Neuromorphic For Kids – Tech Review

Rebecca Croysdale

Fall Term: May 28, 2020

Team 67: "Get Good, Kids!"

I. Abstract

This tech review explores different options for art design, databases for user progress, and the type of application for the lessons. The technologies explored are for a project intended to provide coding lessons to high school students on Native American reservations. The three applications explored for art design are Aseprite, Photoshop and Paint.NET. The databases explored are MySQL, MongoDB and NoSQL. The types of applications explored are a website, a phone application, and a downloadable executable. The pros and cons of each option are listed and then a selection is made at the end of each section.

II. Description of project

Native American high school students do not have sufficient access to the tools required to pursue a career in Computer Science. The proposed solution is to create an Engineering/Computing lesson plan that engages students via challenges in a game-like format that teaches them the basics of coding up through advanced topics such as neuromorphic computing.

III. Piece 1: Art Design

We will be creating a set of lessons that will be packaged as a game. The game needs to be engaging to high school students so that they are encouraged to continue playing and learning. We also need to ensure that the lessons are approved by the elders of the reservations, so we may need to make the art specifically targeted towards goals made by the elders. We may need to make sprites and backgrounds for characters to show stories in the lessons. A good art application will be essential towards these goals.

Option 1: Aseprite

Pros: Aseprite is specifically made for making pixel art. Our game would primarily use pixel art for the art design. It is also easy to make animations using this application. Aseprite comes with an intuitive interface and many extra tools beyond the default art program tools like 'paint bucket.' [1]

Cons: Aseprite costs $15 as a one time fee, but it can be free if you compile the source-code yourself. It can take some time to become familiar and efficient with this tool.

Option 2: Photoshop

Pros: There are an abundance of resources to learn how to use Photoshop. Photoshop can be used to create pixel art, or high resolution images. We may be able to make nice backgrounds or edit pictures of art submitted by elders that we work with. There are many tools included in Photoshop so it is very versatile. [2]

Cons: Photoshop costs $10 per month for students, while other applications are free. The interface can be overwhelming, so designers will need to complete tutorials before using it.

Option 3: Paint.NET

Pros: Paint.NET is free, and at least one of the team members has used it before. There are an abundance of tutorials. There are free plugins that can be downloaded for extra features. Paint.NET can make pixel art and can be used to make tiled images for games. [2]

Cons: Paint.NET is not made for animations. Animations would be important for creating a game. Paint.NET cannot easily layer images to be compared as parts of an animation. Paint.NET can only be used on Windows machines.

Selection

We will be using Aseprite to create our images. This tool is free if we compile it ourselves. It is specifically designed for creating pixel art, which is what we intend to create. We will easily be able to create animations with Aseprite, unlike with Paint.NET, which requires extra work to create animations.

IV. Piece 2: Progress Data Storage

We will store progress data for each user, so that when they return to the application they can resume the lessons. This could also allow the lessons to build off of each other in an overall game. The progress database could be stored on a server, or on each user's device. If the database is stored on the device, we would not need to deal with user logins. However, if there were a leaderboard, this would open up the database to cheating. A user could hack the progress database stored on their client so that they have a higher leaderboard position. If the progress database is stored on a server, we would have to make sure the database is secure and that the login information is encrypted.

Option 1: MySQL

Pros: MySQL is compatible with most operating systems, including Windows, Linux, and MacOS, so we would be able to use it on any devices the app might be run on. MySQL is low maintenance for a small database and is secure. [3] [4]

Cons: MySQL can have issues with unstructured complex data, but since this would be a choice made before having any data, we can ensure that the data is well structured to begin with. MySQL is also not very scalable. [3] [4]

Option 2: MongoDB

Pros: MongoDB has a good and helpful community, so it should be easy to get help with setting it up or working through bugs. MongoDB is good at processing large volumes of data, so it would be a good choice if we choose to have the database on a server, or if we want to have more detailed data. [3] [4]

Cons: Badly structured data could result in duplicates. MongoDB does not do as well with taking in existing data. [3] [4]

Option 3: NoSQL

Pros: NoSQL is designed to have less maintenance. It has auto repair capabilities. NoSQL is good for storing large amounts of data. [5]

Cons: NoSQL is new, so there is not as much support as there is for other database models.

Selection

If we use a database for this project, we will use MySQL. Our database(s) would be small so MySQL would be manageable. We would also be building our data to be well structured from the beginning, so we don't need the flexibility of MongoDB.

Piece 3: Type of application

Our application should be runnable in a convenient way for users. We could have a website run off of a server, a phone application, or a downloadable executable. We expect our users to have phones that they can use at any time. We also have the option of providing an Intel NUC that can serve content to a website that students can access with another device.

Option 1: Website

Pros: This would be easy to access from any device that can run a browser. This includes phones, tablets and computers. This website could be hosted by an Intel NUC. If the website is accessed via a computer, it will be easy for users to type code. The NUC can use the Intel compute stick for the higher level AI coding lessons.

Cons: A website would require either cookies to maintain progress, or a server run database with user logins. We would need to implement security for logins and the database would need to be protected. The progress would need to be maintained in some way if the server is shut down. If the website is accessed through a phone, the small keyboard may make it difficult to write code. In addition, the UI would need to be flexible enough to be viewable on a phone.

Option 2: Phone application

Pros: We would be able to design a simple UI that would fit on a phone screen. Users would be able to work on the lessons whenever they want to. We may not need to create user logins, since the data could be stored locally. We could also have data stored on a server, if desired.

Cons: Users would need to have a phone to use the app. We are only able to make an android app, since none of us have Mac computers to develop Apple applications. It would be difficult to type code on the small keyboard of a phone. A phone app would then consist largely of block based coding which is not ideal for teaching real world coding skills.

Option 3: Downloadable Executable

Pros: This would be able to be used offline on school computers. The progress data for each student would be stored on their individual computer, so it would not need to be as secure and we would not need to run a server.

Cons: Students would likely only use our product while at school during a technology class. Since the problem this project is solving is inadequate technology education resources, it is reasonable to assume that schools may not have enough computers for each student to use.

Selection

We will be making a website served from an Intel NUC. These can be provided to each classroom along with the Intel compute stick for the AI lessons. The website could be accessed via any device that can run a browser.

REFERENCES

[1] Aseprite alternatives & reviews. Available at https://alternative.me/aseprite May 28, 2020.

[2] Best pixel art maker tools. Available at https://www.pcgamer.com/best-pixel-art-maker-tool May 28, 2020.

[3] Mongodb vs mysql comparison: Which database is better? Available at https://hackernoon.com/mongodb-vs-mysql-comparison-which-database-is-better-e714b699c38b May 28, 2020.

[4] Mongodb vs mysql: A comparative study on databases. Available at https://www.simform.com/mongodb-vs-mysql-databases May 28, 2020.

[5] 7 pros and cons of nosql. Available at https://greengarageblog.org/7-pros-and-cons-of-nosql May 28, 2020.