

**CENTRAL ASIAN UNIVERISITY**

**CENTRAL ASIAN INSTITUTE OF TECHNOLOGY**

**CENTER OF INFORMATION TECHNOLOGY AND SCIENTIFIC COMPUTING**

**DEPARTMENT OF COMPUTER SCIENCE**

**AI BOT FOR A 2D GAME**

Project Proposal

**PREPARED BY:**

Sukhrob Sotiboldiev

**ADVISORS**: My Coursemates

**DECEMBER 2023**

**ACKNOWLEDGMENT**

I would like to thank My Coursemates for giving me the chance to work on an engineered project and urging me to give up our unscientific method of programming.

I also would like to thank our adminstration for providing me with internet; it helped me get sprites and tutorials to make this document possible.

**TABLE OF CONTENTS**

Contents

[**ACRONYMS** v](#_Toc510844554)

[**ABSTRACT** vi](#_Toc510844555)

[**1.** **INTRODUCTION** 1](#_Toc510844556)

[**1.1** **Background** 1](#_Toc510844557)

[**1.2** **The Existing System** 2](#_Toc510844558)

[**1.3** **Statement of the Problem** 2](#_Toc510844559)

[**1.4** **Objective of the Project** 3](#_Toc510844560)

[**1.4.1** **General Objective** 3](#_Toc510844561)

[**1.4.2** **Specific Objective** 3](#_Toc510844562)

[**1.5** **Proposed System** 3](#_Toc510844563)

[**1.6** **Feasibility Study** 4](#_Toc510844564)

[**1.6.1.Economic Feasibility** 4](#_Toc510844565)

[**1.6.1.1.Developmental cost** 4](#_Toc510844566)

[**1.6.1.2.Operational Cost** 5](#_Toc510844567)

[**1.6.2.Technical Feasibility** 5](#_Toc510844568)

[**1.6.3.Schedule Feasibility** 5](#_Toc510844569)

[**1.7** **Scope** 5](#_Toc510844570)

[**1.8** **Methodology** 6](#_Toc510844571)

[**1.8.1.Data Collection Methods** 6](#_Toc510844572)

[**1.8.2.Description of Data** 6](#_Toc510844573)

[**1.8.3.Implementation** 6](#_Toc510844574)

[**1.9** **Project Management plan** 7](#_Toc510844575)

[**1.9.1.Time Management plan** 7](#_Toc510844576)

[**1.9.2.Quality Management plan** 7](#_Toc510844577)

[**1.9.3.Communication Management plan** 8](#_Toc510844578)

[**APPENDIX** ix](#_Toc510844579)

[**REFERENCE** x](#_Toc510844580)

**LIST OF TABLES**

[**Figure 1: welcome page……………………………………………………………………………………….4**](#_Feasibility_Study)

[**Table 1: developmental coast……………………..…………………………………………………...…….4**](#_Developmental_cost)

[**Gant Chart**](#_Project_Management_plan) **7**

**[Table 2: management plan……………………………………..…………………………………………….8](#_Communication_Management_plan)**

# **ACRONYMS**

**AI** :: artificial intelligence

**Sci-fi** :: science fiction

**PC** :: personal computer

**LISP** :: LISt processing

# **ABSTRACT**

Artificial Intelligence (AI) is the hottest issue in the tech industry right now. From a primitive AI that plays breakout games to a fully automated AI robot that gained citizenship in Saudi Arabia, which is named Sofia, AI has come a long way. This begs the question; what is the progress of Ethiopian Artificial Intelligence.

Apparently progress in this area is little to none. As one might expect the Ethiopian society may not be in favor of artificial intelligence due to our heavily deep-rooted superstitious beliefs, and Hollywood hasn’t done much to advocate an opposition to the fear of AI.

Many Western countries are making good work use of this technology. AI is learning sophisticated surgery procedures, algorithms, mathematical mysteries to solve them on humanly unachievable levels.

