

UNIVERSITY OF

NAIROBI

FACULTY OF EDUCATION

DEPARTMENT OF COMPUTING AND INFORMATICS

# TOPIC: TEACHING PROGRAMMING THROUGH GAMING

**STUDENT NAME: JOSEPH KENYATTA**

**REG NO. E46/3939/2019**

# COURSE UNIT: ICS 410a

**SUBMITTED ON: JANUARY 30 2023**

Abstract: programming is a concept that most students think is too hard to comprehend. However, this is a very marketable skill it is essential that these students master it. In this paper, I discuss the problems that students encounter while learning programming concepts and propose a simple game they can play to hone their skills.

Table of Contents

[TOPIC: TEACHING PROGRAMMING THROUGH GAMING 1](#_Toc126178273)

[COURSE UNIT: ICS 410a 1](#_Toc126178274)

[Abstract: 1](#_Toc126178275)

[INTRODUCTION 3](#_Toc126178276)

[**Problem Statement** 3](#_Toc126178277)

[**Objectives** 3](#_Toc126178278)

[**Research Questions** 4](#_Toc126178279)

[**Scope** 4](#_Toc126178280)

[**Justifications** 4](#_Toc126178281)

[**Limitations** 4](#_Toc126178282)

[LITERATURE REVIEW 5](#_Toc126178283)

[**Introduction** 5](#_Toc126178284)

[**History of Programming Through Gaming** 5](#_Toc126178285)

[**Literature Review of Selected Games** 5](#_Toc126178286)

[*CSS DINER* 5](#_Toc126178287)

[*CodinGame* 6](#_Toc126178288)

[*Elevator Saga* 8](#_Toc126178289)

[METHODOLOGIES 9](#_Toc126178290)

[**Population of the Study** 10](#_Toc126178291)

[**System Analysis and Design** 10](#_Toc126178292)

[*Requirements* 10](#_Toc126178293)

[*System Design* 10](#_Toc126178294)

[*Implementation* 11](#_Toc126178295)

[*Testing* 11](#_Toc126178296)

[*Deployment* 11](#_Toc126178297)

[*Maintenance* 11](#_Toc126178298)

[**Data Collection and Analysis** 11](#_Toc126178299)

[**Possible Outcomes** 12](#_Toc126178300)

[**Ethical Considerations** 12](#_Toc126178301)

[Project Schedules and Budgets 12](#_Toc126178302)

[**Budget** 13](#_Toc126178303)

[Bibliography 13](#_Toc126178304)

# INTRODUCTION

Programming is a practice that strengthens our capacity for logical thought and problem-solving. It teaches us how to carry out a task with the aid of software or a computer program. So, to put it simply, programming is the process of using computer language to construct a solution to a problem (What is programming?, n.d.). Playing video games is known as gaming and is frequently done on a dedicated gaming console, computer, or smartphone (Gaming: Definition, n.d.).

Nowadays it is compulsory for any student enrolled in any ICT-related program in order to get good grades in the exam these students have knowledge on use of computers for software such as Microsoft Access, publishers as well as gaming software such as FIFA 23 (Sports games, 2022), Grand Theft Auto V (Action games, 2015), Minecraft (Action games, 2021) etc, but have little or no prior knowledge on programming. Therefore, programming languages such as C#, C++, Java, Python etc, are quite an uphill task when it comes to mastering the content. Others lose interest quite easily leading to a loss of motivation when studying for programming exams.

## **Problem Statement**

1. This project will teach students to think critically while creating any software of their choice.
2. The game will assist the students to understand how code works in a program.
3. This program will enhance student creativity thus they will be more productive when it comes to programming.
4. This will be an effective way of ensuring students don’t lose motivation while learning how to program.

## **Objectives**

By mixing programming principles into engaging and interactive gameplay, the aim is to increase the accessibility and attraction of programming education to students. This study also has other objectives which include:

1. To foster students’ problem-solving and critical thinking skills by incorporating programming challenges into the game.
2. To encourage students to explore programming concepts and develop a passion for the subject through hands-on learning
3. To provide a platform for students to practice and apply their programming skills in a creative and interactive environment.
4. To provide a unique and alternative approach to traditional programming education, catering to different learning styles and preferences.

The final result of this study will be very helpful to the education sector as it can be integrated into the curriculum to enable students of all levels to apply it in their daily programming endeavors.

