A.I and Game Design:

The History of Artificial Intelligence in Video Games

Abdellah Regad

Private International Institute of Management and Technology

Bachelor's Thesis

Year 2022

Acknowledgement

First and foremost, I thank my parents for bringing me to life, raising me in the best conditions, and helping me whenever I am down. I would like to thank my sisters for being the best they can be and supporting me in my laziest times.

My utmost thanks to Oussama for being a good friend and giving his input and feedback on the study whenever I asked him for it.

I would also like to thank my friends & colleagues who, without, would have been a rather lonely journey. You have been of great support and created an enjoyable experience.

And last but not least, I also am deeply thankful to Dr Mouti Jalal for being an amazing supervisor whose guidance was of utmost importance in the study.

Abstract

This study exists consequent to our curiosity about the uses of Artificial Intelligence in the industry of video games. As popular as video games are, their development tends to be challenging. They need to tell a story and provide the player with compelling gameplay while ensuring the player is always entertained. It is a job that requires many specialists to work on it, and sometimes, they have to think in ways that seem out of the ordinary, but this, in turn, creates beautiful and masterful pieces that we will be discovering in this work. Artificial Intelligence plays a huge role in making the games what they are, be it narrative directors, finite state machines, or even procedural generation. Each tool in the arsenal of game developers is used to its fullest potential to craft deep, meaningful games with compelling gameplay or games with narratives that dynamically changes every time the game is played.

People's curiosity in the quest to understand the mechanics behind the relation between Artificial Intelligence and Game Design will be left satisfied at the end of the study.

A.I and Game Design:

The History of Artificial Intelligence in Video Games

Table of Contents

Acknowledgement……………………………………………………………..2
Abstract..3
AI and Game Design: The history of Artificial Intelligence in Video Games........4
Table of Contents 5
Table of Pictures..7
Table of Abbreviations...9
Chapter 1: Introduction 10
Background of the study 10
Research Questions...12
Chapter 2: Literature Review..13
Intelligence...13
Definition 13
Artificial Intelligence 14
Definition 14
How Machines Learn...15
The First Artificial Intelligence Victory...16
History of Chess...16
First Chess AI and the first loss of mankind to computers 17
Minimax algorithm and the development of AI algorithms........................18
Finite State Machines...20
Behavioral Scripts & Hierarchical Finite State Machines 25
The STRIPS system and the 3 states of F.E.A.R. 26
Summary of the Literature Review..29
Chapter 3: Methodology...30
Chapter 4: Findings 31
Systemic Games and predictable patterns...31
The A.I Director 32
Generating a Narrative Experience using A.I 34
Nemesis and Kuva Lich systems 34
Head Verbs and Modular Storytelling...36
Machine Learning tribes and EAs 37
Machine Learning, a new frontier in AI research 41
Creativity in AI and Procedural Generation to create Narratives.................43
Procedural Generation 43
Artificial Intelligence, a game designer...47

A bottleneck in the future of AI in game design 48
A dream 50
Chapter 5: Conclusion 51
References 52

Table of Pictures

No.	Name	Description
Figure 1.0	Screenshot of a chessboard from chess.com	
Figure 1.1	Some possible choices that can be made by white in this board state.	In red arrows, the white player predicts from his opponent after he performs his play.
Figure 1.2	A basic Finite State Machine for combat A.I.	
Figure 1.3	The Finite State Machine diagram of the game Pac-Man	From the diagram, we can see that sometimes, two states can switch between one another.
Figure 1.4	A* path	
Figure 1.5	MCTS algorithm	
Figure 1.6	The three states of F.E.A.R's state machine	
Figure 1.7	Complex Squad Behaviors emerging from the situation	
Figure 1.8	The cycle between the build-up, sustain peak, peak fade, and relax states with the build-up gauge on the left	
Figure 1.9	Kuva Lich	A Kuva Lich with different traits
Figure 1.9	Synthesis of all five machine learning tribes into a single algorithm by Pedro Domingos in "The Master Algorithm."	

Figure 2.0	Process of an Evolutionary Algorithm	
Figure 2.1	The influenced map method.	This graph model rudimentary elements of urban decay
Figure 2.2	A small portion of start in the Norma Arm Region of our Galaxy	As of January 20, 2022, only 0.05%, or exactly 222,083,678 unique star systems have been discovered
Figure 2.3	Stellar Forge results, 400-billion-star systems in our galaxy	
Figure 2.4	Level Generation of Spelunky	
Figure 2.5	Yavalath Puzzle	
Figure 2.6	Example graph of path dependency	The graph depicts how the stream of available options thins, locking us in a specific path

Table of Abbreviations

Terms	Full Terms	Term Definition
N.P.C.	Non-Playable Character	Used to denote a Video-Game Character that the player does not control
A*	A-Star	A Path Search Algorithm
AID	AI Director	
AI	Artificial Intelligence	
IQ	Intelligence Quotient	The term referred to the test taken to measure Intelligence
M.T.C.S.	Monte Carlo Tree Search	Searching Algorithm
U.C.B.	Upper Confidence Bound	An algorithm in Reinforcement Learning
F.S.M.	Finite State Machine	
H.F.S.M.	Hierarchical Finite State Machine	
STRIPS	STanford Research Institute Problem Solver	
RPG	Role-Playing Game	A game where the player plays a character that has a narrative.
MMORPG	Massively Multiplayer Online Role-Playing Game	A Role-Playing Game where players in the thousands exist in the same world via online connectivity
E.A.	Evolutionary Algorithm	
GDC	Game Developers Conference	A conference held every year where developers elaborate on the process of the development of their games

Chapter 1: Introduction

Background of the study

Video Games have been a staple in households, specifically in the last couple of decades; it may be obvious why video games have become as popular as they are. In 2021, the global video game market value was over 178 billion U.S dollars, and it is not wrong to assume the numbers would go up by a lot more, with studies showing that by the year 2025, the video game market value would surpass 250 billion U.S dollars. Nevertheless, this study is not about video games' revenue or market value. What we want to study is one of the cores of these games that created such a market, that being Artificial Intelligence.

We were exposed to computers at an early age. At the time, we were fascinated by such technologies. We still are, but it was something new to us; it did not take us long to discover our very first video game. The first time we booted the game "Prince of Persia 1989", the experience was unique, the fact that every time we touched a button, something happened, but more than that is how the game made us feel excited. Of course, as any person, we have decided to search for more games that we can try, testing many consoles and trying to understand the games, and as much as there were variations at the time, we have noticed the same pattern happening, more notably in enemies, they all had one goal, eliminate the player by some means, if the enemies have guns, the player will get shot, if they had swords, they would try to stab him. As basic as they were, it was enough for us to be engrossed by them as they were so intractable and understandable. Rather, there was the hint of a possibility of understanding them. This led us to play the various games we had at hand, and we marveled at the depth of possibilities within those games. We understood that some strict rules were followed, and those rules ultimately led to the patterns we discussed. Following that, we also discoverer the limitations of these video games, the secrets, and weaknesses, which made the games a lot easier. At the time, the patterns exhibited by the N.P.C.s in the games were predictable; the player would know how far they go to the left and what they would do next. However, we wanted there to be more than just that. We wanted there to be secrets, endless worlds to explore within those games, characters with lives of their own, and a never-emptying treasure trove of secrets to discover. Moreover, we wanted there to be things happening that we could not predict, but which still made sense.

Eventually, games have become more complex, with technologies that could support the ideas that men dreamt of, breathtaking ideas, intuitive gameplay, and beautifully crafted stories delivered with amazing sceneries and interactions. However, many of these points are lost if the A.I. at the core of the game is not very well implemented. However, A.I. is not something only games have developed and improved over the years. The idea of Artificial Intelligence has existed long before games, and the study of it is now intertwined with video games.

Research Questions

As developed before, our curiosity around the subject of Artificial Intelligence and Video Games has enabled us to determine and write this research with the main guiding objective being the history of A.I. in video game design. To develop this objective, we have formulated questions whose answers will constitute the study's findings. The questions formulated are as follows:

1. How is Artificial Intelligence defined, and how does it differ from Intelligence?
2. What is considered A.I. in video games?
3. What Algorithms are used to create A.I. in video games?
4. How does A.I. help in the making of video games?
5. How does A.I. affect Game Design, or how is it incorporated into the game's design?

Chapter 2: Literature Review

Intelligence

Definition

We can define Intelligence as a set of behaviours or reactions of something or someone in a scenario that would make him/it distinguishable from the rest. It is the ability to learn, understand, or deal with new or trying situations. There is no agreement as to how to define Intelligence, but it is agreed upon that, to a degree, Intelligence is present in every being or thing that can think. Of course, the Intelligence of a being greatly varies. To measure our Intelligence, we have special tests called I.Q. (Intelligence Quotient) Tests. In the case of other beings, we measure their Intelligence based on their cognitive ability. Looking at Koko the guerilla, she can understand 2000 words and communicate them with hand signs, a sign of Intelligence and high cognitive ability. We have been talking about the Intelligence of animals, and humans, so we can understand their thought processes. We need to study them and know how they react and think. The curiosity in this research is crucial to what is to be said in later developments around this topic. Nevertheless, we can be sure that Intelligence is a research subject in many fields of science for now.

Artificial Intelligence

Definition

John McCarthy defines Artificial Intelligence as the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human Intelligence, but A.I. does not have to confine itself to biologically observable methods.