So, this project aims to put the stepping stone to the Holy Grail that is AI. Almost everyone has played against a computer controlled bot (note, **not** an *AI* bot rather a *hard-coded* one), either on their phones or their computers. If you ask them “would prefer a more competent bot that learns your moves”? Most would say “yes, who would not want a healthy competition”. This stage of the project will just be the beginning, we hope to make it viable for different fields.

# **INTRODUCTION**

## **Background**

AI is a relatively new technology in the tech sector which has proved to be very valuable. It is basically a way for computerized machines to learn, it learns through its own mistakes like an infant. What makes this way of learning different from our way of learning is the time it takes and the precision, machines can learn in days what will take an average human years. So, we thought to capitalize on this technology while helping our country on the way.

The term artificial intelligence was coined in the mid 1950’s. Today it is defined as the “subfield of computer science concerned with the concepts and methods of symbolic inference by computer.” The first project utilizing AI is traced back to 1955 when Carnegie Mellon University researchers developed a computer program to work through proofs resulting in theorems. In the late 50’s, McCarthy developed a computer language Known as LISt Processing (LISP) This is the language used for most artificial intelligence projects. Decades later artificial intelligence is still only a small progression from its birth. Perhaps in the surge of AI optimism was at its best and humanity was oversimplified and underestimated.

The term game AI is used to refer to a broad set of [algorithms](https://en.wikipedia.org/wiki/Algorithm) that also include techniques from [control theory](https://en.wikipedia.org/wiki/Control_theory), [robotics](https://en.wikipedia.org/wiki/Robotics), [computer graphics](https://en.wikipedia.org/wiki/Computer_graphics) and [computer science](https://en.wikipedia.org/wiki/Computer_science) in general. One of the first examples of AI is the computerized game of [Nim](https://en.wikipedia.org/wiki/Nim) made in 1951 and published in 1952. Despite being advanced technology in the year it was made, 20 years before [Pong](https://en.wikipedia.org/wiki/Pong), the game took the form of a relatively small box and was able to regularly win games even against highly skilled players of the game. In 1951, using the [Ferranti Mark 1](https://en.wikipedia.org/wiki/Ferranti_Mark_1)machine of the [University of Manchester](https://en.wikipedia.org/wiki/University_of_Manchester), [Christopher Strachey](https://en.wikipedia.org/wiki/Christopher_Strachey) wrote a [checkers](https://en.wikipedia.org/wiki/Draughts) program and [Dietrich Prinz](https://en.wikipedia.org/wiki/Dietrich_Prinz) wrote one for [chess](https://en.wikipedia.org/wiki/Chess).

Games that featured a [single player](https://en.wikipedia.org/wiki/Single_player) mode with enemies started appearing in the 1970s. The first notable ones for the [arcade](https://en.wikipedia.org/wiki/Arcade_game) appeared in 1974: the [Taito](https://en.wikipedia.org/wiki/Taito_Corporation) game [*Speed Race*](https://en.wikipedia.org/wiki/Speed_Race) ([racing video game](https://en.wikipedia.org/wiki/Racing_video_game)) and the [Atari](https://en.wikipedia.org/wiki/Atari) games *[Qwak](https://en.wikipedia.org/wiki/Qwak!_(arcade_game)" \o "Qwak! (arcade game))* (duck hunting [light gun shooter](https://en.wikipedia.org/wiki/Light_gun_shooter)) and [*Pursuit*](https://en.wikipedia.org/wiki/Pursuit_(arcade_game)) ([fighter aircraft dogfighting simulator](https://en.wikipedia.org/wiki/Combat_flight_simulator)). Two text-based computer games from 1972, [*Hunt the Wumpus*](https://en.wikipedia.org/wiki/Hunt_the_Wumpus) and [*Star Trek*](https://en.wikipedia.org/wiki/Star_Trek_(script_game)), also had enemies. Enemy movement was based on stored patterns. The incorporation of [microprocessors](https://en.wikipedia.org/wiki/Microprocessor) would allow more computation and random elements overlaid into movement patterns.

