UNIVERSIDADE FEDERAL DO TOCANTINS
CÂMPUS UNIVERSITÁRIO DE PALMAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DESENVOLVIMENTO DE UM AGENTE INTELIGENTE PARA O MMORPG TIBIA

THIAGO CASTANHEIRA RETES

PALMAS (TO)
2021

THIAGO CASTANHEIRA RETES

DESENVOLVIMENTO DE UM AGENTE INTELIGENTE PARA O MMORPG TIBIA

Trabalho de Conclusão de Curso II apresentado à Universidade Federal do Tocantins para obtenção do título de Bacharel em Ciência da Computação, sob a orientação do(a) Prof.(a) Dr. Rafael Lima de Carvalho.

Orientador: Dr. Rafael Lima de Carvalho

PALMAS (TO)

2021

THIAGO CASTANHEIRA RETES

DESENVOLVIMENTO DE UM AGENTE INTELIGENTE PARA O MMORPG TIBIA

Trabalho de Conclusão de Curso II apresentado à UFT – Universidade Federal do Tocantins – Câmpus Universitário de Palmas, Curso de Ciência da Computação foi avaliado para a obtenção do título de Bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de aprovação: 10 / 8 / 2021

Banca Examinadora:

Prof. Dr. Rafael Lima de Carvalho

Prof. Dr. Alexandre Tadeu Rossini

Prof. Dra. Anna Paula Rodrigues

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

R437d Retes, Thiago.
Desenvolvimento de um agente inteligente para o MMORPG Tibia. / Thiago Retes. – Palmas, TO, 2021.
8 f.

Artigo de Graduação - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Ciências da Computação, 2021.
Orientador: Rafael Lima de Carvalho

1. Bots automatizados. 2. TIBIA. 3. Inteligência Artificial. 4. Visão Computacional. I. Título

CDD 004

AGRADECIMENTOS

Gostaria de agradecer a minha querida avó que sempre me incentivou a ir além, a minha família que me apoiou durante toda esta trajetória e ao meu orientador que sempre acreditou no meu potencial e foi imprescindível para a conclusão deste ciclo.

RESUMO

A Inteligência Artificial sempre foi usada na concepção de agentes automatizados para jogos como Xadrez, Go, Defense of the Ancients 2, Snake Game, bilhar e muitos outros. Neste trabalho, apresentamos o desenvolvimento e avaliação de desempenho de um agente reativo para o jogo RPG Tibia. O agente inteligente é construído usando uma combinação de técnicas de IA, como o algoritmo de busca de grafos A* e ferramentas de visão computacional, como a correspondência de modelos. Usando quatro algoritmos para obter a posição global do jogador no jogo, lidar com sua saúde e mana, atacar monstros e caminhar pelo jogo, conseguimos desenvolver um agente de Tibia totalmente automatizado baseado em imagem de entrada bruta. Avaliamos o desempenho do agente em três cenários distintos fazendo dez sessões de quinze minutos e cinco sessões de uma hora, coletando e analisando métricas como Ganho de XP, Uso de Suprimentos e Balanço. Os resultados da simulação mostram que o agente desenvolvido é capaz de jogar o jogo de forma consistente de acordo com as métricas do jogo.

Palavra-chave: Bots automatizados. TIBIA. Inteligência Artificial. Visão Computacional.

ABSTRACT

Artificial Intelligence has always been used in designing of automated agents for playing games such as Chess, Go, Defense ofthe Ancients 2, Snake Game, billiard and many others. In this work, we present the development and performance evaluation of a reactive agent for the RPG Game Tibia. The intelligent agent is built using a combination of AI techniques such as graph search algorithm A* and computer vision tools like template matching. Using four algorithms to get global position of player in game, handle its health and mana, target monsters and walk through the game, we managed to develop a fully automated Tibia agent based in raw input image. We evaluated the performance of the agent in three different scenarios doing ten sessions of fifteen minutes and five sessions of one hour, collecting and analyzing metrics such as XP Gain, Supplies Usage and Balance. The simulation results show that the developed agent is able to play the game consistently according to in-game metrics.

Keywords: Automated bots. TIBIA. Artificial Intelligence. Computer Vision.

SUMÁRIO

1 INTRODUÇÃO

Os jogos são uma grande indústria que gerou $22,4 bilhões em vendas em 2014, de acordo com a Entertainment Software Association (CANO, 2016). Em um MMORPG(Massively Multiplayer Online Role-Playing Game), milhares de jogadores podem jogar juntos em um mundo de jogo persistente (WANG, 2017). De acordo com Cano (CANO, 2016), ”das dezenas de milhões de jogadores que jogam diariamente, 20 por cento jogam RPGs multijogador online (MMORPGs)”. Dentro das plataformas MMORPG, muitos jogadores usam para negociar bens virtuais dentro das economias prósperas do jogo.

De acordo com Ferreira et al. (FERREIRA; TROVO; NESTERIUK, 2017), Tibia é um Massive Multiplayer Online Role-Playing Game (MMORPG), desenvolvido pela CipSoft. Tibia também é um dos jogos mais antigos do gênero (1997), com uma comunidade que ultrapassa 500.000 jogadores, dos quais 40% deles são brasileiros.

A Inteligência Artificial possui vários campos de atuação ao lidar com jogos. Depois de muitos esforços de pesquisa para chegar a uma IA capaz de jogar contra um ser humano, em 1997 o computador Deep Blue, fabricado pela IBM, foi capaz de derrotar o campeão mundial de xadrez Garry Kasparov (SCHAEFFER; MüLLER; KISHIMOTO, 2014). Outro marco aconteceu em 2009, quando uma IA chamada Fuego venceu Chou Chun-hsun, o campeão mundial em GO, um jogo de tabuleiro desafiador muito mais complexo do que o xadrez em várias possibilidades (embora estivesse usando um tabuleiro menor do que o oficial) (SCHAEFFER; MüLLER; KISHIMOTO, 2014). Além disso, em março de 2016, o AlphaGo AI do Google derrotou Lee Sedol, vencedor de 18 títulos internacionais, em março de 2016 (SILVER et al., 2017).

