PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 05/2022

Notes on AI in Games and Storytelling

Bruno Feijó
Antonio L. Furtado

(Editors)

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 05/2022 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena December, 2022

Notes on AI in Games and Storytelling

Bruno Feijó, Antonio L. Furtado (eds.)

PUC-Rio, Departamento de Informática, Rio de Janeiro, Brasil
{ bruno, furtado}@inf.puc-rio.br

Abstract: This monograph consists of three short papers, written and orally presented, as part of the discipline INF2609 on AI in Games and Storytelling during the second semester of 2022. The graduate students enrolled in this discipline elaborated on the papers.

Keywords: Artificial Intelligence, Digital Entertainment, Games, Storytelling.

Resumo: A presente monografia consiste de três artigos curtos, escritos e apresentados oralmente como parte da disciplina INF2609, sobre o tópico IA em Jogos e Narração de Estórias, durante o segundo semestre de 2022. Os artigos foram elaborados pelos alunos de pós-graduação matriculados na disciplina.

Palavras-chave: Inteligência Artificial, Entretenimento Digital, Jogos, Narração de Estórias.

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br

Introduction

In the *Game AI and Storytelling* course, we study the application of Artificial Intelligence (AI) techniques in digital games and virtual environments simulating reality in real-time. We start learning player control techniques that influence intelligence, interactivity, interface, and gameplay in the first part of the course. In the second part of the course, considering the importance of narrative in games and simulations, we focus on Interactive Storytelling. We also discuss the limitations and consequences of AI in interactive, immersive, and ubiquitous applications.

In the academic period corresponding to the second half of 2022, we offered the discipline with the following topics:

PART I
- Game AI, Design, and Cognitive Science;
- Logic (1st Order, non-classical, Frame Problem);
- Search, Game* & Waypoints;
- FSM & Behavioral Trees for Games;
- Planning (STN, HTN, Non-deterministic);
- Machine Learning & Real-Time Prediction;
- Movement Control and Planning;
- Gamification;
- Game Analytics;

PART II
- Fundamentals of Narratology;
- Characters, Plot, Theme;
- Folk Genres (Types and Motifs, Functions, Patterns);
- Executable Conceptual Specification;
- Drives, Attitudes, Emotions, Collaboration, and Competition;
- Types of Players;
- Semiotic Relations, Variants, Network Organization.

The articles contained in this monograph were written by the graduate students enrolled in the discipline.

Table of Contents

Game Analytics feature-based input for a Casual game difficulty modeling

João Victor Farrulla Darze

Abstract. This short paper aims at designing a process in which game data is gathered and used to shape a balanced gaming experience. We will focus on building the game features to understand player behavior and also use it as input to a difficulty modeling system that balances mini-games variables that shape level difficulty.

Keywords. Game Analytics, Game Balance, Player Behavior.

1. Introduction

Nowadays the gaming market is full of amazing personalized experiences from a wide range of aspects. All the way from game recommendations from platforms like Steam and Origin to in-game variances of output based on players' behaviors during gameplay.

This is possible due to the intense data-driven moment we are living in our society. Many interesting applications have been powered by data to generate experience enhancements that were previously impossible. Adding technologies like machine learning to this data broadens the interface between humans and their applications.

This work will establish a very specific game in which this framework will be applied. Can be seen as a tangible case study so entanglements, flow, and results are better understood.

In section 2 a game explanation will be given so the reader can easily assess the environment where the analysis would take place and set the outcome expectations properly. Section 3 discusses the literature on game analytics and its usage in the gaming industry. Section 4 concludes.

2. The Metapals game

The game is called Metapals and is based on the old times of Tamagotchi, a small device containing a virtual pet that players should try to keep alive and well. The game contained basic actions like feeding, playing, and putting it to sleep.

This new version of a virtual pet game you need to take care of and enjoy with it now lives in a web browser. Using a browser extension to load the game and its functionalities in users' browsers the capabilities of interaction with web pages are huge. Bringing a virtual pet to a user every day browsing the web experience is meant to generate a sense of companionship and an emotional bond.

More than simply existing in the players' browser (Figure 1) by walking around the screen, it's possible to interact with it by petting, feeding, and playing with it. Several assets - as in-game items - enable different kinds of interactions. The pet has basic stats that every player should strive to keep balanced and in a safe zone.