The game industry in Ethiopia has been none existent in the past. Now though, some startups are beginning to emerge. To point out one of them Dawit Abraham comes to mind, who has a Bachelor of Science degree in Computer Engineering from an Ethiopia university, was of one of the trainees at *Gebeya* – an Ethiopian IT academy training startup which was founded in 2016 by Amadou Daffe, a Senegalese native software developer, and Hiruy Amanuel, an investor and self-made entrepreneur of Ethiopian origin who was then based in Silicon Valley, California, USA. He launched a 3D Game, Kukulu, which is the first of its kind to come out of Ethiopia.

The potential beneficiaries of this product can be a gamer looking for a challenge, a tech student looking to develop on an existing AI, a medical company that wants to develop an AI software that accurately diagnoses a patient and so on. The AI we develop will be built from the ground-up and for this stage we will deliver a primitive AI that plays a particular game that also we will be developing.

## **The Existing System**

The current system that is being employed while making bots for games (especially in Ethiopia) is; by hard-coding all the possible scenarios that the game will play out. Most of the time single player game developers predict a path that a player of their game will use and throw the challenging counterparts, it might be a bot or an obstacle of some sorts, through out that direction.

There is the *checkers* program, developed in the middle 50s and early 60s, that eventually achieved sufficient skill to challenge a respectable amateur. Two text-based computer games from 1972, *Hunt the Wumpus* and *Star Trek*, also had enemies. Enemy movement was based on stored patterns. The incorporation of microprocessors would allow more computation and random elements overlaid into movement patterns. Games like *Madden Football*, *Earl Weaver Baseball* and *Tony La Russa Baseball* all based their AI on an attempt to duplicate on the computer the coaching or managerial style of the selected celebrity. Madden, Weaver and La Russa all did extensive work with these game development teams to maximize the accuracy of the games. Later sports titles allowed users to "tune" variables in the AI to produce a player-defined managerial or coaching strategy

In Ethiopia even for systems that require automation or an AI of some sort, we insist on using a manual system. Large scale factories, industry zones, security systems, traffic systems are still using manual systems in Ethiopia. Computer-controlled bots may play against other bots and/or human players in unison, either over the Internet, on a LAN or in a local session. Features and intelligence of bots may vary greatly, especially with community created content. Advanced bots feature machine learning for dynamic learning of patterns of the opponent as well as dynamic learning of previously unknown maps – whereas more trivial bots may rely completely on lists of waypoints created for each map by the developer, limiting the bot to play only maps with said waypoints.

## **Statement of the Problem**

The nature of the problem in its true self expresses a lack of capitalization upon resources that stem from an entity, which in the non-existent nature it is in, deprives the community of a market.

The market being a gaming industry should not be taken so lightly considering the fact that it ranks very high in the entertainment industry. All things considered most problems either stem from a lack of money or know how, and taking into account the fact that Ethiopia suffers deep and extensively from a lack of the former, one should see all too clearly that Ethiopia’s biggest problem is rest assured not a lack of efficient systems that aid in the day to day life or even proficiency in public systems, be it health care, transportation or education. The emergence of efficiency driven thinking was incentivized as result of irresponsible management of resources, which in all fairness to a certain degree it was, but rather a lack of resources that brought forth a need of efficiency.

In words summed up, the problem is a lack of capitalization in an extremely lucrative gaming industry.

## **Objective of the Project**

### **General Objective**

The objective of this project is to develop a system that entertains people by surprising them with new experiences against a learning AI bot.

### **Specific Objective**

The general objective attainability depends on the specific objectives written throughout this section. First there should be a game and it should be fun, well fun is a hard concept to grasp but we will try to attain it using the methods described in section 1.9.2 (Quality Management) and by gathering requirements that points us to, how to make the game fun.

The user is expected to be faced with a competent game so, I will develop an AI bot that has been tested on test users. There is also the factor of boredom that shall be solved with the AI bot.