## **Research Questions**

This research is being carried out to answer the following questions:

1. How does the use of programming-based games impact student problem-solving and critical thinking skills?
2. How does the integration of gaming elements affect student motivation and engagement in programming courses?
3. What is the best platform for students to engage in programming-based games?
4. To what extent do programming-based games improve student performance on programming assessments compared to traditional teaching methods?

## **Scope**

This study will be carried out on Campuses around Nairobi CBD and its environs. The major part of the study, however, will be carried out at Kenya Science Campus. This will involve students who are taking a course in ICT.

## **Justifications**

The following research justifications will enable this project to take place:

Student engagement. This game will provide a fun and engaging way to learn to program, making it easy for students to retain information and stay motivated.

This game will also challenge students to think critically and creatively to solve problems. By doing so, these students will get hands-on experience with coding, which is very important for developing software. In addition to this, students can also collaborate with each other to complete different levels of the game thus developing teamwork. These skills will also prepare them for careers in ICT-related fields such as software engineering.

## **Limitations**

These are some of the limitations that will hinder this game while being used by the students:

1. The game may not provide a realistic experience of handling errors. Errors are an inevitable part of the programming process.
2. Not every student will enjoy playing this programming game, limiting the audience for this method of teaching.
3. This game will be developed for use by students who have access to a laptop. Those who don’t have them will not be able to use this software, thus they will be disadvantaged.
4. This game will have pre-set objectives that may restrict creative problem-solving.
5. In an attempt to make the game fun, some aspects of programming will be simplified, leading to a distorted understanding of the actual subject.

# LITERATURE REVIEW

## **Introduction**

The practice of teaching programming ideas to students while playing games has grown in popularity in recent years. According to studies, students who learn to program using games retain and comprehend the material better. Furthermore, game-based learning settings can offer rapid feedback, present problems for students to overcome and encourage innovation (Yang & Chen, 2021). This literature review will look at some of the games that have been developed

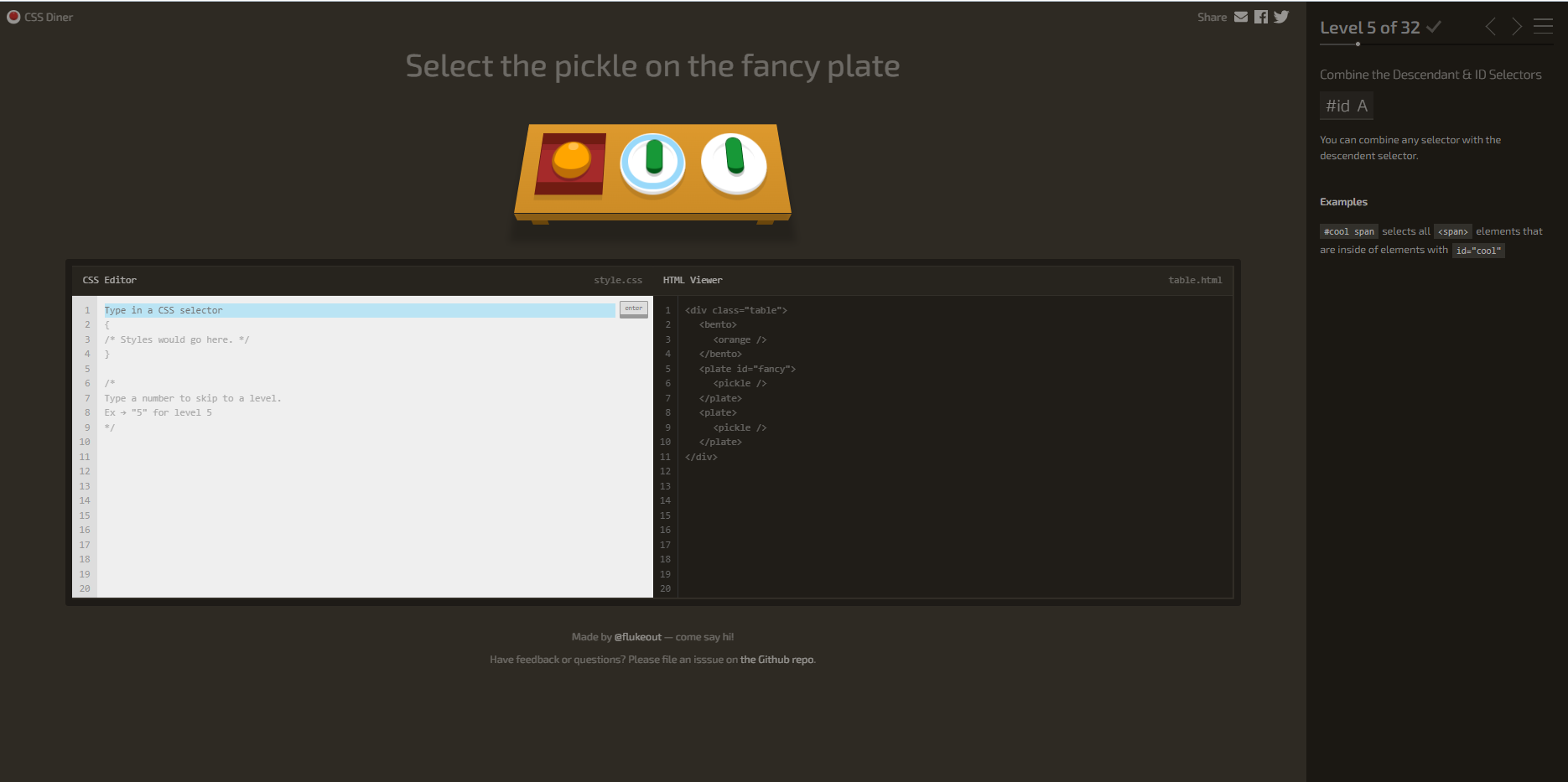
## **History of Programming Through Gaming**