Procurando pesquisar abordagens para resolver o jogo Snake e construir um Bot com IA que o resolva em um número mínimo de etapas, Sharma et al.(SHARMA et al., 2019), comparou Best First Search, A* Search, A* Search com forward check, Random Move e Almighty Move de acordo com suas características e propõe um bot misturando-os, após a avaliação deles, eles propuseram um bot que usou Best First Search nas primeiras 4 frutas, então A* com Forward Checking para as próximas 34 frutas e então Almighty Move para as próximas 62 frutas, o que resultou em um Bot que pode resolver o Snake Game em 100 iterações, que supera os outros algoritmos sozinhos. Os autores propõem na sessão de discussão um jogo diferente para treinar jogadores reais usando uma cobra de sombra controlada por seu Bot em que o jogador tem que imitar seus movimentos para ganhar pontos.

Olhando para o campo dos jogos eletrônicos, os programadores usam a IA principalmente para modelar bots automatizados que compõem o comportamento de inimigos e também de assistentes de jogo. Por outro lado, existe um campo de atuação na IA que tenta criar um agente para jogar no lugar de um jogador humano em jogos eletrônicos.

Este tipo de software é conhecido como bots de jogos (KANG et al., 2016). Portanto, os bots de jogos geralmente usam IA para jogar os jogos automaticamente e esta função é considerada um grande desafio e um benchmark computacional complexo.

Para o desenvolvimento de tais bots de jogos, Dharmawan e Hanafiah (DHARMAWAN; HANAFIAH, 2021) discutem a usabilidade de um bot de cliques baseado na correspondência de modelos para jogos gacha que são jogos com chances em modelos de negócios freemium, esses tipos de jogos compartilham um característica com MMORPGs que são a necessidade comum de trabalhar para adquirir recursos, em que os bots economizam muito tempo e esforço do usuário ao automatizar aquisição de recursos. Em seu trabalho, o autor prefere apresentar os resultados de acordo com o desempenho computacional e a precisão das funções implementadas no bot de cliques, neste trabalho fizemos uma abordagem diferente, avaliando o desempenho do agente de acordo com algumas medidas dadas pelo próprio MMORPG .

Com relação à busca de caminhos em videogames, Cui e Shi(CUI; SHI, 2011) discutem sobre o algoritmo A* e suas otimizações relacionadas Hierachical Pathfinding A* (HPA*) e Navigation Mesh (NavMesh). Além disso, os autores descrevem otimizações para a função heurística usada pelo A* usando uma função de superestimação a fim de trocar o caminho mais curto por uma busca mais rápida, depois descreve maneiras de otimizar o uso de memória do A* com abordagens como alocar um banco de nós mínimo na memória e usando o Iterative Deepening A* (IDA*). Os autores também sugerem o uso de tabelas hash para a lista de nós fechados e um heap binário para a lista de nós abertos, a fim de otimizar as estruturas de dados usadas pelo A*. Cui e Shi(CUI; SHI, 2011) também discutem aplicações relevantes na indústria de jogos apresentando como títulos bem conhecidos como Age of Empires II, Civilization V e World of Warcraft lidam com os problemas de busca de caminhos.

A fim de desenvolver um sistema de navegação autônomo para um robô, Ayala-Raggi et al.(AYALA-RAGGI et al., 2015) usou duas abordagens baseadas em visualização, uma usando SURF, RANSAC e Análise de proscutas e outra utilizando template matching com Normalized Cross Correlation (NCC), ambas utilizando uma imagem panorâmica composta por 3 imagens da câmera em um robô que cobre 76º x 3 = 228º. Os principais resultados indicaram que a abordagem SURF foi mais consistente no que diz respeito à adição de obstáculos nos quais a correspondência de modelos NCC falhou.

Desde a publicação de Mnih et al.(MNIH et al., 2015), que usou imagem bruta da tela como entrada para alimentar um modelo de aprendizado profundo para jogar 49 jogos Atari e alcançou níveis super-humanos em 29 deles, a comunidade acadêmica, nos problemas em que se adotam imagem bruta da tela como entrada, focou em aplicar técnicas de aprendizado de máquina. Por outro lado, neste artigo tentamos trazer outra luz a estes problemas usando sistemas baseados em conhecimento com técnicas de IA e Visão Computacional para jogar um jogo complexo usando imagem bruta como entrada para o agente proposto, mas não utilizando técnicas de alto custo computacional como aprendizagem profunda.

O objetivo principal deste trabalho é apresentar o desenvolvimento de um agente reativo, baseado em inteligência artificial e processamento de imagens, que seja capaz de jogar Tibia usando informações visuais e gerando entradas comuns. O agente proposto é então avaliado em três cenários diferentes e as métricas do jogo são coletadas a fim de medir seu desempenho durante o jogo.

REFERÊNCIAS

AYALA-RAGGI, S. E. et al. A simple view-based software architecture for an autonomous robot navigation system. In: KAMEL, M.; CAMPILHO, A. (Ed.). **Image Analysis and Recognition**. Cham: Springer International Publishing, 2015. p. 287–296. ISBN 978-3-319-20801-5.

CANO, N. **Game hacking: developing autonomous bots for online games**. [S.l.]: No Starch Press, 2016.

CUI, X.; SHI, H. A*-based pathfinding in modern computer games. **International Journal of Computer Science and Network Security**, International Journal of Computer Science and Network Security (IJCSNS), v. 11, n. 1, p. 125–130, 2011.

DHARMAWAN, T.; HANAFIAH, N. Clicker bot for gacha games using image recognition. **Procedia Computer Science**, v. 179, p. 598–605, 2021. ISSN 1877-0509. 5th International Conference on Computer Science and Computational Intelligence 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921000521>.

FERREIRA, N.; TROVO, P.; NESTERIUK, S. Emergence in game design: Theoretical aspects and project's potentialities. In: SPRINGER. **International Conference on Distributed, Ambient, and Pervasive Interactions**. [S.l.], 2017. p. 597–611.

KANG, A. et al. Multimodal game bot detection using user behavioral characteristics. **SpringerPlus**, v. 5, 2016.

MNIH, V. et al. Human-level control through deep reinforcement learning. **Nature**, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 518, n. 7540, p. 529–533, fev. 2015. ISSN 00280836. Disponível em: <http://dx.doi.org/10.1038/nature14236>.

SCHAEFFER, J.; MüLLER, M.; KISHIMOTO, A. Go-bot, go. **IEEE Spectrum**, v. 51, n. 7, p. 48–53, 2014. Cited By 1.