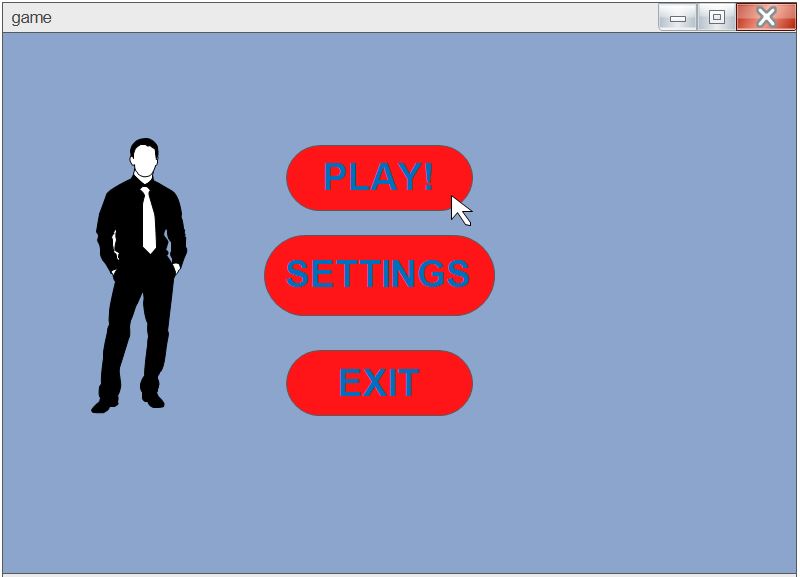
## **Proposed System**

The main target of this software design is to build an AI that can adapt each move made by the player and studies it along the gaming session. Basically the AI is a smarter version of the user; since it studies the user’s moves, it will be able to counter attack in the ways that you unintentionally taught it earlier.

Hence, the next time the player returns to play the game the user has to be smart enough and come up with a new strategy in order to beat the AI or else will end up losing to an AI which picked up every knowledge of the one’s gaming ways through each session as the player faced up against it. Therefore, in this manner the player will be able to discover new gaming styles and strategies that will assist him/her on defeating the AI.

Last but not least and the most important application of this software is; with the process of tackling the AI in each session is that the player’s gaming skill will be far enhanced by this way than that of playing the game for quiet several times over a long period of time. As mentioned earlier, due to the adaption of the player’s moves it’s easier to correct one’s mistakes and not make any blunders in the future; since they are originally the player’s own moves. Thus, the help of this AI improving one’s skills with it being exceptionally effective and less tiring from the wonted method of playing the game again and again is for sure compared to none.

Below is a rough estimation of the game’s welcome screen, it is not exactly what it will look like but just to give the user a visualization of the proposed system.



**Figure 1: welcome page**

## **Feasibility Study**

## **Economic Feasibility**

### **Developmental cost**

While developing this project I am going to use free software and I use our own computers to develop my game application by ourselves with the help of my advisor although, I have some developmental costs and they are described in the figure below.

|  |  |
| --- | --- |
| Type | Cost in birr |
| Mobile card | 50.00 |
| Transportation | 70.00 |
| For launch and other food stuff | 200.00 |
| Printing and Binding | 20.00 |
| Total | 340.00 |

**Table 1: development coast**

### **Operational Cost**

I am expecting for few expenses to promote the game in order to popularize it for the society. I believe that the game will continue serving the interest of game players.

## **Technical Feasibility**

The technical requirement of the system regarding a user is very flexible. Of course the user is expected to be at list a novice for gaming environments and a computer in general, although this requirement can easily be full filled by taking a short tutorial of the game. A user more enthusiastic about games and AI bots would be preferred as this will make the experience deserving and beneficial but this is not a demand. The developers of this system are relatively novice but willing to put the work in the work required. It goes without saying that the user should have an access to a personal computer (maybe not personal, just any computer).

## **Schedule Feasibility**

The overall time we have allocated for the project is around three months. I believe we will have a sufficient amount of time to achieve our goals prior to the deadline. However, the learning curves and the difficulty of the project might affect the Schedule Feasibility.

## **Scope**

The system is developed for gamers but anyone who can use a computer can use this system, the game is 2d plat-former. The game will feature a single level with an AI incorporated to it.

The game will has features like:

* Setting to change audio mode,
* setting to change key bindings,
* setting to change difficulty level,
* welcome screen,
* pause capabilities and
* a dedicated exit button.