The first educational game to teach programming concepts was called “The Adventures of Tron” (Adventures of Tron-atari 2600, 1982). For the purpose of instructing students in programming concepts like logic, algorithms, and problem-solving techniques, educational computer games were created in the 1980s and 1990s. The development of visual programming languages like Scratch in the 1990s has made it simpler for kids to understand programming fundamentals (Combéfis, Dagiene, & Beresnevičius, 2016). There are many programming games available now for kids and beginners of all ages, both in print and digital formats. The learning process is made enjoyable and participatory by the use of a number of interactive and engaging techniques in these games to teach programming topics (Crawford, 1982).

## **Literature Review of Selected Games**

### *CSS DINER*

A fun and engaging method to learn CSS selectors is through CSS Diner (CSS Diner: CSS Selectors, n.d.). Players are given a variety of challenges in the game that require them to use CSS selectors to pick out HTML elements on a website. More complex choices and concepts are introduced to players as they go through the levels. The game is simple to use and has an intuitive design. The gamification of the learning process in CSS Diner is one of its advantages. Users are able to interact with the material in a way that is entertaining and less scary than reading through a tutorial or a textbook by having the material presented in a game format. The difficulties are well-designed and get harder over time, allowing players to expand their knowledge as they advance in the game. The accessibility of CSS Diner is another advantage. The game may be played online and doesn't need to be downloaded or additional software. For those who are learning CSS selectors for the first time and want to do it quickly and simply, it is a fantastic resource.



*Fig. 1 Shows a preview of CSS Diner GUI*

Although CSS Diner is a great game for learning web development, it has some limitations which include:

1. Limited scope- It focuses solely on CSS selectors. It does not cover any other aspects of CSS or web development.
2. Limited interaction- The game is primarily text-based, with little interaction beyond selecting elements on a web page. Some users may find this to be a less engaging way to learn.
3. No feedback- There is no feedback provided when players make mistakes, which can make it difficult for users to learn from their mistakes and improve.
4. No explanation- The game does not provide any explanation for why certain selectors are used making it really hard to understand some concepts.
5. Not suitable for advanced users who are already familiar with CSS selectors thus it may not provide enough challenge or new useful information.