SHARMA, S. et al. Solving the classic snake game using ai. In: **2019 IEEE Pune Section International Conference (PuneCon)**. [S.l.: s.n.], 2019. p. 1–4.

SILVER, D. et al. Mastering the game of go without human knowledge. **nature**, Nature Publishing Group, v. 550, n. 7676, p. 354–359, 2017.

WANG, J. Y. **Combat State-Aware Interest Management for Massively Multiplayer Online Games**. Dissertação (Mestrado) — TECHNISCHE UNIVERSITÄT MÜNCHEN, 2017.

A ARTIGO PUBLICADO

Thiago Castanheira Retes, Rafael Lima de Carvalho. Development of an intelligent agent for Tibia MMORPG, **Academic Journal on Computing, Engineering and Applied Mathematics**

ACADEMIC JOURNAL ON COMPUTING, ENGINEERING AND APPLIED MATHEMATICS, VOL. 2, NO. 2, August 2021

Development of an intelligent agent for Tibia MMORPG

Thiago Castanheira Retes[1] and Rafael Lima de Carvalho[1]

[1] *Federal University of Tocantins, Computer Science Department, Tocantins, Brazil*

Reception date of the manuscript: 26/06/2021
Acceptance date of the manuscript: 05/08/2021
Publication date: 11/08/2021

Abstract— Artificial Intelligence has always been used in designing of automated agents for playing games such as *Chess*, *Go*, *Defense of the Ancients 2*, *Snake Game*, *billiard* and many others. In this work, we present the development and performance evaluation of a reactive agent for the RPG Game Tibia. The intelligent agent is built using a combination of AI techniques such as graph search algorithm A* and computer vision tools like template matching. Using four algorithms to get global position of player in game, handle its health and mana, target monsters and walk through the game, we managed to develop a fully automated Tibia agent based in raw input image. We evaluated the performance of the agent in three different scenarios doing ten sessions of fifteen minutes and five sessions of one hour, collecting and analyzing metrics such as *XP Gain*, *Supplies Usage* and *Balance*. The simulation results show that the developed agent is able to play the game consistently according to in-game metrics.

Keywords— Automated bots, TIBIA, Artificial Intelligence, Computer Vision

I. INTRODUCTION

Gaming is a huge industry that generated $22.4 billion in sales in 2014, according to the Entertainment Software Association [1]. In a Massively Multiplayer Online Role-Playing Game (MMORPG), thousands of players can play together inside a persistent game world [2]. According to Cano[1], "of the tens of millions of players who play games daily, 20 percent play massively multiplayer online role-playing games (MMORPGs)". Inside the the MMORPG platforms, a lot of players use to trade virtual goods within thriving in-game economies.

According to Ferreira et al.[3], Tibia is a Massive Multiplayer Online Role-Playing Game (MMORPG), developed by CipSoft. Tibia is also one of the oldest games in the genre (1997), with a community that surpasses 500.000 players, for which 40% of these are Brazilian.

Artificial Intelligence has many fields of actuation when dealing with games. After many research efforts in order to come up with an AI that is able to play against a human being, in 1997 the Deep Blue computer, made by IBM, was able to beat the world chess champion Garry Kasparov[4]. Another milestone happened in 2009 when an AI named Fuego beat Chou Chun-hsun, the world champion in GO, a challenging board game far more complex than chess in number of possibilities (even though it was using a small board than the official one)[4] . Moreover, in March 2016 Google's AlphaGo AI, defeated Lee Sedol, the winner of 18 international titles, in March 2016 [5].

Looking to survey approaches of solving Snake Game and build an AI Bot that solves it in minimal amount of steps, Sharma et al. [6], compared Best First Search, A* Search, A* Search with forward checking, Random Move and Almighty Move according to its traits and proposes an AI bot mixing them, after the evaluation of them, they proposed an AI bot that used Best First Search in the first 4 fruits, then A* with Forward Checking for the next 34 fruits and then Almighty Move for the next 62 fruits, which resulted in an AI Bot that can solve Snake Game in 100 iterations, which outperforms the other algorithms alone. The authors propose in the discussion session a different game to train normal players using a shadow snake controlled by their AI Bot which the player has to mimic their movements to earn score.

Looking at the field of electronic games, the programmers use AI mostly to model automated bots that compose the behavior of enemies as well as game assistants. On the other hand, there is a field of actuation in AI that tries to create an agent to play on behalf of a human player in electronic games. These piece of software are known as *game bots*[7]. Therefore, game bots usually uses AI in order to play the games automatically and this function is considered a great challenge and a complex computational benchmark.

Towards the construction of such game bots, Dharmawan and Hanafiah [8] discuss the usability of a clicker bot based in template matching for *gacha games* which are games with chances in freemium business models, these kind of games share a trait with MMORPGs which are the common need to farm lots of resources, that the bots saves the user lots of time and effort by automating the farming. In its work, the author prefer to present the results according to the computational

Contact data: Thiago C. Retes, thiagoretes2@gmail.com

performance and accuracy of the functions implemented in the clicker bot, in this paper we took a different approach, evaluating the performance of agent according to some measures given by the MMORPG itself.

Regarding pathfinding in video games, Cui and Shi[9] discuss about A* algorithm and its related optimizations Hierachical Pathfinding A*(HPA*) and Navigation Mesh(NavMesh). Moreover, the authors describe optimizations to the heuristic function used by A* by using an overestimating function in order to trade shortest path for a faster search, then it describes ways to optimize memory usage of A* with approaches such as allocating a minimum node bank in memory and using Iterative Deepening A*(IDA*). The authors also suggest using hash tables for the list of closed nodes and a binary heap for the list of open nodes in order to optimize the data structures used by A*. Cui and Shi[9] also discuss relevant applications in the game industry presenting how well known titles such as Age of Empires II, Civilization V, and World of Warcraft handle the pathfinding problems.

In order to develop an autonomous navigation system for a robot, Ayala-Raggi *et al.* [10] used two view-based approaches, one using SURF, RANSAC and Proscutes analysis and another one using template matching with Normalized Cross Correlation(NCC), both using a panoramic image composed of 3 images of the camera in a robot that covers 76º x 3 = 228º. The main results indicated that the SURF approach was more consistent in regard to the addition of obstacles which NCC template matching failed.