In our own words, Artificial Intelligence is a computer-made intelligence; it is artificial only in the way that it seems to us as human Intelligence; it is a façade, it only shows its intelligent parts and hides the process. We program the machines to act a certain way based on the scenarios they live in; its only known purpose at that moment is to react to something that is within its expectation with something else as an answer; it is a mere tool for us, ask it something outside of its expectation, and there would be no results. We tell the computers what to do to mimic our Intelligence; that is why A.I. is implemented in many fields, and even in fields that seem to be out of reach of human Intelligence. This is a fascinating point regarding artificial Intelligence; with computers getting better and more performant, we can apply theories and algorithms that exist in theory but cannot be applied by us humans as it would take too much time, and it would seemingly be impossible, and with such use, we create tools that help us in our lives, some of which we will see down the line.

Although it is possible to create algorithms and utilize data in support of this process within a computer, a computer's capability to achieve Intelligence is severely limited. For example, we have talked early on about how computers only know what they should do, thus, making them rely on machine processes to manipulate data using pure math in a strictly mathematical fashion. In the same way, computers cannot easily separate right from wrong. Our computers cannot handle any of the mental activities described in the list that describes Intelligence.

How Machines Learn

Now that we agree on Artificial Intelligence, we need to know how to make a machine intelligent. Traditionally, we humans can increase our Intelligence by learning. Knowledge is a key to the growth of our Intelligence. The more a person learns, the more intelligent they become. Of course, there is a genetic component to Intelligence, but we can ignore that fact when speaking about computers. We can argue that the machine's processing power and speed are the "Genetic Component," which can certainly help speed up the learning process or complete tasks faster, but it is a bit different than when we say a person is born a genius. When we use such phrases, we state how the person is already intelligent even though he has not learned much. However, in the case of a machine, its processing power is not intelligent, or simply, it does not have a base intelligence, to begin with, outside of the speed at which it learns and performs tasks.

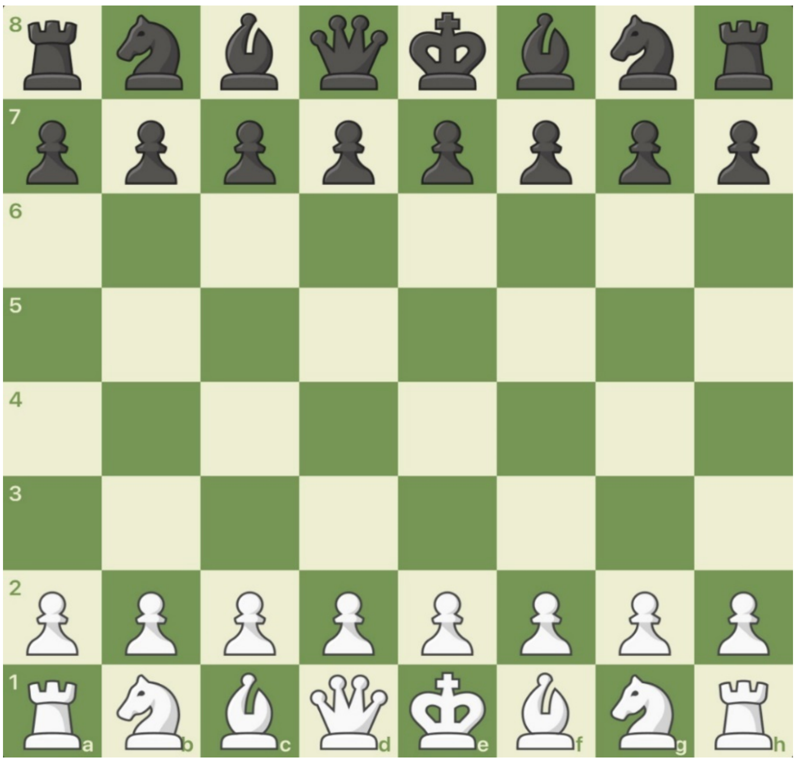
The First Artificial Intelligence Victory

History of Chess

Chess is a game that has existed for thousands of years, tracing back to its earliest known predecessor in the 1500s. The rules are simple to write in English and computer code, and many people play it. Its rise has been very noticeable these last few years, with streamers playing it live on their platforms. It is a board game with a checkered board and pieces with certain constraints; all of this can merge to become a complicated problem, and it would make for a good test to use for creating artificial Intelligence, whose goal is to think like a human. In chess, one must make decisions, predict, and adapt to their opponent's strategy. Reaching master-level play will typically have recognizable playing styles.

Figure 1.0

Screenshot of a chessboard from chess.com



So as long as someone keeps playing, they will eventually get better at it all their life. It is not wrong to assume that chess was suggested as an important problem to work on at the beginning of A.I. research. The game of chess requires high Intelligence to play. So, it was unbelievable that anyone could play it without high I.Q. How can someone play this game without successfully planning, judging the true value of board positions, understanding their opponent's thinking and predicting their moves? The game needed thinking to be able to win. Furthermore, because of that, there exist other activities that require Intelligence other than playing chess. And thus, we assumed that if we made a machine capable of playing chess, we could be able to solve the problem of artificial Intelligence.

First Chess A.I. and the first loss of humankind to computers

The first known Chess-playing program was created by none other than the father of computers, Alan Turing in the late 40s. Chess-playing turned into a vibrant subfield of artificial intelligence research, with conferences, journals, and competitions devoted to studying and developing software that could play chess and similar board games (mainly Go(game)). One of the notable works in board games was done by I.B.M. computer scientist Arthur Samuel in 1958, who invented the first version of what is now called reinforcement learning to make a Checkers-playing program learn from experience.

At first, when Chess-playing programs were being developed, many thought that it was not possible for a computer to beat a human, as humans were intelligent and possessed thoughts, while computers were only made of code that we humans wrote. And a game such as chess requires Intelligence to be played. Hence, at first, Chess-playing programs were not up to even beat a beginner, but along the decades of research, these programs got stronger and better. The fact that computer processors were getting stronger and memory size got larger had a hand in the advancement of the programs. But also, because the programs themselves got better, the code writing was more refined and more algorithms implemented.

In 1997, all the development led to a much-publicized match between I.B.M. with its special purpose Deep Blue Chess computer against the reigning world champion, Garry Kasparov. Ultimately, the computer won. This event was the starting point for a vivid debate about the meaning of Intelligence and artificial Intelligence now that machines had conquered chess. And as we discussed about artificial Intelligence and what it is, we have given our opinion saying Artificial Intelligence is intelligent in only the way we make it. The purpose of Deep Blue was to beat a champion, and it achieved its goal and purpose, but only that, it is not suddenly intelligent though it seems intelligent to us, and it was able to beat this champion with the help of minimax algorithm.

Minimax algorithm and the development of A.I. algorithms

Minimax uses the concept of board states and moves states; board state is the current position of all pieces on the board, and move is the board's transition from one state to another. For example, in the board state 0, all chess pieces are in the default and starting location; once we move one piece, the board state becomes board state 1, and the transition to that board state is called move. Minimax works as follows. Imagine playing from the perspective of the white player. Seeing the state of the board, the player would want to have the best move from said board state. They would, of course, want to know all the possible moves they can make from that state. They would then simulate taking each move and storing all the resulting board states, after doing so, the program took all the state paths that lead to victory

Figure 1.1

Some possible choices that can be made by white in this board state.

Notes: In red arrows can be seen what the white player predicts from his opponent after performing his play.

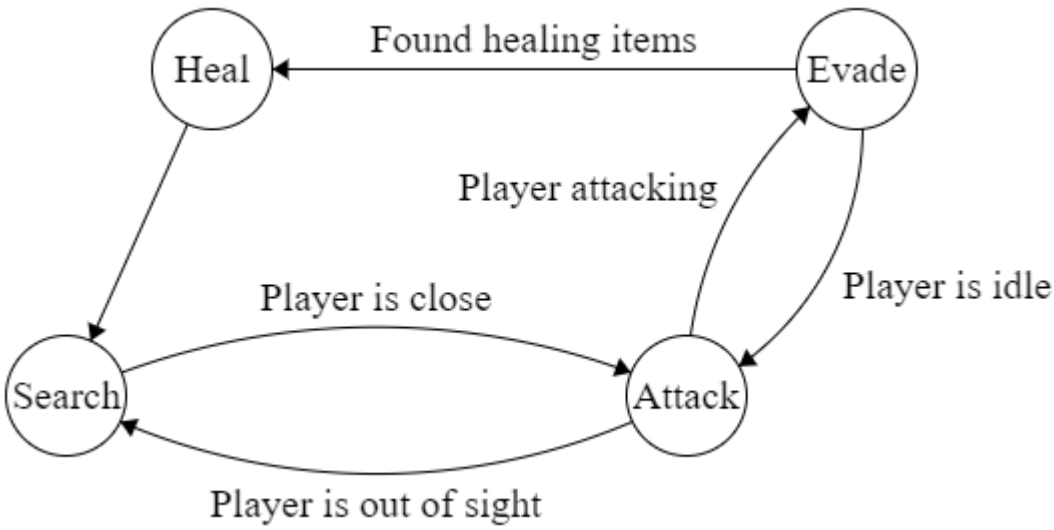
Finite State Machines

A finite state machine is an abstract model of a system (physical, biological, mechanical, electronic, or software). Only a single state can be active simultaneously, so the machine must transition from one state to another to perform different actions.