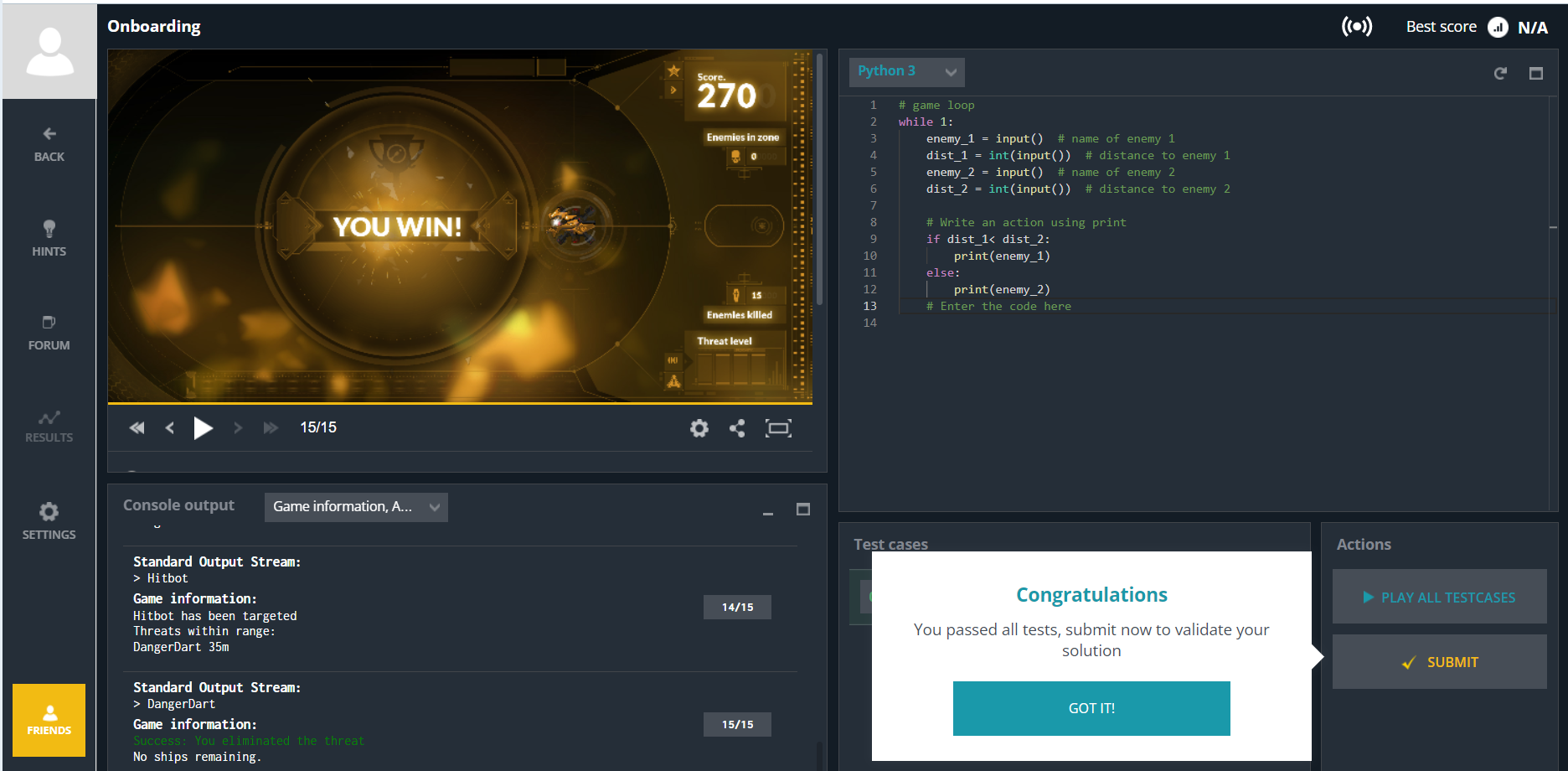
### *CodinGame*

CodinGame (Onboarding, 2022) is a comprehensive and innovative platform that offers interactive coding challenges, tutorials, and assessments to help people develop their coding skills. It covers a broad range of programming languages and topics, including algorithms, data structures, databases, and more, making it a valuable resource for individuals of all skill levels, from beginners to more experienced coders.

Studies and literature reviews have consistently shown that Codingame is an effective tool for learning to code and improving one's coding skills. The platform's gamified approach to coding makes the experience enjoyable and engaging, while its real-world coding problems and supportive community provide users with a practical and practical environment to practice their skills. The gamification of coding has been shown to be an effective way to motivate and engage users, making the experience of learning to code more enjoyable and memorable.

CodinGame also offers a variety of coding challenges that range in difficulty, allowing users to progress at their own pace. The challenges are designed to help users develop their coding skills through hands-on experience, providing them with practical coding problems that they can solve and learn from. This hands-on approach to learning has been shown to be an effective way to develop coding skills, as users can immediately see the results of their work and receive feedback on their solutions.