Since publication of Mnih et al.[11], that used raw input of screen to feed a deep learning model for playing 49 Atari games and achieved super-human levels in 29 of them, academic community mainly focused in solving problems adopting raw image input by applying machine learning techniques. On the other hand, in this paper we try to bring another light in these problems by using knowledge-based systems with AI and Computer Vision techniques to play a complex game using raw image as the input for the proposed agent, but not using expensive computational techniques such as deep learning.

The main objective of this work is to present the development of a reactive agent, based on artificial intelligence and image processing, that is capable of playing Tibia using visual information and generating common inputs. The proposed agent is then evaluated in three different scenarios and the game metrics are collected in order to measure its performance during the game play.

The present work is organized as follows: Section II presents the necessary background with the two main techniques used to develop the intelligent agent, such as A* algorithm and Template Matching for dealing with the raw image input. Section III explains the game Tibia, such as the map size, interface and mechanics. Section IV presents the methodology used to create the agent and evaluate it. Section V presents the simulations we did for each scenario presented in Section IV the results and our interpretation of the results. Section VI presents our conclusions about the work and our possible future developments.

II. Background

Since the proposed intelligent agent is based in two main techniques, in Subsection a it is shown the basic background of A* search algorithm. In Subsection b shows the background knowledge about Template Matching, the computer vision algorithm employed as well.

a. A Algorithm*

The A* algorithm is a path-searching and graph traversal algorithm, which is usually used due to its completeness and optimal efficiency[12]. It can be seen as an extension to Dijkstra's algorithm. Its main difference to Dijkstra's algorithm is the cost function which is: $f(x) = g(x) + h(x)$ where $g(x)$ is the cost to reach current node and $h(x)$ can be a graph of costs or an heuristic function that estimates the cost to the goal node.

For its optimality it is necessary that the heuristic function $h(x)$ meets certain requirements. The first condition is that $h(x)$ has to be an admissible heuristic. An admissible heuristic means that it never overestimates the cost to reach the goal. Since $g(x)$ is the cost of the actual state, then $f(x) = g(x) + h(x)$ never overestimates the cost along of each path[12].

Another requirement is that the heuristic function $h(x)$ has to be consistent. Consistency means for every state x and every successor x', the cost of reaching $h(x)$ has to be lower than the cost of the action from x to x' plus $h(x')$, which can be seen in Equation 1 [12]:

$$h(x) <= c(x, a, x') + h(x') \tag{1}$$

According to Hart et al.[13], given a node s, a set T of goal nodes and the successor operator Γ, the A* search algorithm can be described as:

1. Mark s as open and calculate $f(s)$
2. Select the open node n whose value of $f(n)$ is the smallest, resolving ties arbitrarily, but prioritizing any node $n \in T$
3. If $n \in T$, mark n as closed and terminate the algorithm.
4. Otherwise, mark n as closed, apply the successor operator Γ to n. Calculate $f(n')$ for each successor of n' and mark as open, every successor that has not already been marked as closed yet. Remark as open any closed node n' which is a successor of n and for which $f(n')$ is smaller now than it was when n' was marked as closed. Go to Step 2.

b. Template Matching

Template matching is a method used in digital image processing in order to find a small image in a larger one. It can be implemented by simply sliding the small image window in the larger image as a 2D convolution and comparing patch of larger image under the template one. Several comparison methods can be used, such as Cross Correlation or Square Difference.

The problem of template matching has been studied extensively in the field of computer vision for years. Many approaches were applied in almost all tasks of computer vision such as stereovision, camera calibration, recognition of objects, etc[14].

In Brunelli[15], the author presents the simplest template matching technique used in computer vision known as planar distribution of light. This technique takes the intensity values and transform them into a vector x, that can be compared, in a coordinate-wise fashion. This produces a spatially congruent light distribution represented in a similar form by vector y. The formula is given by Equations 2 and 3:

$$d(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 = \frac{1}{N}||x-y||_2^2 \quad (2)$$

$$s(x,y) = \frac{1}{1+d(x,y)} \quad (3)$$

The lower the value of $d(x,y)$ or the higher the value of $s(x,y)$, the better in indication of pattern similarity.

Since $s(x,y)$ is the comparison function, there are numerous functions that can be used in its place. The library we used in our intelligent agent was OpenCV 4.2.0[16], and it implements three of them with their normalized variants. Our agent was coded to use the comparator named correlation coefficient normalized.

III. TIBIA

As in any MMORPG, the main idea of the game is to evolve the player's character through hunting. This action gives the player money and XP (indicator for experience), and allows it to participate in the game history through quests and wars.

So an intelligent agent is a tool that has been used for a long time by players to evolve their characters through endless hunts. The main idea behind an intelligent agent is to automate an action to evolve the player's character through simple algorithms. In this work the goal is to evaluate an agent throughout measures of efficiency in order to produce the maximum XP per hour on Tibia. The map of Tibia is composed of 2304x2048 SQM(Square Metre) and 14 floors which gives a total exploration size of 66.060.288 SQMs, whether all parts of the map is available to walk through.

As it can be seen in Fig. 1, the Graphics User Interface in Tibia is kept as simple as possible. At the upper right corner, there is the *minimap* which shows the player's position in the game and enables it to navigate through the world with a click. Immediately beneath, there is located the current equipped items. Below it, there are the health and mana bars, which indicates the current amount of life and mana of the character. At the center of the screen, there is the main game screen. It consists of a matrix of 15x11 SQM and shows the actions of the characters in the player vision. Beneath it, there is the hotkeys bar, which contains the shortcuts to spells, items and text. Immediately bellow it, there are the text channels which shows all the actions that happens during the game play. The rest of the interface is placeholder for containers and windows that show specific statistics.

Tibia is a MMORPG which every action has some kind of cooldown involved. In general, there are 4 (four) types of cooldowns that are important to consider when building an intelligent agent for Tibia:

- *Movement*: It is the cooldown between movement from an SQM to another. Every SQM has a type that determines the amount of cooldown in conjunction with the player level.

- *Attack Spell Group*: every attack spell adds some global attack spell cooldown when cast, in general has a cost of 2 (two) seconds from the last time the spells were used;

- *Healing Spell Group*: Every healing spell adds in general a 1 (one) second global healing spell cooldown;

- *General Item use*: Usually the player can use one item for every second with some exceptions like the potion of *mana* shield;

- *Basic Attack*: The basic attack is based in weapon equipped and has a cooldown of 2 seconds.

a. Vocations

Tibia has 4 vocations, each of them with unique traits and its advantages and disadvantages.