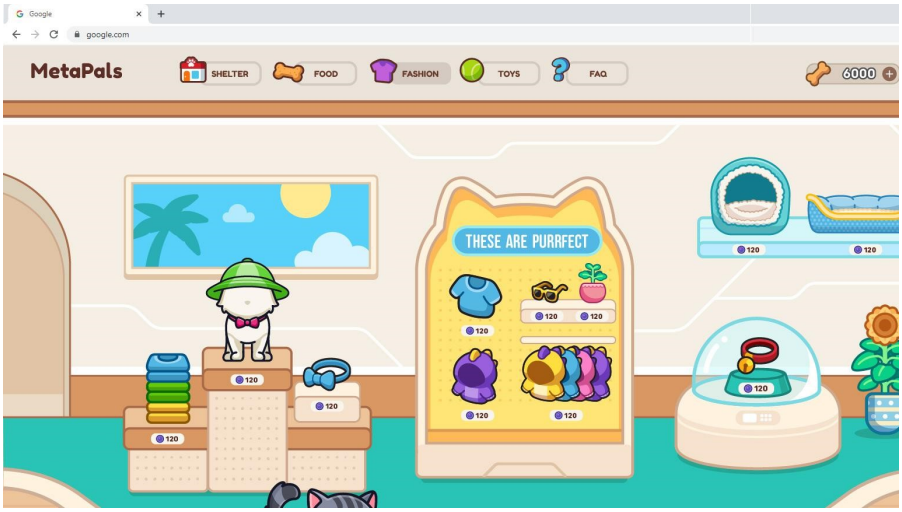

Figure 1: The virtual pet in the browser

Stats	Description
Hunger Metter	It shows how hungry the pet is and helps the player know when to feed it.
Mood Metter	It shows how excited or bored the pet is. Helps the player to know when to play and interact with it.
Health Metter	It shows how healthy is the pet. Helps the player to know what to bring to the vet or give medicine.
Care Index	Is a combined score of all the other stats. It represents the general well-being of the pet.

Table 1. metapal stats

The game also aims at implementing various simple mini games in the future in order to engage players with more options to play in their free time. Mini-games examples are games like temple run, word finder, or whack-a-mole.

3. Modeling Game Difficulty

User Behavior

This critical aspect is tracked by the game features [Lima et. al, 2018], which are several measurable metrics, and events that happen during gameplay. It shows the fundamental decision-making of players.

As an example, we can take a minigame gameplay to analyze. The word finder game takes the player to a Wikipedia page where several hints are given with some time in between so the player can try to find which word is the correct one.

The above-mentioned scene is a simple game that has a good amount of challenge when it comes to balancing the game's difficulty. By looking at the right features we can come to a more defined understanding of how players may perceive the difficulty imposed. The correct questions we ask are like “is 5 seconds apart from a

hint to another enough?" or "as the player gets right should the timer for the next round be reduced linearly?". All such questions can be answered by a measurable single feature or a set of features.

Feature Generation

Here we present the game feature list generated (table 2) where most of the word finder mini-game features take place.

Feature ID	Description
F1	**Average** number of hints needed to find the word
F2	**Average** time spent to find the word after hint #1
F3	**Standard Deviation** of time spent to find word after hint #1
F4	**Average** time spent to find the word after hint #2
F5	**Standard Deviation** of time spent to find word after hint #2
F6	**Average** time spent to find the word after hint #3 (last hint)
F7	**Standard Deviation** of time spent to find word after hint #3
F8	**Average** time spent to find the word
F9	**Average** wrong try during hint #1
F10	**Average** wrong try during hint #2
F11	**Average** wrong try during hint #3
F12	**Percentage** of wins (in relation with *Difficulty*)

Table 2: Features for word finder mini game.

Analysis

Game analytics involves collecting data about how players interact with the game world, analyzing it, then converting the results into useful knowledge [Seif El-Nasr 13].

This should power a system with *Skill-based adaptation capabilities*: techniques that aim to adapt the game mechanics to the user's skill level. In order to keep players engaged and challenged, we need to maintain a

level of challenge that won't overwhelm them with too many difficult tasks [Nguyen, Truong-Huy D. et al, 2015]

4. Conclusion

This short work is an attempt to bring together a game feature-based analysis into difficulty-level modeling. Future works could potentially add and experiment on machine learning models to, on real-time, change dynamically the difficulty to match the skill of a player.

REFERENCES

de Lima, E.S., Feijó, B., Furtado, A.L. Player behavior and personality modeling for interactive storytelling in games. Entertainment Computing, v. 28, p. 32-48, 2018.

El-Nasr, Magy Seif, Anders Drachen, and Alessandro Canossa. "Game Analytics: Maximizing the Value of Player Data." (2013).

Nguyen, Truong-Huy D. et al. "Analytics-Based AI Techniques for a Better Gaming Experience." (2015).

Personality Traits and User Behavior as a Metric on Game and Storytelling Development

Jose Rodrigo Vilca Vargas

Abstract. The constant growth and expansion of the industry of gaming has positioned big franchises and AAA games in such competitive positions that it is almost impossible for new or independent games to compete. That is why new mechanics and strategies such as the use of user behavior and personality traits are employed to maximize the enjoyment of the player in new games.

Multiple approaches have been used to collect the data necessary to define a personality type of player and how it affects and changes the game and the storytelling of it. We explore the definition of archetypes and characteristics to categorize players, as well as some limitations and obstacles that we must have to be aware of in the collection of this data.

Keywords. Game analytics, Games, Personality, Personality Computing, Personality Detection, Personality Modeling, Personality Traits, User Behavior.

1. INTRODUCTION

The industry of gaming is in constant growth and expansion through the years, becoming a highly competitive industry [Bicalho et al. 2019] where new games and independent ones stand out in comparison to established franchises as well as big AAA games.

It is because of the highly competitive background in game development that new games must search for new strategies to innovate and improve in-game mechanics as well as storytelling to create an attractive, high quality game that new players would enjoy.