In addition to its interactive coding challenges, Codingame also offers a range of tutorials and assessments that provide users with a comprehensive overview of specific coding topics. The tutorials are designed to be easy to follow, providing step-by-step instructions and clear explanations of the concepts being taught. The assessments are used to evaluate the coding skills of individuals, providing a standardized way to measure their abilities and track their progress.



*Fig. 2 Shows an interface for playing CodinGame using Python3*

Employers and universities have also taken note of Codingame's potential (Screen developers with coding assessments, 2022), using it as a tool for assessing the coding skills of potential hires and students. The platform provides a standardized and efficient way to evaluate the coding abilities of individuals, making it a valuable resource for these organizations. The assessments offered by Codingame allow employers and universities to quickly and accurately assess the coding skills of potential hires and students, reducing the time and resources needed to conduct these evaluations.

CodinGame also provides a supportive community for users (Onboarding - Puzzle discussion, 2022), making it a great place to connect with other coders and share knowledge and experience. The community is made up of individuals from all over the world, providing users with a diverse group of individuals to connect with and learn from. The community also provides a place for users to receive feedback and support on their coding projects, helping them to overcome challenges and continue to develop their skills.

In conclusion, Codingame is a highly recommended platform for anyone looking to improve their coding skills. Its interactive coding challenges, tutorials, and assessments, combined with its gamified approach and supportive community, make it an effective and enjoyable way to learn and develop coding abilities. Whether you are a beginner or an experienced coder, CodinGame provides a comprehensive and practical environment for developing and improving your coding skills.

Although CodinGame is advantageous to programmers, it has some limitations, which include but are not limited to:

1. No access to local file system. This makes it difficult to manage large projects.
2. It provides limited resources and tutorials for learning to code.
3. It does not provide collaborative coding, thus one can’t work on a project with other programers.

### *Elevator Saga*

Coding game Elevator Saga is incredibly well-liked by fans of programming. The objective of the game is to challenge players to design code that will manage elevators in a virtual building and move passengers quickly and with the fewest number of steps ( The elevator programming game, 2015). The game has garnered accolades for its straightforward but difficult gameplay as well as for being a fun and useful way to learn about algorithms and programming ideas (TOP 5 coding games to learn programming, 2022). The game is made to be simple to use and open to players of all experience levels, from novices to specialists. Players may easily begin playing and programming right away thanks to the game's simple design and interface, and the progressive increase in complexity enables them to pick up new abilities at their own speed. The need to constantly enhance one's code and advance to new levels makes the game extremely addictive. Elevator Saga's educational value is one of its main advantages. State machines, multi-threading, and optimization are just a few of the programming principles and algorithms that the game instructs players on. For instance, in order to optimize their code and obtain the best score, players must think about the order in which they pick up and drop off passengers as well as how they prioritize different elevators. This teaches participants the value of optimizing algorithms and using resources wisely, two abilities crucial in the field of computer science.



*Fig. 3 shows the user interface for Elevator Saga*

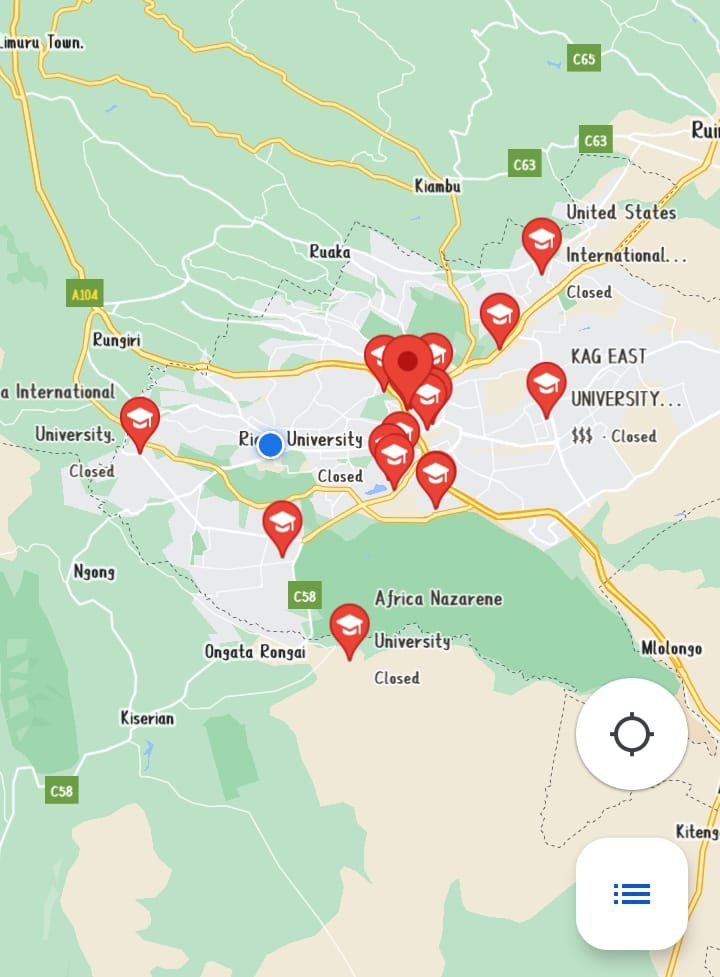
For those who enjoy programming, Elevator Saga is a popular game that blends learning and entertainment.

