

METHODOLOGY

This chapter shows the materials and methods that the researchers will use for the development of CodeNect: Visual Programming Software for Learning Fundamentals of Programming.

Materials

For the development of CodeNect: Visual Programming Software for Learning Fundamentals of Programming, the researchers use the following specifications:

For the software requirements, the following materials are used in the development of the software: Linux 5.4 Kernel with Manjaro distribution as Operating System, Terminal for running commands, Vim for text and code editor, Kha for the graphical, media, and control framework, zui for the base user-interface, and Haxe as programming language. The following are used for the deployment of the software: Microsoft Windows 7 and above, C++ Runtime libraries, and the Haxe programming language.

For the hardware requirements, the following materials are used in the development 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 will use 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 5).

The V-Model figure shows the following stages:

Requirements.

In this stage 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, and Bachelor of Science in Computer Engineering as respondents. The data gathered (See Appendix Figure 3) are evaluated and assessed to determine the knowledge and understanding of the respondents in regards to the fundamentals of programming, experience and feedback on traditional text-based tools and software. Problems are identified and the Ishikawa Diagrams are constructed (See Appendix Figure 16, 17, and 18).

System Design. The researchers will assess the gathered data and study the information in order to construct a context diagram representing the manual way that the study will solve (See Appendix Figure 3). The schedule of the development and the allotted time for each task is planned through the Gantt chart (See Appendix Figure 5).

Architecture Design.

The researchers in this stage will design and develop specifications that will serve as the blueprint of the software. The libraries, packages, tools, and more are will be finalized and prepared for later use (See Appendix Figure 4).

Module Design.

The researchers will identify and define the scopes and specific features of each module and how will each be integrated along the system to work with other modules and components to ensure that each module is decoupled and can be tested without dependency in other module (See Appendix Figure 6).

Implementation and Coding.

The researchers will start to program each module in the Haxe programming language 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 will be the Kha and zui framework. Each functionality of each module will be tested using unit tests by running the test every functionality that will be implemented. After each module passes the associated test, all will be coupled and integrated to a single system. The end product of this phase is the CodeNext: Visual Programming Software for Learning Fundamentals of Programming.

Testing.