- *Knight*: is the vocation which excels at melee combat. It has the highest amount of health points gaining 15 health points, 25 oz of capacity and 5 mana points per level. It evolves faster in melee skills(club, axe and sword) and shielding skill. Its spells are mostly focused in melee combat and tanking damage. Its main damage attribute is based on the weapon it uses(some weapons can have elemental damage thus making Knights elemental damage dealers).

- *Paladin*: is the distance combat class. It has a balanced amount of health points and mana points gaining 10 health points, 20 oz of capacity and 15 mana points per level. It evolves faster in distance fighting and shielding skill. Its spells are mostly focused in distance combat and self regeneration.

- *Druid*: is the healing mage class. It has the least amount of health points and the most amount of mana points gaining 5 health points, 10 oz of capacity and 30 mana points per level. It evolves faster in magic level. Its spells are mostly focused in healing, although it has some nice combat spells with Terra and Ice attributes.

- *Sorcerer*: is the damage dealer mage class. It has the least amount of health points and most amount of mana points, exactly like druids, sorcerer gains 5 health points, 10 oz of capacity and 30 mana points per level. It evolves faster in magic level. Its spells are most focused in dealing damage with Fire and Energy attributes.

b. Skills

Each player in Tibia can evolve different skills during their gameplay, they represent their expertise with said type of action, these skills can be separated in 3 main types:

Fig. 1: Tibia Interface. (a) In Red: Minimap, (b) In Yellow: Equipped Items, (c) In Orange: Health and Mana Bars, (d) In Green: Main Game Screen, (e) In Blue: Shortcuts and Hotkeys, (f) In Violet: Text Channels

- *Melee skills*: They are 3 skills (club, axe, sword), responsible for the damage done using melee weapons and knight attack spells. Can be evolved by using basic attack while using a weapon of the skill type.
- *Distance skill*: It is the skill responsible for the damage done using distance weapons (bows and crossbows). Can be evolved by using basic attack while using a bow or a crossbow.
- *Magic Level*: It is the skill responsible for the power of spells in general(healing and attack spells). Can be evolved by spending mana points.

c. Combat

The action of hunting creatures in Tibia involves attacking them until their health points reach zero. Players can attack creatures in two ways: basic attack and attack spells. Most creatures that give experience points in Tibia are aggressive, this means that they will engage in combat with the player as soon its with a certain range of them. Creatures don't have a global spell cooldown which means every two seconds they can cast all of their spells, however its unlikely to happen because the spells are cast by chance. In order to survive the engagement players can use healing spells and potions to restore their health points and to keep using spells they can use mana potions to restore their mana points.

d. Hunting

A variety of places in Tibia has spawns of creatures which each of them gives a specific amount of experience points and items dropped(loot) while having a specific amount of health points and attack(which is a combination of basic attack and spells), some creatures are better to hunt for XP and others for loot. What makes a good hunting spot in Tibia is the combination of amount, density, respawn time of creatures in it, the difficult of hunting there and its popularity. Since Tibia is not a instanced MMORPG and its world is shared between the entire game world, the best hunting spots are usually disputed which may turn them a bad choice for hunting, so the popularity of a hunting spot is a factor to consider too because most hunting spots does not support more than a player hunting which will decrease the amount creatures each player will be able to kill. The hunting in Tibia basically consists of killing creatures doing a specific circular route, ensuring that when the player completes the route, monsters have respawned. Each vocation has particular hunting spots that are best suited to it.

IV. Methodology

The world of Tibia is composed of 2304x2048 SQMs and 14 floors, which gives a total size of 66.060.288 SQMs. This amount of information must be loaded by the proposed agent in order to use the GPS (Global Positioning System) of Tibia. Since each SQM is represented by 3 bytes (BGR), the whole map size accounts to 189 MB.

In order to make a fully automated hunt agent we built 4 agents that run simultaneously:

- GPS Algorithm: it is responsible for retrieving the global position of the character;
- Healing Algorithm: it is responsible for healing the character and not letting it die;
- Target Algorithm: it is responsible for targeting monsters, hence defeated them and "looting" (collect) their items;
- Cave Bot Algorithm: it acts to walk around the hunting place according to the GPS;

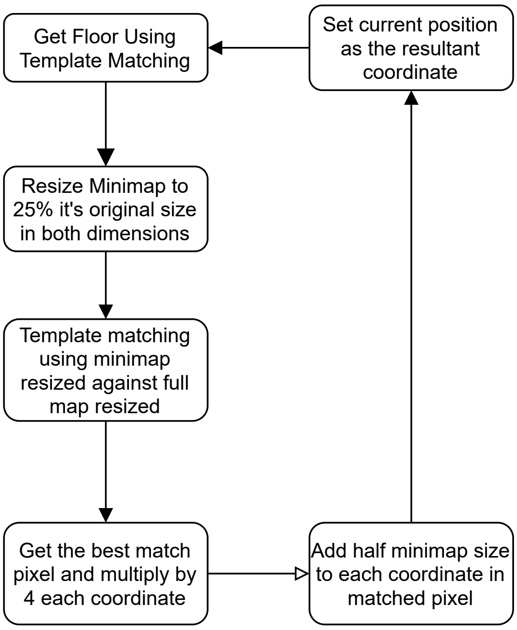

Fig. 2: Diagram of GPS Algorithm

In Fig. 2 is shown the proposed GPS algorithm, which is an optimized template matching to get the minimap and match against the full map. The first step is getting the floor which will be a template matching against each of 14 possible floor positions located at the right side of the minimap which provides the correct floor. After that, to find a good match, the algorithm resizes the minimap to 25% of its origianl size, at each dimension and match against a copy of the full map, also resized to 25% of its original size. This process reduces the search space to 1.179.648 pixels, which in BGR means 3.375MB.

The healing algorithm is a simple pixel check in the health and mana bars. If the specific pixel has not the expected color, then a configured healing shortcut is sent to the game.