Some of the limitations that a user encounters when playing Elevator Saga are as follows:

1. The game is only available for single players. There is no interaction with other players.
2. It has a limited number of levels. There are 30 passengers to transport in a building with one elevator and ten stories.
3. Elevators cannot be operated manually, which decreases player interaction and control.

# METHODOLOGIES

The methodologies and strategies that will be employed to construct the system are covered in this chapter. The location of the project and the system's beneficiaries will be thoroughly and in-depth explored in this chapter. The Teaching Programming Through Games system's target audience, data sampling techniques, system analysis and design, data gathering techniques, and analysis and testing strategies will all be discussed. Considerations for this system's ethical clearance round out the chapter. The Nairobi County area's campuses will adopt this system. Students at these universities and anyone else who might be interested in participating in the exercise are the system's main beneficiaries.



*Fig. 4 Shows a map of major universities around Nairobi County*

*Source: https://maps .google.com/*

## **Population of the Study**

The targeted population will be majorly students in the above-shown map that are taking a course in ICT or a related one. They should be conversant with the concept of programming, data structures, and algorithms. Samples will be taken from 500 students who are interested in taking programming to the next level. This study will use the probability sampling technique using the formula :

Sampling size

Where:

* N=population size
* Z=Critical value of the normal distribution at the required confidence level, (1.96), the significant level set at 95% confidence which corresponded to Z value of 1.96.
* P= Sample proportion, i.e. (0.4)
* e= Margin of error, i.e. (0.05)

## **System Analysis and Design**

The system will be developed using the Waterfall model of the System Development Life Cycle.

### *Requirements*

* Through interactive challenges, players will be able to learn programming fundamentals.
* The game will give the player fast feedback on their choices.
* Beginners should be able to play and comprehend the game with ease.

### *System Design*

* The game will be a 2D platformer based on hangman in which the player uses code to direct a character through numerous levels.
* The player will write code snippets in a programming language before the hangman completes attaching all its limbs in order to progress to the next level.
* The game will include a code editor, visual feedback on the player's code execution, and instructional text.
* The game will have a level progression system to introduce new programming concepts as the player progresses.

### *Implementation*

Code will be written to develop the game. The game will be developed using Pygame which is a python framework for developing 2D games.

### *Testing*

I will test the game to ensure that it meets the requirements before making a beta version available to a test audience for further testing

### *Deployment*

The game will be deployed on a free hosting site for ease of download by any student who would to learn programming concepts using this game.

### *Maintenance*

I as the developer will provide continued support for the game as well adding new features in other versions of the game.

## **Data Collection and Analysis**

* Surveys: This study will Collect feedback from players through surveys to understand their experience and opinions on the game's effectiveness in teaching programming.
* User Testing: The study will conduct user testing sessions to observe players as they play the game and identify areas of difficulty or confusion.
* Analytics: This research will collect data on player behavior and progress through the game, such as completion rates, time spent on specific levels, and progress towards learning programming concepts.
* Focus Groups: The developer will gather a group of players to discuss their experiences playing the game, what they learned, and any suggestions for improvement.
* Interviews: The developer will conduct one-on-one interviews with players to gather in-depth feedback and insights on their experiences with the game.
* Questionnaires- To collect data on what players would like to interact with in the game.