F.S.M.s are commonly used to organize and represent an execution flow, which is useful to implement Ai in games. The "brain" of an enemy, for instance, can be implemented using an F.S.M.: every state represents an action, such as **attack** or **evade**

Figure 1.2

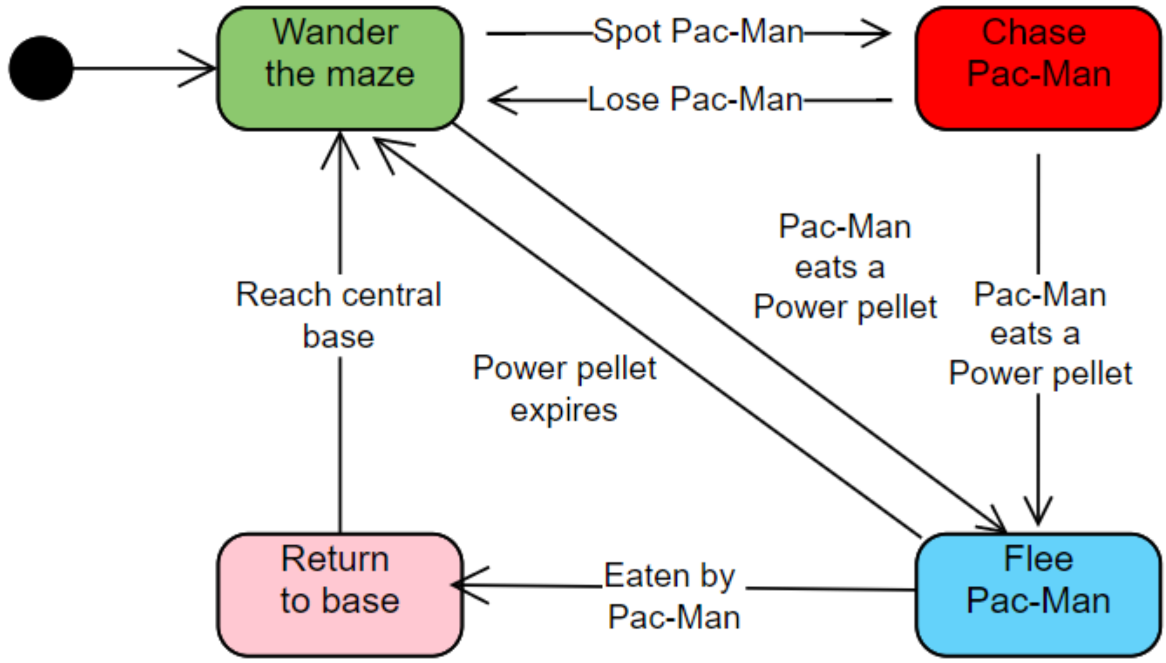
A basic Finite State Machine for a combat AI



A good example of a game that uses F.S.M.s is Pac-Man. In the game Pac-Man, the neutral state consists of four ghosts, each following a specific script chasing Pac-Man; one of the ghosts chases him normally, one looks two tiles ahead of him, another combines these two, and the last retreats when he gets too close to him. However, as soon as Pac-Man eats a power pellet, the ghosts enter a new state where they all start fleeing from him, and in case they are eaten, they return to base. To demonstrate this with an F.S.M. diagram, it would look like the following:

Figure1.3

The Finite State Machine diagram of the game Pac-Man



Notes: From the diagram, we can see that sometimes, two states can switch between one another.

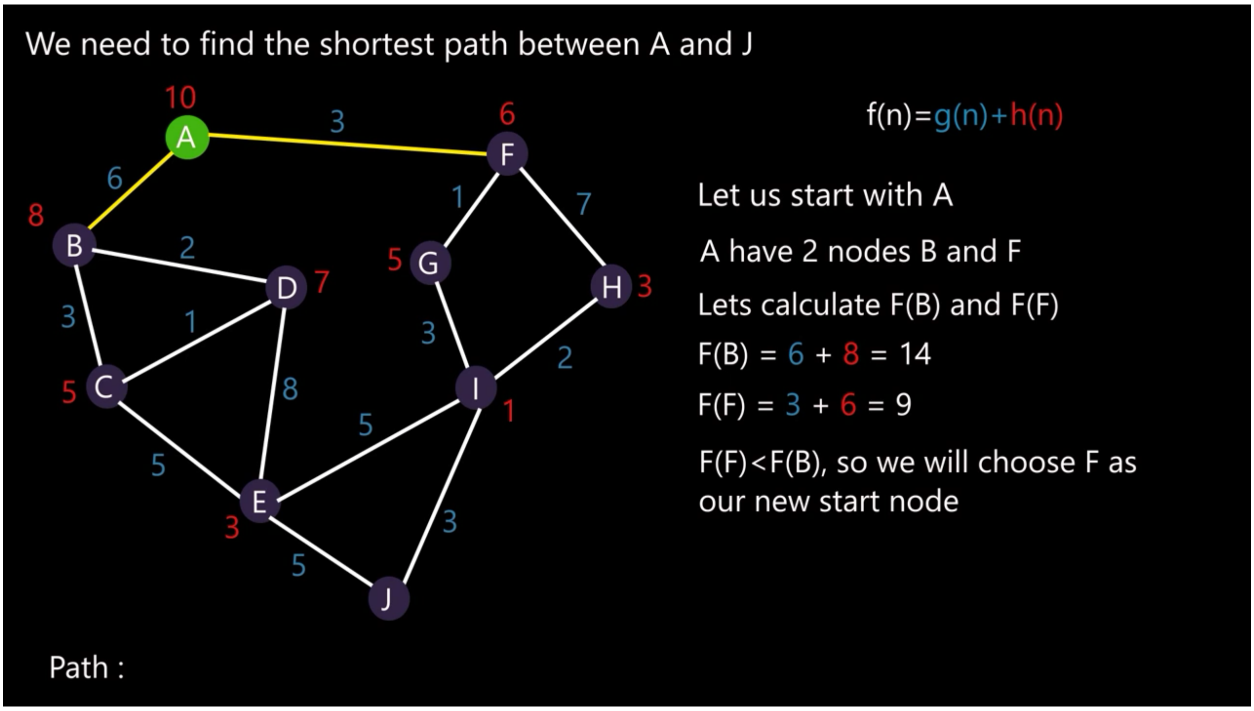
Thus, creating a dynamic feeling where the ghosts act depending on the user's input or actions. This can also be found in stealth games such as Metal Gear or Batman. In these games, Enemies use a path-finding algorithm to search for the player and based on his actions, it forces them to change behaviour. The algorithm mainly used for this is the A* algorithm, which gives a sense of Intelligence to the N.P.C.s.

A* Algorithm & MCTS

A* is a graph traversal search algorithm and path search algorithm. It is essentially a Best First Search algorithm(Informed Search). A Best First Search algorithm is a class of search algorithms in which nodes are picked by checking the most prominent one of them and expanding with a specified rule. The graph traversal technique or graph search technique refers to the process of checking and updating each vertex in a graph. Such traversals are classified by the order in which the vertices are visited. As mentioned, A* maintains a tree of paths originating from the start node. Those are extended one edge at a time; this procedure is repeated until its termination criterion is satisfied. The function used to extend the path is the following: f(n) = g(n) + h(n). n is the last node on the path, g(n) is the path's cost from the start node to node n, and h(n) is a heuristic function that estimates the cost of the cheapest path from node n to the goal node—speaking of heuristic functions. A heuristic function is an evaluation function that estimates the cost of getting from one place to another. It is a way to inform the search about the direction to a goal.

Figure 1.4

A* paths



Monte Carlo Tree Search is a search technique in A.I. It is a probabilistic and heuristic-driven search algorithm that combines the classic tree search implementations alongside machine learning principles of reinforcement learning. There is always the possibility that the current best action is actually not the most optimal action in tree search. In such scenarios, the M.C.T.S. algorithm becomes useful as it continues to evaluate other alternatives periodically during the learning phase by executing them instead of the current perceived optimal strategy. This is known as the "exploration-exploitation trade-off." It exploits the actions and strategies that are found to be the best until now and must continue to explore the local space of alternative decisions and find out if they could replace the current best.

Exploration helps explore and discover the tree's unexplored parts, which could result in more than its depth. Exploration can be useful to ensure that M.C.T.S. is not overlooking any potentially better paths. However, it quickly becomes inefficient in situations with many steps or repetitions. In order to avoid that, it is balanced out by exploitation. Exploitation sticks to a single path that has the greatest estimated value. This is a greedy approach, extending the tree's depth more than its breadth. In simple words, the U.C.B. formula applied to trees helps balance the exploration-exploitation trade-off by periodically exploring relatively unexplored nodes of the tree and discovering potentially more optimal paths than the one it is currently exploiting. For this characteristic, M.C.T.S. becomes particularly useful in making optimal decisions in Artificial Intelligence problems.

