ensured that the functionality, efficiency, usability, portability, and reliability of the software meets the standard.

Finally, the software was subjected to acceptance testing, ISO 9126 - Product Quality software standards was used. Several evaluators tested the software thoroughly to specifically evaluate their experience and their learning using the standard metrics. The technical and non-technical evaluators assessed the functionality, reliability, usability, main- tainability, portability, efficiency, and user-friendliness of the software.

The functionalities of the software were evaluated accordingly. The features and effectiveness of the software will be verified by ten college students taking programming subjects as the non-technical evaluators and another ten IT professionals as technical eval- uators. Feedbacks and remarks are collected from the evaluators for further analysis by the researchers.

The rating used for the evaluation is described in Table [3:](#_bookmark9) 4.21 - 5.00 as Excel- lent which indicates that the software fully meets and far exceeds the most expectations and requirements. 3.41 - 4.20 as Very Good which indicates that the software fully meets and exceeds several expectations and requirements. 2.61 - 3.40 as Good which indicates that the system fully meets the requirements. 1.81 - 2.60 as Fair which indicates that the software lacks in meeting the expectations and requirements. 1.00 - 1.80 as Poor, which

indicates that the software fails to meet the expectations and requirements.

**LIKERT SCALE**

**RANGE**

**INTEPRETATION**

4.21 - 5.00

Excellent

3.41 - 4.20

Very Good

2.61 - 3.40

Good

1.81 - 2.60

Fair

1.00 - 1.80

Poor

35

Table 3. Likert Scale for the System Evaluation

The post-assessment test was conducted on May 25, 2021 in response to the pre-

assessment test which was conducted during the initial stage of the study. The questions for the post-assessment are the same question during the pre-assessment test.

**Post-Assessment Test Results**

For the post-assessment test, the researchers included assessment exercises per- taining to various programming concepts in visual code code and gathered twelve respon- dents with programming subjects through the use of online survey and assessment form. The respondents were not given any time limitation.

Table [4](#_bookmark10) shows the result of the post-assessment test.

Table 4. Post-assessment Result

**CORRECT**

**INCORRECT**

**TOTAL**

**TOPICS**

**n**

**%**

**n**

**%**

**n**

**%**

1. Data Types
2. Unary Comparison
3. Binary Comparison
4. Ternary Comparison
5. Expression Evaluation
6. Array
7. Branching
8. For-Loop

10

12

11

7

8

9

12

10

83.33%

100%

91.67%

58.33%

66.67%

75%

100%

83.33%

2

0

1

5

4

3

0

2

16.67%

0%

8.33%

41.67%

33.33%

25%

0%

16.67%

12

12

12

12

12

12

12

12

100%

100%

100%

100%

100%

100%

100%

100%

Table [5](#_bookmark11) shows the result of the post-assessment test (mean = 4.12, SD = 1.57) in

the Likert scale.

Table 5. Post-assessment Result Evaluation

**STANDARD**

**DEVIATION**

**TOPICS**

**MEAN**

**INTERPRETATION**

1. Data Types
2. Unary Comparison

4.17

5.00

1.94

0.00

VERY GOOD

EXCELLENT

36

1. Binary Comparison
2. Ternary Comparison
3. Expression Evaluation
4. Array
5. Branching
6. For-Loop

**AVERAGE**

4.58

2.92

3.33

3.75

5.00

4.17

**4.12**

1.44

2.57

2.46

2.26

0.00

1.94

**1.57**

EXCELLENT GOOD GOOD

VERY GOOD EXCELLENT VERY GOOD

**VERY GOOD**

Table [6](#_bookmark12) shows the result of the pre-assessment test (mean = 2.55, SD = 2.17) and

the result of post-assessment test (mean = 4.12, SD = 1.57). The comparison shows an increase of 61.57% in using the software (mean difference = -1.57, SD difference = 0.6).

Table 6. Pre and Post Assessment Result

**STANDARD**

**DEVIATION**

**TEST**

**MEAN**

**INTERPRETATION**

1. Pre-assessment
2. Post-assessment

**DIFFERENCE**

2.55

4.12

**-1.57**

2.17

1.57

**0.6**

FAIR

VERY GOOD