One of the approaches to perform this innovative system is based on personality computing, an interdisciplinary study field that focuses on integration of personality psychology theories with computing systems [Dhelim et al. 2022]. This approach is being studied in terms of user behavior as well as the extraction of personality traits through the data analysis of in-game behavior [Bicalho et al. 2019] to define the archetype of the player as well as adapting the storytelling [de Lima et al., 2020] to match with the personality of the player.

The personality traits are commonly extracted from psychological tests and measurement scales that are usually to be composed of several questionnaires and interviews. However, applying in-game big questionnaires could result in loss of interest of the players. For this reason, to achieve the goal of a personality based experience, in-game personality traits extraction is needed.

In-game personality traits extraction and user behavior analytics could be performed through the analysis of decisions and behavior of the player through the main story and side quests [de Lima et al. 2018][de Lima et al. 2020]. Thus, once extracted the players traits, the storytelling, NPC's behavior, world challenges, among other aspects, can vary towards the best fit for the players traits and their behavior.

In this document, we perform a brief analysis on some of the personality traits collection methods found in the literature, the possible obstacles and limitations that we must be aware of when trying to collect this information and the possibility of integration of some personality models commonly used nowadays. At the end, we include a section exposing our concluding remarks.

2. PERSONALITY TRAITS COLLECTION ON LITERATURE

The literature on personality traits and archetypes is heavily marked by some characteristics of the players. In this section we will see a taxonomy of player archetypes as well as the definition of traits based on characteristics and personality.

Archetypes Taxonomy

A well known taxonomy of players is based on two principal axes [Bartle, 1996] players/world and interacting/acting. It classifies players, depending on whether they are more interested in the interaction or in the active action, over the world or over the other players. These axes give us a total of four possible archetypes (Figure 1).

- Achievers: focused on mastering the game, on the rewards it has to offer.
- Killers: focused on acting on other players, or NPCs, most of the time showing their superiority over them.
- Socializers: focused on interacting and talking with other players, or NPCs
- Explorers: focused on interacting with the world, the game environment.

Therefore, depending on the game genre these archetypes could be reduced to two or three.

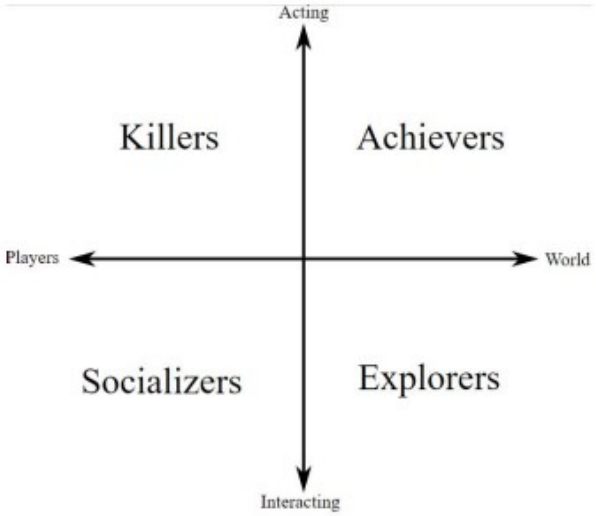

Figure 1. Bartle's taxonomy

An expansion is proposed in order to make a distinction between casual players, those who use to play in mobile devices during their free time for fun or taking a break of their daily activities, and hardcore players that dedicate a big part of their time to play different game genres, having a good knowledge of the industry, and spending more money than the average users [Bicalho et al. 2019] -- resulting in eight possible archetypes (Figure 2), considering world/players, acting/interacting and casual/hardcore axes.

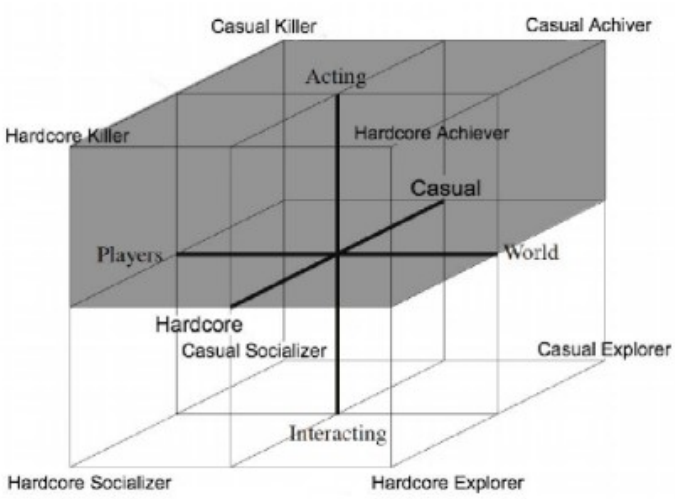

Figure 1. Bicalho et al. [2019] proposal

Furthermore, the dedication level of a player can be as well evaluated by fifteen characteristics [Adams & Ip, 2002].