In M.C.T.S., nodes are the building blocks of the search tree. These nodes are formed based on the outcome of several simulations. Monte Carlo Tree Search can be broken down into four distinct steps, viz, selection, expansion, simulation, and backpropagation. Each of these steps is explained in detail below:

Selection: In this process, the M.C.T.S. In this process, the M.C.T.S. algorithm traverses the current tree from the root node using a specific strategy. The strategy uses an evaluation function to select nodes with the highest estimated value optimally. M.C.T.S. uses the Upper Confidence Bound (U.C.B.) formula applied to trees as the strategy in the selection process to traverse the tree. It balances the exploration-exploitation trade-off. During tree traversal, a node is selected based on some parameters that return the maximum value. The parameters are characterized by the formula that is typically used for this purpose is given below

$$S_i = x_i + C\sqrt{\frac{\ln(t)}{n_i}}$$

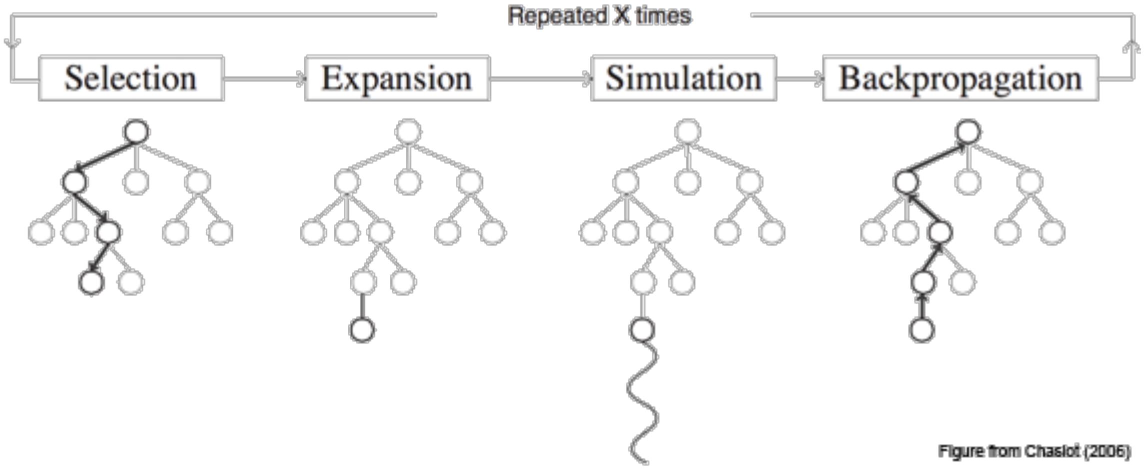
Expansion: In this process, a new child node is added to the tree to that node which was optimally reached during the selection process.

Simulation: In this process, a simulation is performed by choosing moves or strategies until a result or predefined state is achieved.

Backpropagation: After determining the value of the newly added node, the remaining tree must be updated. So, the backpropagation process is performed, where it backpropagates from the new node to the root node. The number of simulations stored in each node is incremented during the process. Also, if the new node's simulation results in a win, then the number of wins is also incremented.

Figure 1.5

Example of a Monte Carlo Tree Search algorithm at work



This leads us to a variant of the Finite State Machine.

Behavioural Scripts & Hierarchical Finite State Machines

In the behaviourist approach to psychology, behavioural scripts are a sequence of expected behaviours for a given situation. For example, when individuals enter a restaurant, they choose a table, order, wait, eat, pay the bill and leave. This is often used in video games and was popularized by Bungie in Halo 2. The main goal was to fix the complex problem of creating believable characters. Because of how wide the levels were, Bungie could not simply rely on patterns and had to use behavioural scripts for idle combat and defensive states. Behaviour trees are very similar to Finite State Machines, with the main difference being that the former is hierarchical. With the idea implemented, some behaviours exhibited by the A.I. were prioritized over some other behaviours. Behaviour trees consist of nodes and child nodes, each representing a sequence of actions, conditions, logical operations, or negation that can be triggered by monitoring the game's state. One important behaviour that is implemented in Halo 2 is Stimulus behaviour.

Stimulus behaviour forces an impulse on the part of the N.P.C. to give a particular action a priority at a given point in the game. A common example of stimulus behaviour is the fleeing of enemies when the squad leader dies; another example is how jackals can group to create a wall of shields. This impulse essentially waits until an "actor died" event occurs, then springs into action, testing whether the actor who died was a leader. In his paper, Damien Isla expands further on the topic, but Bungie was able to create a more dynamic A.I. in wide areas by using this method.

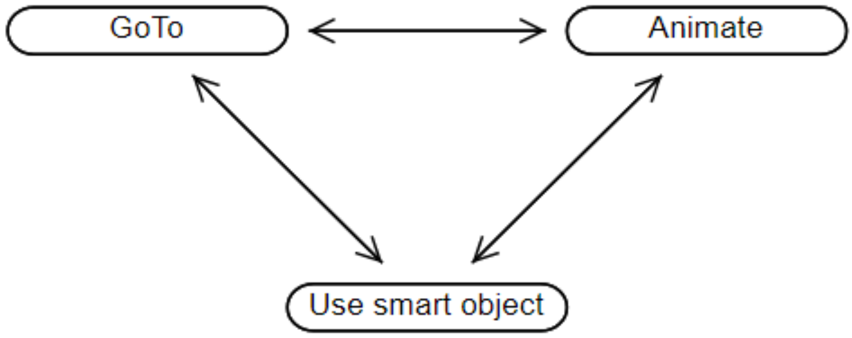
The STRIPS system and the three states of F.E.A.R.

With all the knowledge we have garnered so far, including patterns, dynamic state adjustment, and intelligent H.F.S.M. prioritization tools for design, we can talk about our next game, which implemented all of these tools. Rather than using conventional methods for A.I., *F.E.A.R* uses a goal-oriented method. As the developers said;

"We wanted *F.E.A.R.* to be an over-the-top action movie experience, with combat as intense as multiplayer against a team of experienced humans. A.I. takes cover, blind fires, dives through windows, flushes out the player with grenades, and communicates with teammates. So, it seems counter-intuitive that our state machine would have only three states."

Figure 1.6

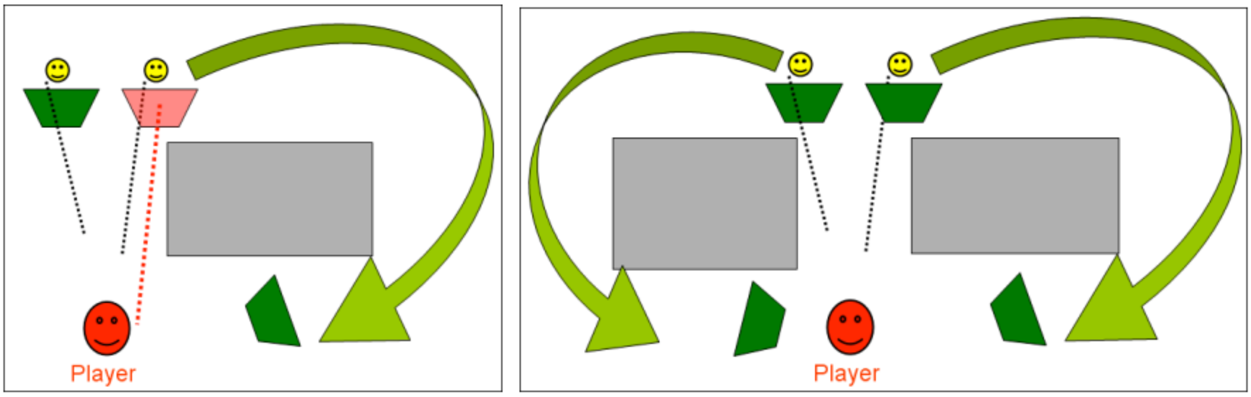
The three states of F.E.A.R's state machine



For F.E.A.R., this is where planning comes in. As the developers said, "We decided to move that logic into a planning system, rather than embedding it in the F.S.M. as games typically have in the past. A planning system gives A.I. the knowledge they need to be able to make their own decisions about when to transition from one state to another. This relieves the programmer or designer of a bigger burden with each generation of games. Fundamentally, a planning system gives the A.I. goals and lets it find the ways to achieve them. This was done using the STRIPS system, which consists of goals and actions, where goals describe some desired state of the world we want to reach, and actions are defined in terms of preconditions and effects. The benefits of this method are that they could layer behaviours to produce complex results, decouple goals and actions to allow different characters to satisfy goals in different ways, and empower characters with dynamic problem-solving skills. There were many ways to solve this problem, and the A.I. would dynamically shift strategies to achieve goals.

Figure 1.7

Complex Squad Behaviors emerging from the situation



This, in turn, creates gameplay that is quite intense where the enemies dynamically adapt to our behaviour coordinating their actions and working together to try and kill the player; they would use melee combat when close to them, they would try to corner them, would throw grenades at them when they are cornered, they would also use cover intelligently. This gives us the illusion of a smart or intelligent A.I. While *F.E.A.R.* focused on creating a cinematic action experience, Bungie focused on wide-area levels.

From the variety of gameplay elements in these video games, we can gather that A.I. does not need to rely on technological advancement; rather, it works depending on the context of the game it lives in. Of course, finite state machines still exist in video games, games like "Metal Gear" have patrolling enemies that follow a certain pattern like Pac-Man, but they also have sensory detection, which they use to detect the player, whether be it sound, vision or fallen comrades, bringing to bear elements of systemic design. In the same, enemies in Batman Arkham will never abruptly turn around, thus creating a sense of predatorial empowerment.

Summary of the Literature Review

With the ending of this chapter, it is only logical to summarize what has been developed.

First and foremost, we have tried to define Intelligence and discovered that there is a list of features that describes Intelligence.

Following, we have defined Artificial Intelligence as a computer-made intelligence and claimed that Artificial Intelligence is only intelligent because we make it do tasks that we humans think to require Intelligence before they can be worked on.

Thirdly we talked about the very beginning of A.I. and how chess was a focus point in studying artificial Intelligence, leading to the defeat of humankind. This, in turn, made us talk more about some of the main Algorithms incorporated in video games; we defined, explained, and provided examples in support of our explanations.

Chapter 3: Methodology

The findings of the next chapter were collected through documentary analysis, reading through books written by experts in the field, official reports done by game companies, journals published by authors who analyze games, and most importantly, by watching old and new GDC (Game Developers Conference) talks in which both developers and enthusiast developed ideas, asked questions, and elaborated on the process that goes behind the development of the video games we analyzed, further improving our understanding of said games.