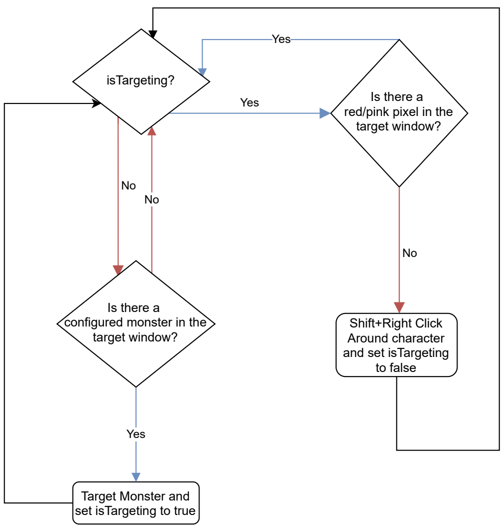

Fig. 3: Diagram of Targeting Algorithm

In Fig. 3 is shown the target system, which is a template matching to get the monsters in the target window and matches them against a preconfigured monsters hunting list. If the match is above some preconfigured threshold and there is no red/pink pixel in the target window (which means the character has already engaged some other monster), then the algorithm will click the monster to attack it. After defeating the engaged monster, the target algorithm sends specific keybindings to the positions around the character, in order to collect the items generated by the victory.

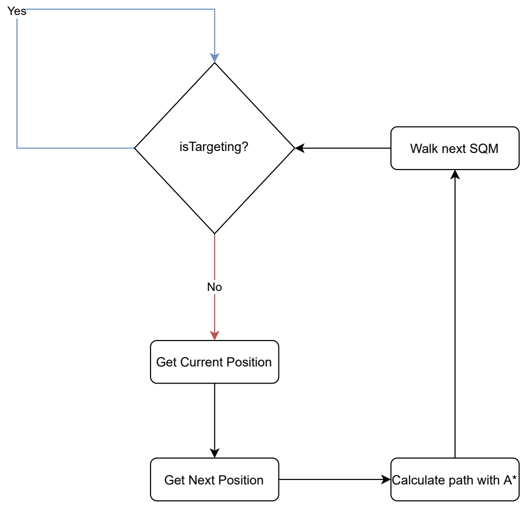

Fig. 4: Diagram of Cavebot Algorithm

In Fig. 4 is shown the cave bot algorithm, which takes as input a circular buffer of positions to walk. It walks to the next position while there is no monster targeted. If there is a targeted monster it stops, then continue walking. The walking stage is based on the A* search of map space doing optimal search decision based on the speed of the current position towards destination. The map of tibia contains information about the speed at each SQM which is used as weights for A* algorithm.

a. Agent Evaluation

During the gameplay, the TIBIA's character has some performance measures. In general, TIBIA measures the following indexes:

1. *XP Gain*: this is the mainstream measure. It increases when the player kills creatures and decreases when the player dies;
2. *Loot*: this is the sum of the value of all items dropped by creatures in gold pieces;
3. *Supplies*:this is the sum of the value of all items wasted to hunt the creatures in gold pieces;
4. *Balance*: this is the Loot minus the Supplies in gold pieces;
5. *Killed Monsters*: the amount of killed monsters.

Because Killed Monsters and Loot are linear dependencies of the other three, during the simulations, we evaluate the agent performance using three main measures: *XP Gain*, *Supplies*, and *Balance*.

V. Simulations and Results

There is a variety of monsters in Tibia for hunting although since we only had a character of level 89 we had a limited variety of hunting places to choose, for this reason we chosen three hunting places that would fit our test character.

In order to evaluate the performance of the proposed agent, we designed three test scenarios: Scenario I: Wasp cave(Ab'Dendriel), Scenario II: Nomads Cave(Ankrahmun/Darashia), and Scenario III: Lion Sanctuary(Darashia). The scenarios are described in Table 1. For each scenario, we ran the agent for 10 times, with sessions of 15 minutes and for 5 times, with sessions of 1 hour. For each run session, the agent began in similar starter points.

Our choice of doing 10 sessions of 15 minutes was due to two main reasons. First to test its reliability and stability against bigger sessions in terms of results and second due to Tibia stamina system, which gives bonus XP in the first 180 minutes of gameplay each day, hence, with our choice we have been able to conduct one test scenario per day.

TABLE 1: DESCRIPTION OF THE EVALUATED SCENARIOS.

Scenario	Description
I. Wasps	This is the easy level and the main monsters the agent has to face is Wasps. Each Wasp may attack only for a low amount of damage which is healed by only food the agent is eating. The main difficulties an automated agent can face here are the map exploration.
II. Nomads	This is mid-level hunting spot which holds a high-demand item(rope belt) which has a good value for Loot hunting. The main difficulties an automated agent may face here are the high amount of creatures.
III. Roaring Lions	This is a high-level popular hunt spot for great XP Gain with little to no waste in terms of Balance. The main difficulties an automated agent may face here are the high amount of damage creatures inflict to it.

a. 15 minutes sessions

In Fig. 5 it is shown the agent performance during the 10 sessions, using the Scenario I. It can be noticed that the balance variable has a high standard deviation, deviating from the average especially in runs 2,6,7,8,9. This happened because of the random nature of drops in Tibia. Once a Wasp monster is defeated, the dropped items does not follow a pattern. Notice, however, that the XP gain is stable, because it is related to the experience in defeating the related monster. Considering the *Supllies* measure, it can be seen that there was a higher use of supplies in run 1 and a suddenly loss during run 5. This happens due to the use of *brown mushrooms* as supplies which is used once every 264 seconds. So as the agent is running during sessions of 15 minutes, it means that it will be used around $15 * 60/264 = 3.4$ mushrooms for every session. Therefore, there are sessions that will use more than the expected 3 mushrooms.

In Fig. 6 it is shown the agent performance during 10 sessions, using Scenario II as well. It can be noticed a lower standard variation in balance when compared with Scenario I. This is due to the drop rate of valuable items from nomads being higher than of the wasps. This summed to the higher

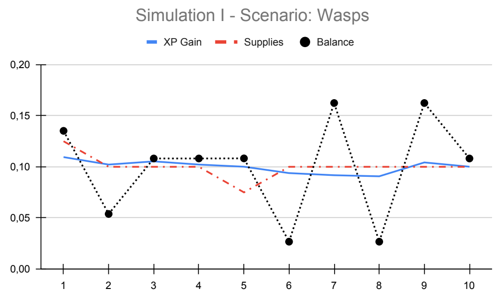

Fig. 5: Normalized results of Wasps Scenario.

amount of monsters defeated resulted in a lower standard deviation of the balance variable. The variation that can be noticed in the XP Gain is due the nature of the cave scenario. Thus, sometimes the agent end up walking through a hole in the cave, having restarted its route.