- Technologically savvy: Dedicated gamers usually have more interest in new technologies, mainly the ones related to the game industry.
- Have the latest high-end gear: Dedicated gamers will be up to the latest hardware news on the industry, acquiring the best consoles and computers. They usually own, or have owned, older game platforms.
- Willingness to pay: Dedicated gamers usually don't wait for the promotions or special offers to get the games they want. They need to play it, and even buy products related to their favorite franchises.
- Prefer violent/action games: Dedicated gamers prefer games that are more violent and action-packed than the market average.
- Prefer games that have depth and complexity: Dedicated gamers prefer games that challenge their knowledge, that make them spend more time trying to beat or master it.
- Play games over many long sessions: Dedicated gamers play regularly and usually spend hours in a single session.
- Hunger for gaming-related information: Dedicated gamers constantly search for the latest game news, previews and reviews, also looking forward to interviews with industry experts, game magazines, books and strategy guides.
- Discuss games with friends online: Dedicated gamers love to discuss about the game industry trends and news, mostly through forums and social media groups.
- Play for the exhilaration of defeating (or completing) the game: A dedicated gamer will play a game for the pleasure of beating it, defeating difficult enemies.
- Much more tolerant of frustration: Dedicated gamers don't abandon games because of frustration. They are used to play difficult and challenging games, helping them to mitigate this possible frustration.
- Engaged in competition with himself, the game, and other players: Hardcore gamers want to feel happy when rewarded after beating a difficult challenge, or by improving their skills.
- Age at which first started playing games: If someone has aged constantly playing video games, since he/she was younger, he/she can be considered an experienced and dedicated gamer.
- Comparative knowledge of the industry: Dedicated gamers are likely to have more knowledge of the game industry trends and new technologies, not just because of their will to search for this type of information, but also because they play various game genres.

- Early adoption: Dedicated gamers seek for over midnight game releases, pre-orders and beta or alpha test events, being one of the first with access to new games.
- Desire to modify or extend games in a creative way: Dedicated gamers sometimes feel the need to create mods, altering from graphic, character skins, to creating new game modes, when the game offers a customization level.

The five factor model (FFM also known as Big Five or OCEAN) has been used to define personality traits [de Lima et al. 2018]. The FFM is a dimensional representation of human personality structure, which claims that, by using five personality traits, it can suitably account for personality diversity. The Big Five factors are:

- Openness: those who are high on this factor are imaginative, curious and open to new ideas. In contrast, those who score low on this factor are indifferent and uninterested.
- Conscientiousness: the ones that display a high degree of this factor are meticulous, efficient and systematic. Who scores low is careless, chaotic and disorderly.
- Extraversion: high scorers are characterized by enjoying social activities. On the opposite side, low scorers are reserved and shy.
- Agreeableness: a high score on this factor characterizes helpful, cooperative and friendly people. In contrast, low scores characterize selfish and hostile people.
- Neuroticism: those who score high on this factor are emotionally unstable, anxious and aggressive. In contrast, those who score low are well-adjusted and calm

Two of the bigger advantages of using the FFM for in-game personality trait extraction are the positive scoring of each factor (Figure 3) where a negative score in a specific factor brings more information than the lack of the positive scoring and the existence of a reduced or simplified questionnaire.

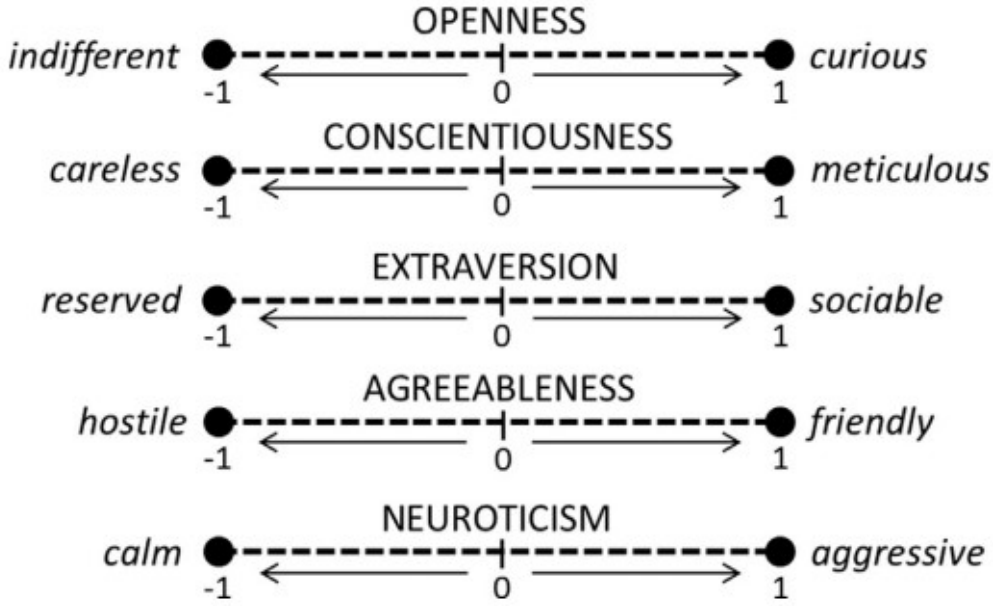

Figure 3. FFM factors represented as behavioral axes.

3. SOME OBSTACLES AND LIMITATIONS

Beyond the technical issues faced when we intend to collect personality and behavioral data of the players, there are many social and cultural difficulties that we must be aware of to perform a better data collection.