Chapter 4: Findings

Systemic Games and predictable patterns

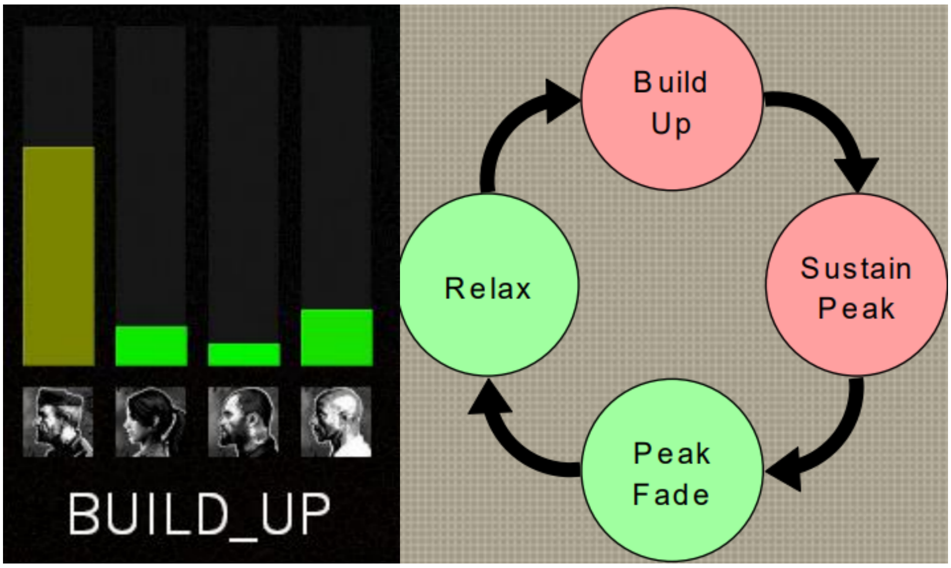
Our research discovered that depending on the type of game being made, giving the A.I. the possibility to interact with systems like in “Zelda breath of the wild” or having the A.I. exhibit predictable scenarios might be a better idea; otherwise, the A.I. would either be too unfair or out of place. A good example of this is the title from 2016, “DOOM,” with over 90 types of A.I. and many types of animations, “DOOM's” gameplay and A.I. encourages the player to keep shooting and move forward, thus eliminating the need to take cover. This is achieved by making the Enemies less likely to shoot the player while he is moving, these same enemies drop health packs and ammo upon performing glory kills, and the enemies also follow a queue system as to when they can attack the player and with how many enemies appear on the screen, this was necessary so the players will not feel overwhelmed, as it could lead to a frustrating experience to the player. Games like assassin's creed benefit from the same idea where enemies will never attack the player all at once, either to work with the game's mechanics or deliver a specific goal. With immersive games like “Dishonored,” “BioShock,” and “Deus ex,” all enemies have predictable patterns that make the possibilities of adaptabilities, improvisation, creative foresight, and pre-planning, leading players to experiment more with the system and what it offers.

The A.I Director

From systemic games, we look into A.I. from a higher perspective. For example, “Left 4 Dead”, a game where you have to fight off waves of the undead, uses something they call "The A.I Director." This A.I. monitor the player's skill level and the intensity felt by each player, thus adapting slowly depending on them; in a scenario where the players' tension is too high, the A.I. adjusts and stops sending enemies; otherwise, it begins a new scenario sending off hordes of undead.

Figure 1.8

The cycle between the build-up, sustain peak, peak fade, and relax states with the build-up gauge on the left



Now we have a fundamental element that attempts to read the player and manage the player experience, not just simply program behaviour. This same behaviour exists in a recently developed game called “Phasmophobia,” using sanity as the term to monitor the player's tension. It uses lighting, location, and overall performance of the player to determine their sanity; the lower their sanity is, the more likely they are to be attacked by the ghost. The good thing about this is the fact that “Left 4 Dead” and “Phasmophobia” are not the only game to implement an "A.I. Director."

Another game that implements an A.I.D. is “Alien Isolation” the game is a lovingly crafted homage to Ridley Scott's movies, where your goal is to escape the titular Alien that is stalking you in a remote space station.

During development, the team behind Alien: Isolation decided that rather than sticking with the newest movie at the time, it would be best to remake the Alien just like the first movie and decided to remake the Alien. For that, it was agreed upon for the Alien not to be scripted and make it unpredictable, as, of course, if the players were able to know what the Alien might do next, all sense of tension would evaporate, thus making the players believe that the key to this Alien's power is a mystery. The Alien is one of the most iconic and popular characters in pop culture, so many people already knew what the Alien is capable of except for what it would do next, and they used that to power the fear of the unknown in the player, and it was their goal to achieve it. Implementing this idea is not easy, especially when you introduce a systemic A.I. into play: a collection of decision-making systems that react to what is happening in the game. Moreover, for that, a design principle was introduced called 'Psychopathic serendipity' in which the Alien always finds himself in the right place at the right time. As explained by Andy Bray in his talk at the 2016 nucl.ai conference, the player cannot be scared senseless all the time, or else they might walk off. The tension had to vary but also maintain an air of unpredictability in the xenomorph itself. Fear is a careful blend of abject horror and stressful anticipation, and the xenomorph's behaviour had to be adjusted with this in mind.

What is used by the A.I.D. to monitor the game is a 'Menace gauge'; What this does is monitor the tension felt by the player and uses three variables to do so, proximity to the player but also the line of sight. This is coupled with an A.I. with a job system that gives the alien prioritized position to search for the player, and a behavioural tree system drives the xenomorph itself, but what is interesting about this is that some nodes in the behaviour tree are locked until further into the game before they are unlocked, giving the Alien a sense of adaptability and difficulty progression organically. The game is a harrowing cat and mouse between the player and the xenomorph that goes on for hours on end, and this was made possible with the A.I. system at the heart of the game.

Generating a Narrative Experience using A.I

The idea that we can use A.I. to generate a certain narrative experience is not something that we could have conceived at the start of video games, but it has eventually happened, and a good example is the 2014 action-adventure video game Shadow of Mordor.

Nemesis and Kuva Lich systems

The game Shadow of Mordor uses a nemesis system, in which bosses have a hierarchical tree where they get promoted after each meeting with the player where the player is defeated. Creating dynamic stories of vengeance and betrayal, the developers also gave unique traits to each boss, making the encounters more memorable.

Another example that takes a more straightforward approach to the nemesis system is the popular MMORPG Warframe. With update 26.0 was introduced a new system called Kuva Liches. The idea involved is the same as the nemesis system and uses the same features, with each character having different names, traits, and a base system(planet) as their headquarters. These liches use an anger meter to measure when it is possible to attack the player; this meter fills up whenever the player decides to eliminate the off-springs of the leader or its followers. Once the meter is full, the chances of being ambushed by the Lich are more than likely to happen, and in that scenario, a fight between the Lich and the player is set up. In the unfortunate scenario where the player is defeated, the Lich proceeds to evolve and gets promoted in the hierarchy, getting stronger each time as a result while also taunting the players whenever possible and also increasing the tax on the player's rewards, robbing him sometimes from rare or important items, creating this hateful relationship with the player, in the case where the player manages to defeat his Lich, the possibility to bring him to the player's side is available as an option among the three options given to him. They can both fight off other groups of enemies, there exist currently two types of this same system in the game, Kuva Liches, and Sisters of Parvos.

Narrative Design

With narrative design in mind, inconveniencing the player could be better to deliver onto a narrative. In the game The Last Guardian, where you and your pet friend are supposed to escape the confines of an isolated castle, you are supposed to tell your pet companion what he needs to do, giving him commands, but he does not always answer to the player's commands, giving the impression that he is an independent agent. The game creates a beautiful story that replicates the mundane and frustrating elements of taking care of a pet, but this only helps increase the bond between the player and the creature.

Head Verbs and Modular Storytelling

The problems with creating robust and creative A.I.s are numerous and often related to input limitations, recognition, and systems management.

In his talk "The future of Storytelling," Jesse Schell talks about how games need to start using head verbs; that being, text, voice, and speech recognition, if they want more robust avenues for storytelling. Similarly, Kevin Levine in "Narrative Legos" tells us to understand the structure of modular storytelling so that we know how to manage director A.I.s and systems.

The problems with extensive narratives have been spoken about extensively in books like "Hamlet on the Holodeck" by Janet H. Murray and an "Interactive Storytelling" by Chris Crawford, and for now, we are still using very static systems to deliver a story in video games.

One attempt at creating more dynamic narratives can be seen in the game "Façade," a game where you play as a friend of a couple having a dispute. The A.I.D. in the game crafts a narrative by intelligently choosing the next story beat based on your interactions and the need to satisfy an overarching arc. The game also has a text parser that allows the system to recognize the player's inputs and a reactive planner that controls the moment-to-moment performance of the characters, dynamically blending multiple simultaneous behaviours. It is a mix of "F.E.A.R," "Alien Isolation," and "The Last Guardian," and has multiple outcomes that play out based on your decisions. It is a very crude and basic system, but it demonstrates a template of what is to come or might be made on a larger scale in the future.