Simple, descriptive, statistical, and analytical methods including histograms, time series, frequencies, percentages, graphs, and charts will be used to assess the data gathered from the various sources.

## **Possible Outcomes**

The following are the possible outcomes of this project:

* Successful adoption as well as its effects on academic results
* Favorable response from consumers and educators
* Increased interest in technology and computer science increased demand for programming courses
* Greater financial gain for game producers and educational organizations
* Increased tech literacy
* Improvement of the gaming sector as a teaching aid
* Better employment chances for programmers.

## **Ethical Considerations**

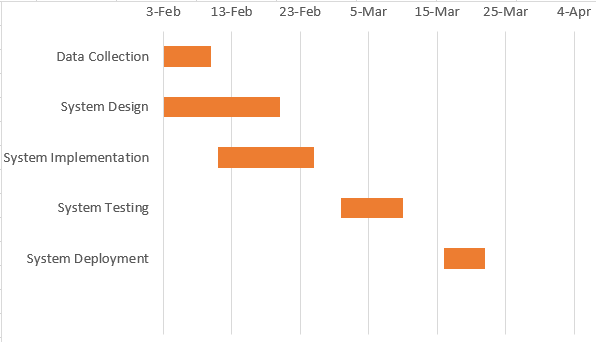
This study will take the following ethical issues into consideration:

* Privacy: Ensuring that user information is safe and not misused.
* Making the game accessible and engaging for a wide variety of players.
* Fair play: Avoiding game design that promotes cheating or unfair advantages.
* Making sure the game's content is suitable for the target age range.
* Avoiding game designs that promote excessive play and disregard for obligations in real life to avoid addiction.

# Project Schedules and Budgets

The proposed schedule is presented in the form of a chart

|  |  |
| --- | --- |
| Task | Start Date |
| Data Collection | 3rd February 2023 |
| System Design | 11th February 2023 |
| System Implementation | 1st March 2023 |
| System Testing | 16th March 2023 |
| System Deployment | 26th March 2023 |

****

## **Budget**

The researcher is planning to visit at least 12 campuses around Nairobi. The transport cost that will be incurred will be about KSH 150 per campus. This will attract a fee of about KSH 1,800 total.

# Bibliography

(n.d.). Retrieved from CSS Diner: CSS Selectors: https://flukeout.github.io

*The elevator programming game*. (2015). Retrieved from Elevator Saga: https://play.elevatorsaga.com

*Action games*. (2015). Retrieved from Steam: https://store.steampowered.com/app/271590/Grand\_Theft\_Auto\_V/

*Action games*. (2021). Retrieved from Steam: https://store.steampowered.com/app/1672970/Minecraft\_Dungeons/

Adams, D. (n.d.). *Core Techniques and Algorithms in Game Programming*.

*Adventures of Tron-atari 2600*. (1982). Retrieved from RetroGames: https://www.retrogames.cz/play\_574-Atari2600.php?language=EN

Combéfis, S., Dagiene, V., & Beresnevičius, G. (2016). Learning Programming through Games and Contests:. *Overview, Characterisation and Discussion*.

Crawford, C. (1982). *The Art of Computer Game Design*.

*Gaming: Definition*. (n.d.). Retrieved from TechTarget: https://www.techtarget.com/whatis/definition/gaming

*Onboarding - Puzzle discussion*. (2022). Retrieved from CodinGame: https://www.codingame.com/forum/t/onboarding-puzzle-discussion/119

*Onboarding*. (2022). Retrieved from CodinGame: https://www.codingame.com/ide/puzzle/onboarding

*Screen developers with coding assessments*. (2022). Retrieved from CodinGame: https://www.codingame.com/work/

*Sports games*. (2022). Retrieved from Steam: https://store.steampowered.com/app/1811260/EA\_SPORTS\_FIFA\_23/

*TOP 5 coding games to learn programming*. (2022). Retrieved from Comarch: https://career.comarch.com/blog/top-5-coding-games-to-learn-programming/

*What is programming?* (n.d.). Retrieved from Javatpoint: https://www.javatpoint.com/what-is-programming#:~:text=Programming%20is%20an%20exercise%20or,the%20form%20of%20computer%20language.

Yang, K.-H., & Chen, H.-H. (2021). Interactive Learning Environments. *What increases learning retention: employing the prediction-observation-explanation learning strategy in digital game-based learning*.