Evaluation apprehension is an uneasiness or worry about being judged by others, especially worry experienced by participants in an experiment as a result of their desire to be evaluated favorably by the experimenter or by others observing their behavior. Participants with evaluation apprehension may inhibit reactions that they believe will lead the experimenter to regard them as psychologically unhealthy [APA, 2022]. In this way, players' awareness of being evaluated can produce tension and a change of behavior in the attempt to perform "the better version of themselves", especially if it is not clear how the evaluation results will be used or if the information collected will be private or public.

Thus, the evaluation of the players must be performed discreetly, so the player may ignore or not notice that it is being evaluated. Also, as personality traits and their evaluation are part of the sensitive of the user, how it is being used and how it is going to be stored must be informed to the user who must agree with it to proceed.

The player **characterization** could perform an obstacle in the personality data collection. This characterization could make the player project different values and characteristics to their in-game character. This way, the actions performed by the player will be their idea of what their character would do, that sometimes could be an extreme opposite of their true personality, inducing some incoherences between the personalities of the character and the player itself. This may or may not be an obstacle since it could generate a better immersion in the players roleplaying. Nevertheless. it can be a factor of displeasure in case that the character and the player could become totally incompatible.

In the same way as the evaluation apprehension, some strategies could be performed to collect the true traits of the player. Since the roleplaying attitude may lead the player to perform specific actions in the skin of their character, usually secondary actions as secondary quests or "unimportant" dialogs can be out of the radar of this personification in a way we could achieve the collection of the player and the character traits and perform a weight of both to maximize the satisfaction of the game experience for the player.

Cultural bias is the tendency to interpret and judge phenomena in terms of the distinctive values, beliefs, and other characteristics of the society or community to which one belongs [APA, 2022]. There is no surprise that, in the psychological field, personality evaluations must pass through a rigorous process of validation before being able to be correctly applied in different groups of people around the world. In the same way, the scope of the implications of the players behavior must be aware of the possibility of leading with a cultural bias. For example, Latin American people tend to be much more socially active than Asian people, so it could be usual that the first group create relationships easier than the second group, but this difference does not necessarily mean that the second group must have a higher level of introversion than the first one because they come from a whole different culture background.

Lack of background knowledge can be an obstacle when it comes to taking decisions. When a player must decide to save or not save a NPC friend or has the option to steal from them, the lack of background knowledge or the lack of empathy with the NPC characters can lead them to make a decision that in a real situation they would not take [de Lima et al., 2018].

3. Other Possible Personality Traits Recollection Approaches

Personality is a widely studied concept in psychology and it has several approaches and may use different questionnaires to collect the information needed. Some of them could be used in the gaming industry depending on the desired goal, since they have different evaluated characteristics and results as well as they can show new information that can be used not just in the game industry but also for guaranteeing the players mental health.

The Myers-Briggs Type Indicator (MBTI) introspective self-report questionnaire indicating differing psychological preferences in how people perceive the world and make decisions based on Jung's psychoanalysis [NERIS, 2022]. The MBTI has gained a lot of popularity in recent years through social media, especially among youngsters that like to share their results and take them as an important factor of each other's personalities. The MBTI evaluates five personality aspects:

- Mind: How the person interacts with their surroundings. It can be categorized as **I**ntroverted or **E**xtroverted.
- Energy: How the person sees and processes the information. It can be categorized as Ob**s**ervant or I**n**tuitive.
- Nature: How the person makes decisions and copes with emotions. It can be categorized as **T**hinking and **F**eeling.
- Tactics: How the person approaches to work, plans and makes decisions. It can be categorized as **J**udging and **P**rospecting.
- Identity: Shows how confident the person is with their own abilities and decisions. It can be categorized as Assertive (**-A**) and Turbulent (**-T**).

One of the biggest advantages of MBTI is its popularity among the population, making it easy to collect data from external sources where the users had already been evaluated with their MBTI type.

The Minnesota Multiphasic Personality Inventory (MMPI) is a psychological test that assesses personality traits and psychopathology. It evaluates eight clinical scales: Hypochondria, depression, hysteria, psychopathic deviance, paranoia, psychasthenia, schizophrenia, and mania. [Ruhl, 2021].

The MMPI has an abbreviated questionnaire of 71 items, and it can help with detection of some psychopathologies such as OCD (obsessive-compulsive disorder), paranoid disorder, personality disorders, among others. The use of this questionnaire may offer a huge opportunity in the area of serious games where it can help to perform an early detection of possible mental health issues that, later, must be confirmed by a mental health professional.

4. Final Remarks

In this short work, we summarized some of the approaches to integrate personality traits and user behavior to improve the in-game experience. Such approaches as archetypes and personality traits enlighten the possibility of an unique individual experience for each player attempting for maximum enjoyment.

Since the game industry is a global phenomenon, the research on these issues must be widely expanded so the experiments can be performed in a more varied group of people of different ethnicities and cultures to avoid, as much as possible, that the cultural bias ruins the game experience.

The psychological aspects of the subject or player is a main concern in this kind of research, since we are trying to improve their experience through those same aspects. That is why multidisciplinary research is in need in order to achieve better results.

Finally, we just showed a few of the currently used questionnaires for personality evaluation, but since it has been an area of study in psychology for so many years, there are also many tools for personality evaluation, each of which is based on different theories and enlightens different characteristics about the subject. Research on personality evaluation offers a huge amount of opportunities to collect data and improve the user experience as well as developing serious games that may help with the users mental health.