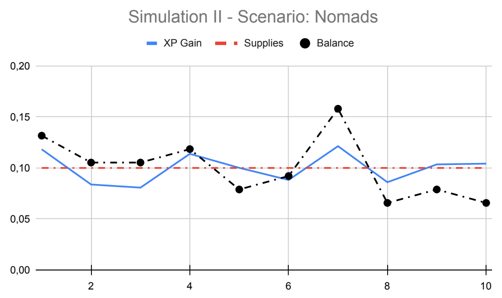

Fig. 6: Normalized results of Nomads Scenario.

In Fig. 7 it is shown the agent performance during 10 sessions, using Scenario III. It can be noticed a high standard deviation in balance. This is due to the low drop rate of items from Roaring Lions. It can be noticed that the XP Gain and Supplies are tied due to the amount of supplies it takes to defeat a Roaring Lion. This is the unique Scenario that the proposed agent had to use *Mana Potions* to defeat the monsters, since each *Mana Potion* costs 56 gold pieces the balance got hit by it. It can be noticed too, that there is a standard deviation in XP too, which is mainly because some players killed monsters of the hunting place while we were running the session during some sessions and secondarily because the agent actually killed more Roaring Lions during the session sometimes.

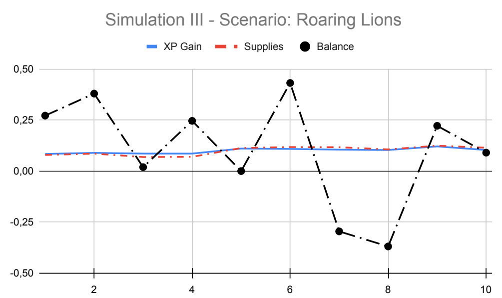

Fig. 7: Normalized results of Roaring Lions Scenario.

We also conducted another analysis, plotting each measure into the three evaluated scenarios. In Fig. 8, it is shown the XP for all scenarios (with no normalization). Looking at the

8, as expected the XP Gain of Roaring Lions was clearly above Wasps and Nomads due to the nature of each scenario. We note too that nomads has a high XP Gain which is of course due to the Scenario being a higher level of difficult than Wasps, since a Wasp gives 36 XP while a Nomad gives 90 XP.

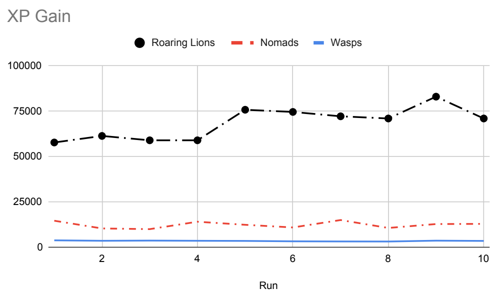

Fig. 8: XP Gain results of all simulations.

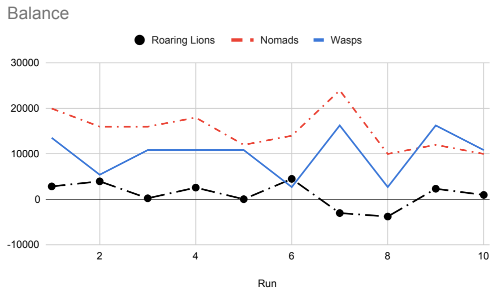

Fig. 9: Balance results of all simulations.

In fig. 9 as expected balance of Roaring Lions scenario runs were the lowest since its nature is basically XP Gain without any profit. We also noticed that Wasps balance are under Nomads balance because Nomads scenario is a higher level of difficulty than Wasps. In addition, it is noticeable that there is a high standard deviation in the balance which is due to the randomness nature of loot in Tibia.

b. 1 hour sessions

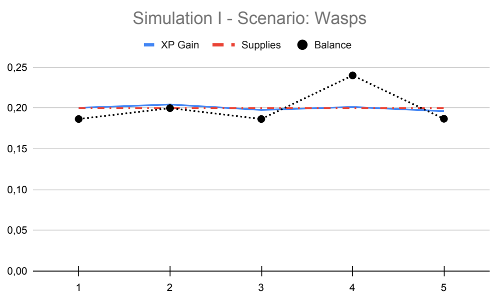

Fig. 10: Normalized results of wasps for 1 hour session.

In fig. 10 is shown the results of 5 sessions using the Scenario I. It can be noticed a much lower standard deviation in comparison with fig. 5, which is probably due to the higher duration of the sessions. This is a good result since it signals the stability of the agent performance in this scenario. The Balance still has higher deviation due to its random nature but the impact has been smaller in these higher duration sessions.

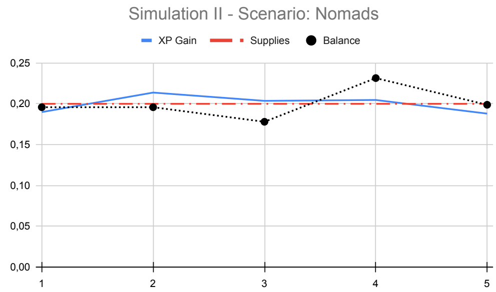

Fig. 11: Normalized results of Nomads for 1 hour session.

In fig. 11 is shown the results of 5 sessions using the Scenario II. It can be noticed a lower standard deviation in comparison with fig. 6, which shows us that the higher the duration of the sessions, stabler are our agent results. The Balance metric still shows a some standard deviation, but lower when compared to the 15 minutes sessions.

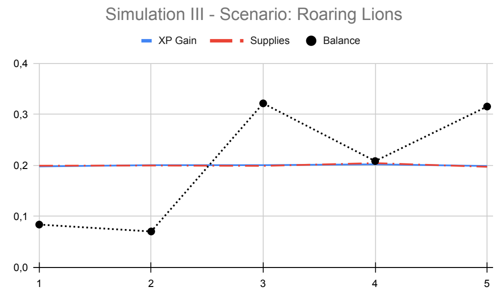

Fig. 12: Normalized results of Roaring Lions for 1 hour session.