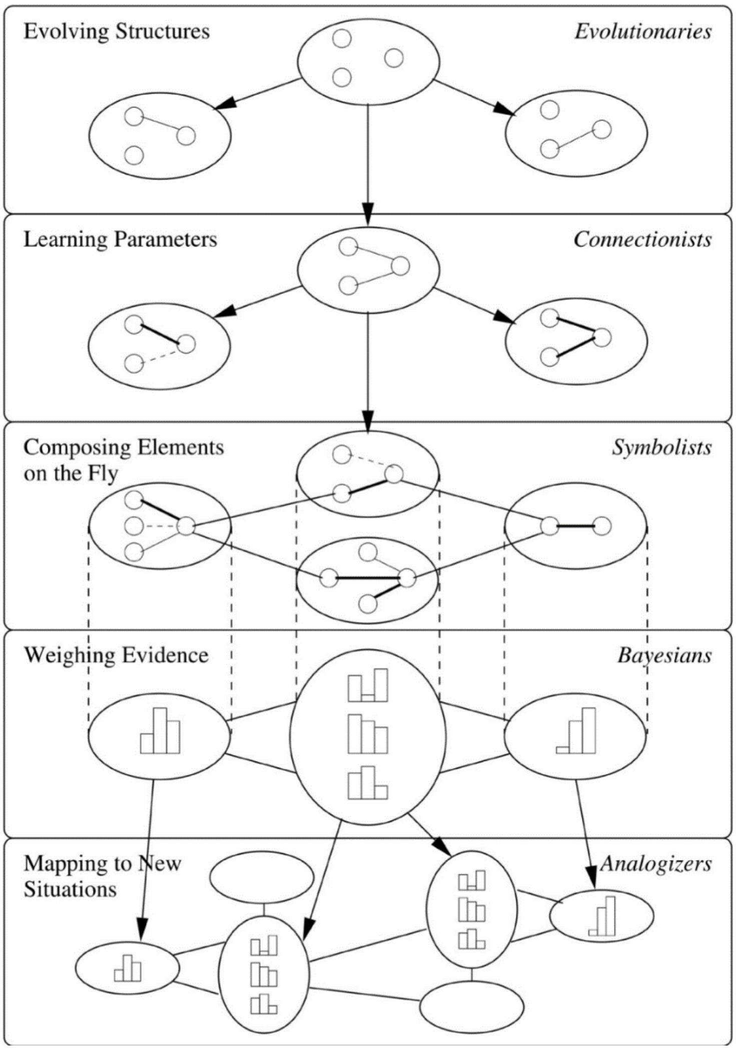
Machine Learning tribes and E.A.s

In his book The Master of Algorithms, Pedro Domingos argues that the future of A.I. is about learning. The idea is that we cannot rely on static A.I. systems to create, but we must enable A.I. systems to learn for themselves if we are ever going to push towards new horizons. Domingos is talking about a much broader project than just games, but as we have seen before, games can play an instrumental part in realizing that future. According to Domingos, machine learning exists in five tribes, each replicating a natural process in the real world.

There is Inductive Reasoning or "Symbolism," which focuses on the premise of inverse deduction; rather than starting with a premise, symbolists use a set of premises and conclusions to work backwards and fill in the gaps. "Connectionism," which replicates the way our brain works like a neural net, the most famous example being "Deep Learning," these are efficient in image recognition and machine translation. "Evolutionism," which replicates natural selection, focuses on applying ideas of genomes and D.N.A. in the evolutionary process to data processing. "Bayesian-ism," which deals with probabilities. "Analogies" try to model how we think creatively.

Figure 1.9

Synthesis of all five machine learning tribes into a single algorithm by Pedro Domingos in "The Master Algorithm."

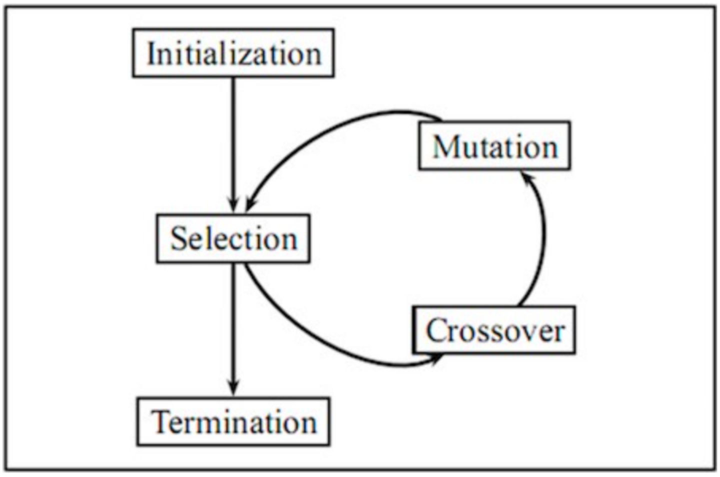


Games can and have been used to illustrate these domains, much like how probability theory came out of the study of games and how poker can show how Bayesian probability works through updating prior probabilities.

In his book, Togelius states: "If we are going to realize the vision of A.I. driven games, we will need A.I. that can adapt its behaviour, learn, understand, and even create new games." He uses this to introduce how machine learning is used in games and expands on Evolutionary Algorithms. Evolutionary Algorithms (E.A.s) replicate evolution by creating variation, a way to store hereditary characteristics, and a selection funnel called a fitness function.

Figure 2.0

Process of an Evolutionary Algorithm



Simply put, in an E.A., fitter members will survive and proliferate, while unfit members will die off and not contribute to the gene pool of further generations, much like natural selection. A study demonstrated E.A.s. An example of this can be seen in games like "Galactic arms race," which creates weapons using a system that selects for weapons based on their usage. He also talks about neural nets or reinforcement learners and how they use backpropagation to teach A.I. how to act. Neural nets can also be used to teach A.I. to recognize faces and cars, and compose music.

A study conducted using these concepts emerged in 2019 by an Open A.I. team. They state, "Through multi-agent competition, the simple objective of hide-and-seek, and standard reinforcement learning algorithms at scale, we find that agents create a self-supervised auto-curriculum inducing multiple distinct rounds of emergent strategy, many of which require sophisticated tool use and coordination. We find clear evidence of six emergent phases in agent strategy in our environment, each of which creates a new pressure for the opposing team to adapt; for instance, agents learn to build multi-object shelters using moveable boxes, which in turn leads to agents discovering that they can overcome obstacles using ramps. We further provide evidence that multi-agent competition may scale better with increasing environment complexity and leads to behaviour that centres around far more human-relevant skills than other self-supervised reinforcement learning methods such as intrinsic motivation.

Finally, we propose transfer and fine-tuning as a way to evaluate targeted capabilities quantitatively, and we compare hide-and-seek agents to both intrinsic motivation and random initialization baselines in a suite of domain-specific intelligence tests." the studies main goal was to teach a group of agents to learn the game of hide and seek and adapt to environmental changes".

Another simpler example of a game that uses these concepts is "Black & White" by Peter Molyneux. In the game, you are a god tasked with getting denizens of the land to worship you, and you are eventually given a creature. What is fascinating about it is that the player can teach it, and it will then think and act based on what was taught by the player. Preachers can learn facts about the environment, how to do tasks, and which strategies to apply in certain situations; after a creature completes a task, the player can use a mouse to stroke the creature affectionately or give it one or more slaps, much like how reinforcement works with a child, this system allows players to train the creature for their desires, the creature also has metrics by which to craft an internal life of beliefs and desires based on the information it has gathered.

Machine Learning, a new frontier in A.I. research

Machine learning is the new frontier of artificial intelligence research, and if what is said by Togelius is to be right, and to think that A.I. and games are fundamentally intertwined, it is interesting to think of the many ways machine learning can either enhance games or be further studied by using games. Games, however, are built by rules and generate systems, so understanding the systemic structure of games is important to close this loop. Realtime strategy games show some interesting things about managing A.I. systems at higher levels of resolution. For example, the game "Total War Shogun" has a finite state machine that applies to groups and manages their action following precepts from Sun Tzu is the art of war. For example, if they outnumber the opponent ten to one, they surround them, whereas they divide them if it is two to one; there is also unit A.I., combat A.I., and campaign A.I. that exist at different resolutions, showing how A.I. needs to be integrated at every level of the game. In his article on the subject of the design goals for the system, the researcher Tommy Thompson says, "With these three systems, the core tenets of the A.I. of Total War are in place: to manage individual units while retaining cohesive behaviour, to group units of troops units together in a manner that is responsive, reactive and challenging towards players and to create strategic opponents with personality."

Another game designer who appreciates the use of unconventional A.I. methods is "The Sims" creator Will R. Wright. "The Sims" has a system called smart terrain, where the objects in the world signal their affordances and their uses to the sims inhabiting the world. This is interesting because it shows us how we can conceptualize the play space as a space of narrative and gameplay possibilities that signal things to different agents based on their internal desires in "SimCity" a method to map out domains called influenced maps was used that can be analogized to the function of a city. Will Wright's conceptualization of these games came out of books like "A Pattern Language" by Christopher Alexander and "Urban Dynamics" by Jay Forrester, showing his fluency in understanding the emergence and the dynamic of urban living. SimCity even starts to model rudimentary versions of phenomena like urban decay, illustrating how the patterns of reality can be laid bare in simulation form.

Figure 2.1

The influenced map method.

Notes: This graph models rudimentary elements of urban decay

Creativity in A.I. and Procedural Generation to create Narratives

If one questions the relation of this to the future of A.I., the answer is very subtle and deeply speculative. In his GDC talk, the nature of order and game narrative, Jesse Schell argued that an understanding and recognition of patterns is what might drive A.I. to new realms of creativity; he further argues that people like Christopher Alexander, who tries to map the many patterns of reality will be as instrumental to the coming century as Einstein was to the previous one. What is interesting is that Jesse thinks A.I. will need a repository of patterns to learn from and a way to generate new ideas, both true and beautiful, dynamically. What is also fascinating is that Jesse thinks of systems and narratives as one dynamic structure and that there is no real distinction. When we look at procedurally generated narrative systems like "Dwarf Fortress" and "RimWorld", we get a better sense of how dynamic narrative and systems converge; we are generating rules that, in turn, generate stories. "RimWorld" gives you a choice of different narrative directors that change how the game's systemic events can transpire. This shows the power of A.I. tools to manage gameplay and narrative possibilities.

Procedural Generation

Procedural tools in games are now fairly pervasive, with games like “Stardew Valley,” “Minecraft,” and “Spelunky” using these systems to craft randomized levels for dynamic play. This tradition goes back to a certain game.