References

NERIS. 16 Personalities, Our Framework. https://www.16personalities.com/articles/our-theory, 2022 accessed 13/12/2022

APA (American Psychological Association). APA Dictionary of Psychology. https://dictionary.apa.org/, 2022 accessed: 13/12/2022

de Lima, E.S., Feijó, B., Furtado, A.L. Player behavior and personality modeling for interactive storytelling in games. Entertainment Computing, v. 28, p. 32-48, 2018

de Lima, E.S., Feijó, B., Furtado, A.L. Adaptive storytelling based on personality and preference modeling. Entertainment Computing, v. 34, 2020

Dhelim, S., Aung, N., Bouras, M. et al. A survey on personality-aware recommendation systems. Artif Intell Rev 55, 2409–2454 (2022).

E. Adams and B. Ip, "From casual to core: A statistical mechanism for studying gamer dedication," https://www.gamedeveloper.com/business/from-casual-to-core-a-statistical-mechanism-for-studying-gamer-dedication , 2002, acessed: 13/12/2022.

L. F. Bicalho, A. Baffa and B. Feijó, "A Game Analytics Model to Identify Player Profiles in Single player Games," 2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 2019, pp. 11-20.

R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit muds," 06 1996. http://mud.co.uk/richard/hcds.htm , accessed 13/12/2022.

Ruhl , C. (2021, Sept 29). The Minnesota Multiphasic Personality Inventory (MMPI). Simply Psychology. www.simplypsychology.org/what-is-the-minnesota-multiphasic-personality-inventory.html

Designing Behaviors for Games on Limited Systems

Lucas Cordeiro Marques

Abstract. This work presents a reasoning for designing object or enemy behaviors for games on systems with limited computing resources in the form of a developed Atari 2600 game**.** It begins by reviewing the landscape of retro gaming in modern days, then proceeds by explaining some 2600 development concepts, the design of the game in question and how to expand its concepts for more modern systems.

Keywords. Game Design, Game Development, Software Engineering.

1. INTRODUCTION

Retro gaming has been a rising trend in recent years. There are the so-called "mini" console relaunches, like the NES mini or the PlayStation classic. Entirely new consoles with limited hardware have been launched recently as well, like the play.date[1]. Even "fantasy" consoles like the Pico-8[2], which exist only in emulators, have seen some really interesting developments.

All of this goes to show that retro gaming is a popular hobby but, in fact, demo and homebrew scenes have always existed for retro consoles, many times while some of them were still being supported by their respective manufacturers, like the nintendo 3ds[3].

The existence of these communities means that there is room for aspiring game developers to try to make games for retro systems. Tools that allow for easier development for some of these hardwares are constantly being released, like the GB Studio[4], which exports games that can run on original Nintendo Game Boy consoles. With so many new and established tools and growing communities focused on retro game development, it is useful to talk about how one would go about designing a game that fits the limited specifications of older hardware.

Part of what makes these consoles so interesting to play with -- and develop -- are their limitations. Modern machines have large amounts of memory and processing power, giving developers a lot of space to work with, but limited hardware can promote creativity, which can make the developed games really unique.

The most obvious limitations are the things that a player immediately perceives, like graphics and audio. Old systems tend to have very limited image output resolutions, small color palettes and few sound channels to work with. However, less mentioned are the indirect restrictions that these limitations cause, specially those of designing the softwares themselves. A lack of hardware capacity usually means the developers cannot implement too many gameplay mechanics or, in the case of this article, complex AI or behavior systems.

2. MAKING A GAME FOR THE ATARI 2600

One of the most active and prolific development communities is that of the Atari 2600. There are many websites with resources to aid developers in making a game for the platform[5][6]. In this section we are going to discuss the development of a game made by me and two other participants of the Atariando 2.0 event. The event happened over the course of three days, between the 8th and the 10th of November 2022, at PUC Rio.

The theme chosen for the game was selected among a list of various random subjects, and was just the generic word "dinosaur". With this theme in mind, the team brainstormed some ideas and came up with the concept of a dinosaur surviving a meteor shower, which ended up being the game Meteorfall.

The game was developed using batari-BASIC[7][8], a language similar to BASIC that simplifies the development of software for the Atari 2600. The language clarifies some of the limitations of the platform. Among them: the need to read or write directly from and to registers in order to check input or display graphics and play sounds, or the set amount of only 26 freely usable variables—No more can be declared.

So, for example, in order to make our dinosaur move, we read the registers corresponding to the input of the first player, then set its horizontal position accordingly, using a variable that is reserved by the system for this use. This position is the place where we will then render the dinosaur's sprite.

On that note, there are also a limited number of objects that can be rendered at once: Two of them have fully "customizable" sprites, for the first and second players, and three more have simpler sprites that can only be resized. Each sprite is monocolored and displayed in a grid described by bytes corresponding to the cells of the grid that are to be colored. So, in order to display the dinosaur, we set the bytes that describe the dinosaur, as designed with an external tool, like PlayerPal 2600[9]. Sprites can be alternated between frames in order to make the character animate.