## **Methodology**

## **Data Collection Methods**

The data required to build the Gamebot AI will be entirely collected from past gaming session with the player (we intend to have at least 10 sessions with different players before finalizing). So as to implement a full-fledged AI we will not hardcode any escape functions for the AI nor will the AI rely on coded methods to play the game, as the data collected from every player will suffice.

## **Description of Data**

The system aims at creating an advancing gameplay interactive experience with a learning AI gamebot. The data collected will compromise of previous gaming sessions and the data created by the AI bot so as to counter certain gaming patterns.

## **Implementation**

The system that is going to be developed is a game, the difficult part about developing a game is requirement gathering. Meaning users are intimidated by details and don’t know what they want until they see it. This makes most methods of development impossible to use. So, if the user is unsatisfied with the system the developers should be able to amend other possibilities.

The users of the system want to visualize what work is done on the go, because the system is a game there’s no question that most end users have some difficulty putting elements together with just a written requirements to arrive at a good picture of what they will be getting. Meaning the developers need to have an executable (not fully functional and done) to show the user at each stage.

The system should welcome some changes it should be flexible, being restricted means the system will be useless and no one will use it. The developers are not large in numbers but are very dedicated and scope isn’t well known either.

By analyzing the facts stated above most methods of development are eliminated (like waterfall and staged delivery). The only method that will work for this system is ***agile development***, its iterative behavior means that we can deliver some part of the game to the user at every stage to get feedbacks and work for the best end system. The requirements are going to stay flexible enough (but not too much), therefore customer satisfaction is more likely attainable.

## **Project Management plan**

## **Time Management plan**



**Chart 1: Gant chart**

## **Quality Management plan**

The earlier problems can be found, the better. It is therefore very important to verify quality as soon as possible. The system will be tested for functional and quality problems throughout the entire project. I will test the game for emotional responses from test subjects while developing it as this will result in a more interactive and immersive system at the end.

I want to test the system every stage for best quality. So, I will develop it iteratively meaning we will be splitting up the development of the software into several miniature projects, called iterations. Each iteration continuously delivers a working executable that allows for proper assessment of current project quality and progress. This flow repeats not only on a per-iteration basis but also on a project basis; the focus of the earliest iterations being more towards nailing down the fun of the game, often working in conjunction with focus groups and requirements, while later iterations focus more on detailing features of the product.

The system requirements are going to be kept under control, because overgrown requirements and features will likely lead to an abrupt fail to deliver the stated system. There is also the “the fun factor” which is hard for game designers to accommodate, we will try to solve this by playing the game or a rough version of it at every stage. I propose that since testing is such an important part of development, every developer should be involved in testing and quality assurance. The requirements form the basis of different test cases to be executed, and when writing requirements, test cases will be written in parallel. This will both raise the quality of the requirements and also make testing activities a more integrated part in development. Care should also be taken not to duplicate information between requirements and test.

Best practices of software engineering are observed approaches to successfully delivering software products. A software process for the game development will therefore have the best practices of general software engineering as a base, and this base will then be tailored into a development process specifically aimed at the game development. The process will also support further refinement on corporate and even inter-project levels.

## **Communication Management plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of communication | Objective of communication | frequency | Method of communication | Members |
| Proposal development | Discussing on how to Gather information on the project | 2 days a week  For 2 hour | Face to Face | The Project team |
| Project team meeting | Review the status of the project with the team | As Needed | Face to Face | The Project team |
| Final project status report | Finalizing Reports of status of the project | 8/23/2010 | Email | The Project team |

**Table 2: management plan**

# **APPENDIX**

The questions for the interview were:

* What is your experience with game bots? Are they challenging or straight up dumb?
* Are you satisfied with the currently existing game bots?
* Would you prefer a more challenging gamebot?

# **REFERENCE**

Wikipedia: [**https://en.wikipedia.org/wiki/Artificial\_intelligence**](https://en.wikipedia.org/wiki/Artificial_intelligence) On March 27 2018

Wikipedia: [**https://en.wikipedia.org/wiki/Video\_game\_bot**](https://en.wikipedia.org/wiki/Video_game_bot) On March 27 2018