Rogue was the first step into automating the job of creativity itself. Inspired by the dynamic storytelling of Dungeons and Dragons, the designers decided to use procedural tools to create the same sensation of novelty and adventure; the algorithm first generates a dungeon, then divides a dungeon into segments, then creates rooms and marks them as visited as it proceeds through them, and then creates connections for traversal, it then adds items and monsters to create gameplay in this procedural world, suffice to say, this created the roguelike genre, characterized by permanent death and novel strategies for emergent play every time you engage with it, showing A.I. can be instrumental in guiding new design. Some other games have implemented it on a much larger scale. Games like “No Man's Sky” and “Elite Dangerous” have been able to generate a world on a scale so large it represents our Galaxy on a one-to-one scale

Figure 2.2

A small portion of start in the Norma Arm Region of our Galaxy

Notes: As of January 20, 2022, only 0.05%, or exactly 222,083,678 unique star systems have been

discovered

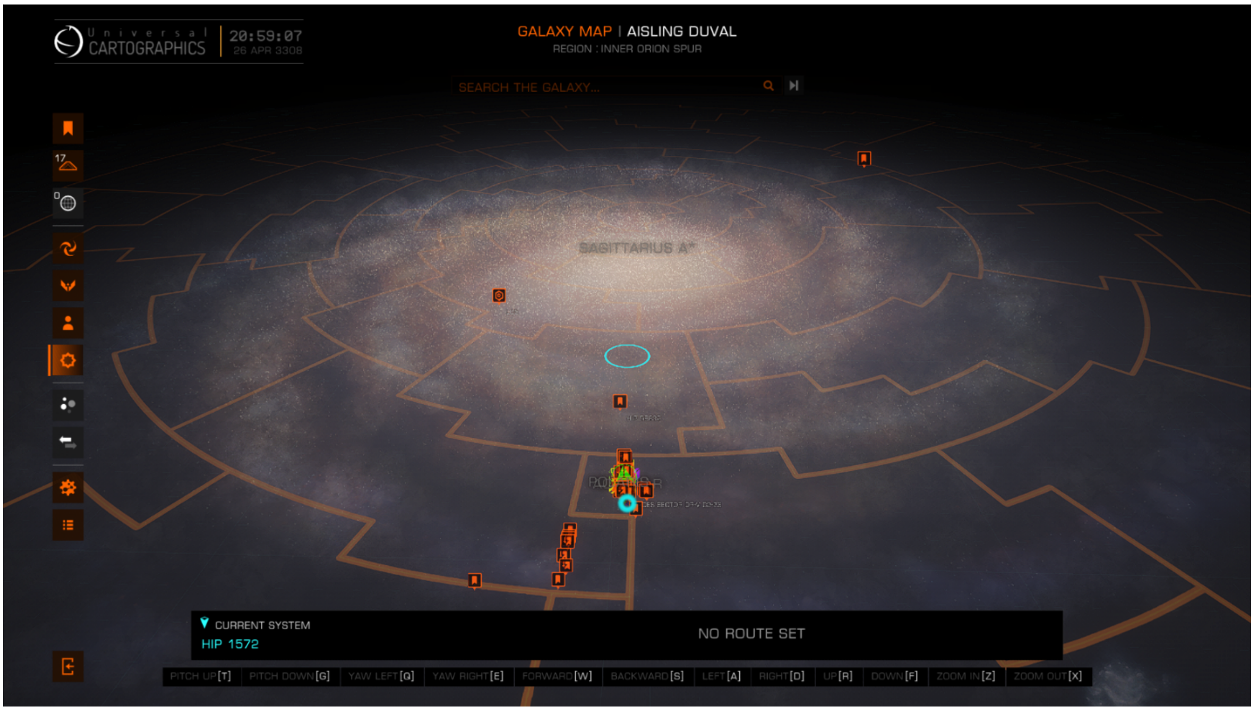
With how big the game becomes, it feels like the player is living in an actual universe, and this method can be seen in games like Minecraft and No Man's Sky. Elite used a seed value to dynamically generate areas of the universe without needing to store thousands of star systems. The seed encodes the emergent space and constructs a new reality under a simple set of guidelines. In the developer's words: "In Elite: Dangerous, when we are generating a system procedurally, each planetary system is formed from first principles. Bodies are gradually aggregated over a very long simulated time from available matter, taking into account their chemical composition. Depending on the angular momentum, this might begin to form into a single central body, or multiple co-orbiting bodies.".

As the gases collapse together under the force of gravity, matter tends to orbit these bodies in protoplanetary discs, which further coalesces into smaller bodies within those discs. Tidal forces, orbital resonances, and gradual accretion of mass gradually change their orbits, causing collisions, collapses, and close encounters – which in turn, bodies might capture each other or fling each other into new orbits or out of the system altogether.

At some point, the stars in the system ignite one by one, and the resultant stellar winds gradually drive off the lighter non-gravitationally bound gases.

Figure 2.3

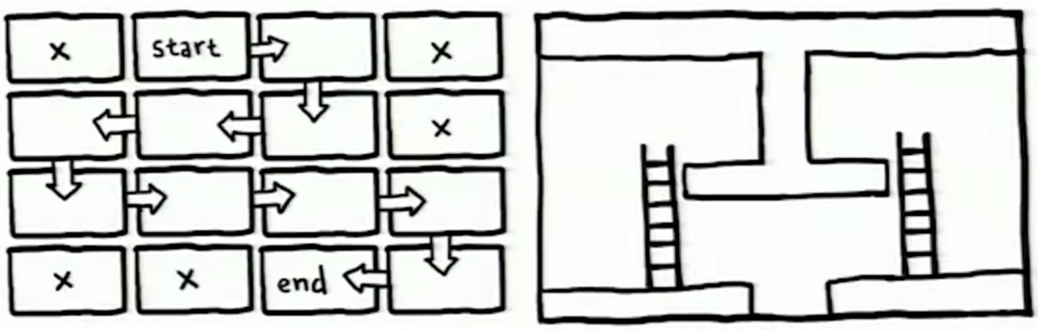
Stellar Forge results, 400-billion-star systems in our galaxy



More recent forays into procedural generation, like "Spelunky", combine random and authored elements to ensure the levels themselves have some coherence and flow. In his book, Derek Yu explains how procedural generation in Spelunky works, crafting levels and blocks with connections between them that always have a clear path to the exit. It then layers additional enemies and items per distributive rules, and the result is something that feels both random and scripted.

Figure 2.4

Level Generation of Spelunky

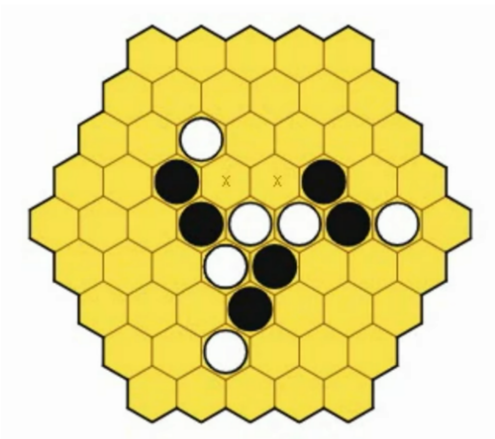


Artificial Intelligence, a game designer

With procedural generation, we get a glimpse into how A.I. itself can be used to design games, which comes with the question of how we get A.I. to exhibit creativity. Cameron Brown used an automated system called Ludi to create the board game Yavalath which has a surprising amount of depth.

Figure 2.5

Yavalath puzzle



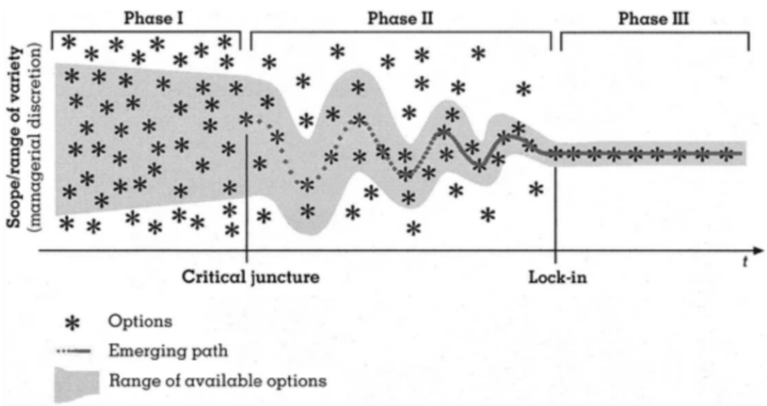
Similarly, by using Evolutionary Algorithms and a store of database of games like chess and Go, Michael Cook created a different system, "Angelina", that creates bizarre but fascinating worlds that show the inkling of some kind of artificial creative spirit, this brings up an interesting question of authorship, at what point do humans end and A.I. begins, there exist programs that have fooled people into thinking that computer constructed musical productions are authentically human, and so, it may only be a matter of time before A.I. itself is nominated for awards in game design.

A bottleneck in the future of A.I. in game design

In his book entitled "Designing for A.I.", Julian Togelius argues that game design today is stuck with principles developed when A.I. was at a rudimentary stage. He states, "At conferences, I would try to convince game designers that their company stood to gain from using new ai methods, with the response being that it was unnecessary." For example, why do we need Robust A.I. in a game like "Mario Kart" when we can just use rubber-bending techniques to create an illusion of a scripted race? The problem with this thinking, according to Togelius, is that our design practice of today is a vestige of the past and needs to adapt to new designs. This is something referred to as path dependence, where past events have fixed us on a path to a specific future.