Finally, we would like the player to have a dash mechanic available: with the press of the action button, the dinosaur advances a few extra steps in the direction it is already going, with a cooldown so the move cannot be done in rapid succession. With this, the player can dodge faster when in a tough spot.

3. Designing, Implementing and Tuning Object Behaviors

Having decided on the interaction with the game, how does the game world actually change in order to present a challenge to the player? As mentioned above, the central idea is that the player has to dodge from meteors falling from the sky, so we need to turn that idea into a game mechanic and implement it into the code.

The first thing to make clear is how to represent the meteor in the game. As mentioned earlier, the Atari 2600 has registers and sprites reserved for a second player. As the game is single player only, we can use these resources to represent a non-player object. The names of the registers, for example, lose their meaning here, as we aren't actually using, say, player1x to represent data relevant to some other player, but as the meteor's horizontal position. That is ok, though! Many projects of the time already used this and many other unusual development patterns in order to extract the most from the 2600, that is just how developing for limited systems works: sometimes the developers have to think beyond what the platform was intended to be used for in order to get the most out of it.

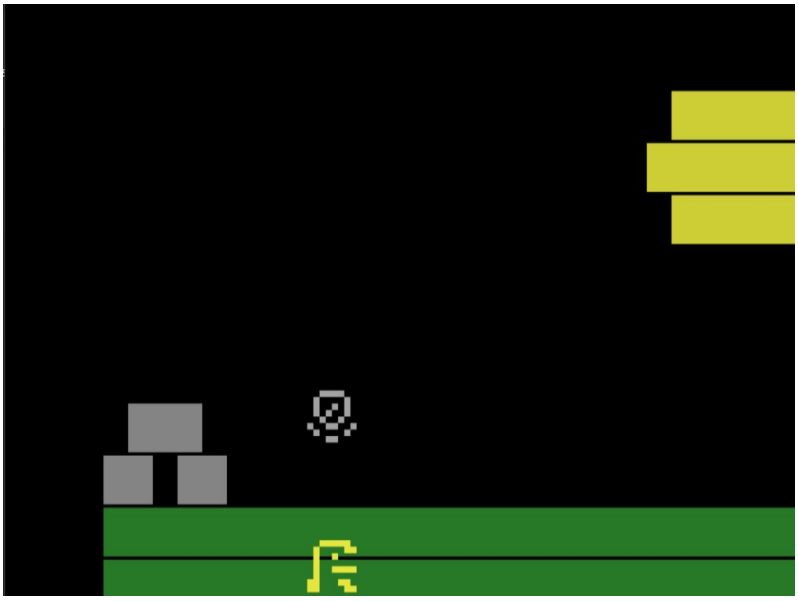

Figure 1. Overview of the Game Screen with the Dinosaur and a Meteor

We also wanted the game to have a shock wave after the meteor's impact, in order to make the rounds more challenging and add a bit more flavor to the game world. We also mentioned there were three more objects available, so that is how two of them were going to be used. Weirder still, these two objects have the names "missile0" and "missile1", intended to be the weapons of each player, and their sprites are simple line segments that cannot be changed in any complex way. While, again, the names do not represent what we use them for in our project, vertical bars moving away from the impact point are enough to represent our meteor's shock waves. Another limitation is that each missile has the same color as its corresponding player so, unfortunately, one of our shock waves has the dinosaur's color and the other has the meteor's, which can be seen in figure 2.

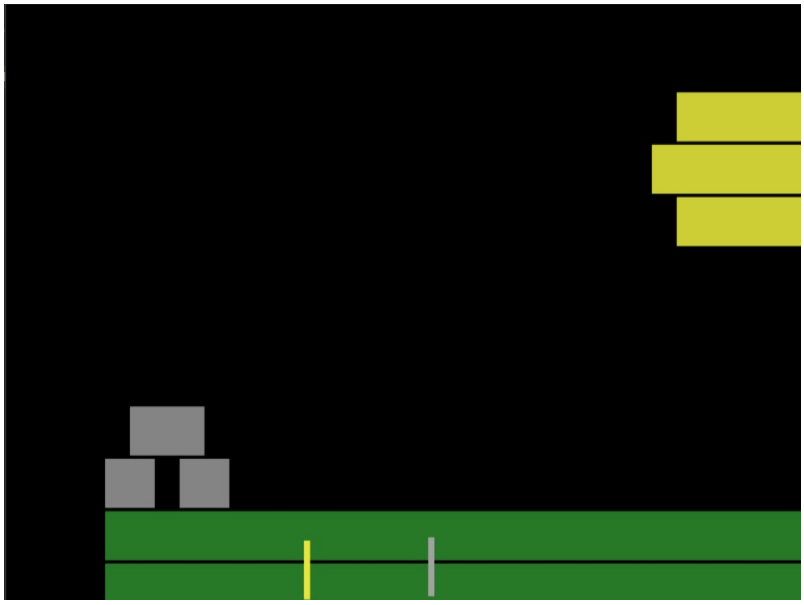

Figure 2. The two shock waves around the impact point

The game loop is simple: a meteor falls from the sky, generating a shock wave on impact that moves to the sides of the impact point, and then another meteor falls and the cycle repeats until the dinosaur is hit, at which instant points are displayed based on the number of survived meteors. One key point is missing though: how to choose the positions where the meteors fall from?