In fig. 12 is shown the results of 5 sessions using the Scenario III. It can be noticed that there is almost no deviation in XP Gain and Supplies, which signals the stability of the proposed agent in this scenario. The Balance still has a high deviation due to the low drop rate of some high value items from Roaring Lions.

c. Comparison with human players

In order to evaluate the performance of the proposed agent we got a volunteer to run a session of 1 hour in Scenario I and Scenario II. The volunteer was not able to run a session in Scenario III, after trying to go to the hunting place one time and dying, he gave up due to his skills not being enough for it. Our volunteer was a 26 years old with basic knowledge of the game. The sessions were done using the same character which we conducted the agent sessions.

In fig. 13 is shown the comparison of the session done by the volunteer with the best and worst sessions done by the agent in Scenario I. It can be noted that the agent achieved almost the same XP Gain of the human player, which means it killed almost the same amount of creatures as the human player did. In regards to balance, due to the random nature of loot, the agent may had a bad luck in its worst session but almost the same balance of the volunteer in its best session. The Supplies were in the range of 60 to 140, so it did not appear on the chart.

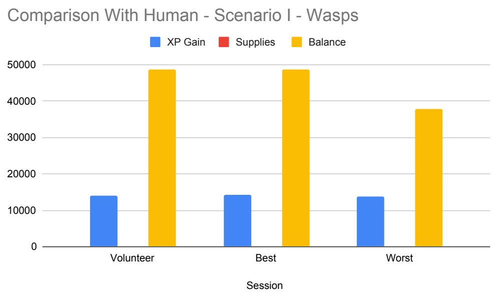

Fig. 13: Comparison of Human Volunteer with Best and Worst session of 1 hour for Scenario I - Wasps.

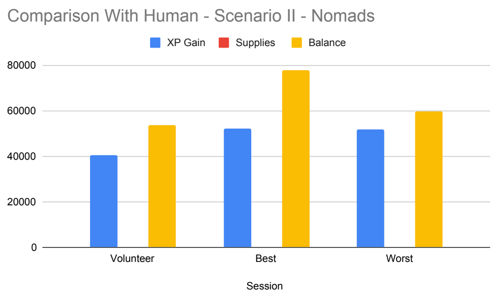

Fig. 14: Comparison of Human Volunteer with Best and Worst session of 1 hour for Scenario II - Nomads.

In fig. 14 is shown the comparison of the session done by the volunteer with the best and worst sessions done by the agent in Scenario II. It can be noted that the agent's performance was better than the volunteer in both XP Gain and Balance. Supplies were in the range of 130 to 140, so they did not appear on the chart.

When doing a preliminary comparison with public sessions recorded by some gamers, in the environment described here as Nomads Scenario the proposed agent produced around 15.000 balance of gold pieces per session of 15 minutes which translates into 60.000 gold pieces per hour against 78.000 of Cyf [17] and in the environment described here as Roaring Lions Scenario the proposed agent produced a result of around 70.000 experience points per session of 15 minutes which translates into 280.000 experience points per hour against 350.000 experience points per hour of [18]. This results are primarily due to faster input/output of the agent and its nature as an automated agent of never being affected by emotions.

VI. Final Remarks

This paper presented the development and evaluation of a reactive agent for the MMORPG Tibia. The proposed agent used only raw data input as the main screen of the game and an algorithm based on Template Matching to process the images. Furthermore, it has been implemented a GPS module based on the A* algorithm which is used to guide the agent through the map and target monsters along the way.

The evaluation consisted of three distinct scenarios with three different levels of difficulties. For each scenario, the agent is submitted in sessions of 15min and 1 hour, and three main measures are collected: XP, Supplies, and Balance. As expected, in the most difficult scenario, the Roaring Lions, the agent made XP close to what an human player would do, but without dying a single time in the evaluated sessions. In the easier scenarios, suited to farm gold pieces, the agent behaved well farming values near to what a human player would farm.

Future research investigations include: a) improvement of the proposed agent framework to allow the evaluation of players, gathering all their inputs and outputs visible through the screen, in order to evaluate the agent's performance against other players; b) comparison with agents using other approaches such as reinforcement learning; and c) benchmark with more human volunteers; d) Research the optimum value of map resize for the best performance of the GPS Algorithm.

References

[1] N. Cano, *Game hacking: developing autonomous bots for online games*. No Starch Press, 2016.

[2] J. Y. Wang, "Combat State-Aware Interest Management for Massively Multiplayer Online Games," Master's thesis, TECHNISCHE UNIVERSITÄT MÜNCHEN, 2017.

[3] N. Ferreira, P. Trovo, and S. Nesteriuk, "Emergence in game design: Theoretical aspects and project's potentialities," in *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer, 2017, pp. 597–611.

[4] J. Schaeffer, M. Müller, and A. Kishimoto, "Go-bot, go," *IEEE Spectrum*, vol. 51, no. 7, pp. 48–53, 2014, cited By 1.

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[6] S. Sharma, S. Mishra, N. Deodhar, A. Katageri, and P. Sagar, "Solving the classic snake game using ai," in *2019 IEEE Pune Section International Conference (PuneCon)*, 2019, pp. 1–4.

[7] A. Kang, S. Jeong, A. Mohaisen, and H. Kim, "Multimodal game bot detection using user behavioral characteristics," *SpringerPlus*, vol. 5, 2016.

[8] T. Dharmawan and N. Hanafiah, "Clicker bot for gacha games using image recognition," *Procedia Computer Science*, vol. 179, pp. 598–605, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921000521

[9] X. Cui and H. Shi, "A*-based pathfinding in modern computer games," *International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 125–130, 2011.

[10] S. E. Ayala-Raggi, P. d. J. González, S. Sánchez-Urrieta, and A. Barreto-Flores, "A simple view-based software architecture for an autonomous robot navigation system," in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds. Cham: Springer International Publishing, 2015, pp. 287–296.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: a modern approach*, 3rd ed. Pearson, 2009.

[13] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[14] B. Cyganek, *Object Detection and Recognition in Digital Images*. John Wiley and Sons Ltd, 2013.

[15] R. Brunelli, *Template matching techniques in computer vision: theory and practice*. Wiley, 2009. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=3F5932473E6E9C09092322AC4F5EC6E0

[16] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[17] C. Plays, “Rope belt farm (level 20+ / 70k+ profit) - bonus : Arito’s task quest.” 2018. [Online]. Available: https://www.youtube.com/watch?v=f_si_oDb1LI

[18] Itexo, “Ek 54 lion’s rock - 350k/h - os melhores lugares de hunt para knights,” 2018. [Online]. Available: https://www.youtube.com/watch?v=7q6fD6q2cYc