Figure 2.6

Example graph of path dependency



Notes: The graph depicts how the stream of available options thins, locking us in a specific path

Because A.I. was extremely simple, games were designed around simple patterns, whether it be the simple pattern of enemies in a game like Mario to how enemies in Doom are transient and meant to be gunned down quickly.

This can be seen more clearly in certain design patterns. For example, why are boss fights simple pattern recognition tasks that require players to do the same thing repeatedly, rather than being a more dynamic affair like in the case of Alien Isolation? Why are dialogue options based on canned responses rather than the genuine internal states of different characters, and why is difficulty scaling done on a quantitative basis rather than more intelligent A.I. that responds to the player. This position does not stand without criticism, as one could argue that it is unnecessary to create compelling gameplay or that this betrays a fundamental misunderstanding of how difficult and expensive it is to design games with A.I. in mind.

In the GDC talk "What do designers want from A.I.", a panel of designers lay out what they want from an A.I. driven future; they outline things like how they want characters that recognize the player and respond to our actions that have flaws and vulnerabilities that they act on and have meaningful and dynamic storytelling abilities. Raph Koster explained this best when he said that we have reactive props, not actual characters, and we need to craft dynamically simulated universes that, in the vein of Warren Spector's thoughts on the subject, are rooted in non-combat A.I.

This brings us back to the connection between mechanics and meaning, systems and storytelling, and what is a future with non-violent verbs and the disavow love of the path dependence of our design sensibilities look like.

Often, games that craft deep emotional experiences do not need sophisticated A.I.; the A.I. of Yorda and Ico in the game "The Ico and the Shadow of the Colossus" was a very simple set of behaviours. However, by having her helpless and tying our well-being to hers, the game elicits a sense of selfless concern in the player; without the need for complex tools, Ellie, in “The Last of Us”, is invulnerable to the virus in the game, a game contrivance for gameplay's sake and the connection instead was forged during the cutscenes.

A dream

A storytelling future that pushes towards the holodeck might combine tools like this. Elizabeth in BioShock Infinite had a team of designers dedicated to her creation with animation inspired by the work of Disney, they also had A.I. techniques that prompted her to do things based on where the players looked, even the game's narrative sequence is a dynamic flow, however, what if we combined this with the gameplay of Ico as well, where we actually have to care for her, creating an embedded narrative with meaningful mechanics, now, what if we add Façade's or Alien Isolation's narrative directors that recognize her actions in the game to tailor a narrative experience that is dynamic every time you play it, what if like Jesse Schell argued in his talk, Elizabeth now recognized our text, voice, and mannerisms, and the story being told changes in accordance with our actions in the vein of something like Detroit become human, what if A.I. was smart enough to craft not just dynamic gameplay scenarios for us, but also script an opera of adventure, hope and love in the vein of mass effect, except it is a procedurally generated cosmos that has a scale of No Man's Sky, what if we can craft a universe of play and storytelling that can reflect back on ourselves in the real world and help us be better people in our day to day.

Chapter 5: Conclusion

A.I. is the future of games, but games are also the future of A.I.; the final synthesis is that they can both illustrate something about Intelligence itself, but more than that, they can speak to who we are. We may create A.I. that can usurp us or almost entirely replace us, but if harnessed intelligently, A.I. can be used to craft deeper games, more meaningful storytelling experiences and can be used to examine ourselves. The history of A.I. in games is one of close collaboration, intertwined inspiration, and symbiotic synergy, there is no reason to assume that the future will be any different.

References

Abercrombie, J. (2014). Bringin BioShock Infinite's Elizabeth to Life: An A.I. Development Postmortem. *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1020831/Bringing-BioShock-Infinite-s-Elizabeth

Bevilacqua, F. (2013). Finite State Machine: Theory and Implementation. *Artificial Intelligence*. Retrieved from https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867

Booth, M. (2009). The A.I. Systems of Left 4 Dead. Valve. Retrieved from https://steamcdn-a.akamaihd.net/apps/valve/2009/ai_systems_of_l4d_mike_booth.pdf

Bourse, Y. (2014). Artificial Intelligence in The Sims. Retrieved from https://team.inria.fr/imagine/files/2014/10/sims-slides.pdf

Cameron, B., Edward, P., Daniel, W., Simon, L., Cowling, P. I., Philipp, R., . . . Simon, C. (2012). *A Survey of Monte Carlo Tree Search Methods* (Vol. 4). Retrieved from http://www.diego-perez.net/papers/MCTSSurvey.pdf

Campbell, J. (2017). Bringin Hell to Life: A.I. and Full Body Animation in 'DOOM'. *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1024186/Bringing-Hell-to-Life-AI

Crawford, C. (2017). *Chris Crawford* (2nd ed.). New Riders. Retrieved from https://www.amazon.com/Chris-Crawford-Interactive-Storytelling-2nd/dp/0321864972/ref=sr_1_1?crid=N2DNDT33TUC3&keywords=On+interactive+storytelling+chris+crawford&qid=1652283555&sprefix=on+interactive+storytelling+chris+crawford%2Caps%2C838&sr=8-1

Dohta, T., Fujibayashi, H., & Takizawa, S. (2017). Change and Constant: Breaking Conventions with 'The Legend of Zelda: Breath of the Wild'. *Game Developers*

Conference. GDC. Retrieved from https://gdcvault.com/play/1024562/Change-and-Constant-Breaking-Conventions

Editorial Team. (2018). The Five Tribes of Machine Learning. *World Of Analytics*. Retrieved from http://worldofanalytics.be/blog/the-five-tribes-of-machinelearning

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics, IEEE*, 100-107. doi:10.1109/TSSC.1968.300136.

Hoge, C. (2018). Helping Players Hate (or Love) their Nemesis. *Game Developers Conference.* GDC. Retrieved from https://www.gdcvault.com/play/1025150/Helping-Players-Hate-(or-Love)

Isla, D. (2005). GDC 2005 Proceeding: Handling Complexity in the Halo 2 A.I. *Programming*. Retrieved from https://www.gamedeveloper.com/programming/gdc-2005-proceeding-handling-complexity-in-the-i-halo-2-i-ai;

Koste, R., D. M., Lemarchand, R., McWilliams, L., Falstein, N., & Hunicke, R. (2018). A.I. Wish List: What Do Designers Want out of A.I.? *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1024900/AI-Wish-List-What-Do

Levine, K. (2014). Narrative Legos. *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1020434/Narrative

Michael, M., & Andrew, S. (2003). Facade: An Experiment in Building a Fully-Realized Interactive Drama. *Game Developers Conference.* GDC. Retrieved from https://faculty.cc.gatech.edu/~isbell/reading/papers/MateasSternGDC03.pdf

Molyneux, P. (2000). Excerpt: Peter Molyneux introduces Black & White. *Game Developers Conference.* GDC. Retrieved from https://www.gdcvault.com/play/1014842/Excerpt-Peter-Molyneux-introduces-Black; https://www.youtube.com/watch?v=69HTviSeQeE

Murray, J. H. (2017). *Hamlet on the holodeck: The Future of Narrative in Cyberspace.* The M.I.T. Press. Retrieved from https://www.amazon.com/Hamlet-Holodeck-Future-Narrative-Cyberspace/dp/0262533480/ref=sr_1_1?crid=KAIZVFX036A&keywords=Hamlet+on+the+holodeck+Janet+Murray&qid=1652283415&sprefix=hamlet+on+the+holodeck+janet+murray%2Caps%2C288&sr=8-1

Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. *Game Developers Conference.* GDC. Retrieved from http://alumni.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf

Pedro, D. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.* Basic Books. Retrieved from https://www.amazon.com/The-Master-Algorithm-audiobook/dp/B014X01SS0/ref=sr_1_1?keywords=the+master+algorithm&qid=1652282204&s=books&sprefix=The+master+al%2Cstripbooks%2C288&sr=1-1

Schell, J. (2013). The Future of Storytelling: How Medium Shapes Story. *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1018026/The-Future-of-Storytelling-How

Soni, D. (2018, February 18). Introduction to Evolutionary Algorithm: Optimization by Natural Selection. *Towards Data Science*. Retrieved from https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac

Thompso, T. (2018). The Road To War | The A.I. of Total War. *Game Developer*. Retrieved from https://www.gamedeveloper.com/programming/the-road-to-war-the-ai-of-total-war-part-1-

Thompson, T. (2017). The Perfect Organism: The A.I. of Alien: Isolation. *Game Developer*. Retrieved from https://www.gamedeveloper.com/design/the-perfect-organism-the-ai-of-alien-isolation; https://youtu.be/Nt1XmiDwxhY

Togelius, J. (2018). *Playing Smart, On Game, Intelligence, and Artificial Intelligence.* The M.I.T. Press. Retrieved from https://www.amazon.com/Playing-Smart-Intelligence-Artificial-Thinking/dp/0262039036/ref=sr_1_1?crid=GG3JRLXVMTMR&keywords=Julian+Togelius&qid=1652282298&sprefix=julian+togelius%2Caps%2C186&sr=8-1

Wexler, J. (2002). *Artificial Intelligence in Games.* The University of Rochester. Rochester, NY 14627: University of Rochester. Retrieved from https://www.cs.rochester.edu/~brown/242/assts/termprojs/games.pdf

Wright, W. (2003). Dynamics for Designers. *Game Developers Conference.* GDC. Retrieved from https://www.gdcvault.com/play/1019938/Dynamics-for

Yu, D., & Hull, A. (2011). The Full Spelunky on SPELUNKY XBLA. *Game Developers Conference.* GDC. Retrieved from https://gdcvault.com/play/1014436/The-Full-Spelunky-on-SPELUNKY