This is where a naive approach, which was the first one our team implemented, falls flat. The simplest way to choose these positions is to just generate a random number between 0 and 160 (the 2600's horizontal resolution). There are two main problems with this approach: First, if a meteor falls too close to the borders of the screen, the sprite loops to the other side (this happens because of the way the 2600 renders). Second, with so many possible positions, there are times when multiple meteors fall very close to each other in sequence. This leads to the player being too still, not really giving the feel of someone trying to avoid a meteor shower.

The second idea was to limit the possible meteor spawn locations to just a few. Each time a meteor was going to spawn, we generated an integer between zero and nine (inclusive), and then spawned the meteor in one of the correspondent locations. Unfortunately, the screen is so small that meteors still kept appearing too close to the ones that came immediately before. We could reduce the number of possible spots to, say, five, and have them be more spaced. This, however, felt very unnatural, as it increased the chances of a meteor falling two or more times in sequence at the same spot.

Looking at these ideas, it starts to seem we would need a complex system for generating these spawn spots, with a mechanism to avoid meteors spawning close to the ones before and still keep the player moving. The solution we came up with, however, was quite simple: Keep the previous idea, reducing the number of possible spots to eight and adding an extra possibility: if the generated number was not between zero and seven, the meteor is instead spawned directly above the dinosaur's position. This guarantees that players cannot be still for too much time: if they are, eventually the meteor will come right at them. As we didn't want this to happen that often, we increased the maximum possible number that the random variable could reach to just thirteen. In short, the probability of a meteor falling in one of the fixed spots is $8/14 \cong 0.57$ and of falling above the player is $6/14 \cong 0.43$.

4. EXPANDING FOR MORE COMPLEX SYSTEMS

The finalized game ended up very hard, with most players that tested it surviving for just a couple of seconds, but supposing we wanted to make it harder, how would we do it? One possible way of doing it is trying to spawn the meteor ahead of where the player is going. Say, for example, that the player is going to the right. Then, we could calculate where the player would be at the moment of impact, assuming they are going the same way and, knowing the speed of the meteor, we would know where to spawn it in order to hit the player.

That would work *if* we assume the player keeps moving the same way, but the player could very well change directions, not walk at all, or even use the dash so the calculations would be off. If the system kept spawning meteors like this, the player would eventually notice and take advantage of it, maybe alternating directions after a while in order to trick the system into spawning ahead but then not going there.

A way to make a smarter decision would be to try to *predict* where the player is going, based on the previous actions, is to divide the play time into "segments" and note which action the player took in each one. If each time segment has a duration of, say, 0.2 second, we note if in the current segment the player stood still (S), walked more to the left (L) or more to the right (R). Then, in the span of one second, we could have a sequence of actions like this: [S, L, L, R, S]. With these sequences, it is possible to apply prediction methods like MLE (*Maximum Likelihood Estimation*) with an N-gram model[10] to the player's actions with more reliable results. Although these methods do not tend to require much memory, the 2600 does not have much to spare, and applying them to the project could prove quite difficult. Verifying if it is even possible is a matter for future possible work.

However, if the design ideas presented here are applied to a project targeted at more modern systems, it might improve the quality of the game's balance, as it gives more tools for the developers, to tune the difficulty throughout a game session. For example, one could remake the game presented here, with the beginning of a round being easier (say, with more time between meteors spawning) and scale up the difficulty as more time passes and the game starts using the player's actions to predict their next move.

5. CONCLUSION

In this work we presented design ideas that can work for projects made for limited systems and be expanded for more complex ones, with a real example of a game made for an old but still very popular system, the Atari 2600.

ACKNOWLEDGMENTS

I would like to thank Augusto Baffa and Guilherme Xavier for organizing Atariando 2.0, which led me to the materials and idea for this article, as well as Bernardo Vieira Santos and María Cruz Cáceres for helping me with the development of the game during the event. Finally, I'd like to thank Bruno Feijó for helping me decide on the theme for the final section and giving general feedback on my ideas.

REFERENCES

[1] Panic — Playdate console page (https://play.date/)

[2] Lexaloffle Games — Pico 8 fantasy console (https://www.lexaloffle.com/pico-8.php)

[3] Nintendo Homebrew Community — 3ds hacks guide (https://3ds.hacks.guide/)

[4] Chris Maltby — GB Studio (https://www.gbstudio.dev/)

[5] ICAD Games— Projeto Atariando (http://atariando.inf.puc-rio.br/ferramentas.php)

[6] Atarius Maximus — Code Snippets & Samples for bB Beginners (https://forums.atariage.com/topic/109288-code-snippets-samples-for-bb-beginners/)

[7] Fred Quimby — Batari BASIC (https://github.com/batari-Basic/batari-Basic)

[8] Duane Alan Hahn — Batari BASIC commands (https://www.randomterrain.com/atari-2600-memories-batari-basic-commands.html)

[9] Kirk Israel — PlayerPal 2600 (https://alienbill.com/2600/playerpalnext.html)

[10] Lei Mao — Maximum Likelihood Estimation of N-Gram Model Parameters (https://leimao.github.io/blog/Maximum-Likelihood-Estimation-Ngram/)