Game Design Document

Júlio César Enezio Júnior

January 22, 2025

Contents

List of Symbols

Symbol	Description
SYSID	System Identification
FT	Transfer Function
NARMAX	Nonlinear Auto-Regressive Model with eXogenous inputs
SFX	Sound Effects
NPC	Non-Playable Character
MEF	Finite Element Method

Introduction

This document compiles all the relevant information for the conception of the game AlexandriA. AlexandriA is a platform game designed as a learning tool for teaching mathematical modeling and system identification. The game's name references the Library of Alexandria, one of the greatest symbols of historical and intellectual preservation in antiquity. Founded around the 3rd century BCE, it aimed to gather all the knowledge of the known world, housing thousands of manuscripts from various cultures. In addition to being a center for studies and intellectual debates, the library represented the ideal of preserving and disseminating knowledge. Its destruction is remembered as one of humanity's greatest cultural losses.

The development will utilize the PyGame-CE module [6] from the Python programming language. This project aims to address the need for theoretical grounding in the subject and contribute to advancements in both mathematical modeling of systems and engineering learning tools.

Below are the main topics to be addressed in the game, presented in the sequence in which they will be implemented in the storyline:

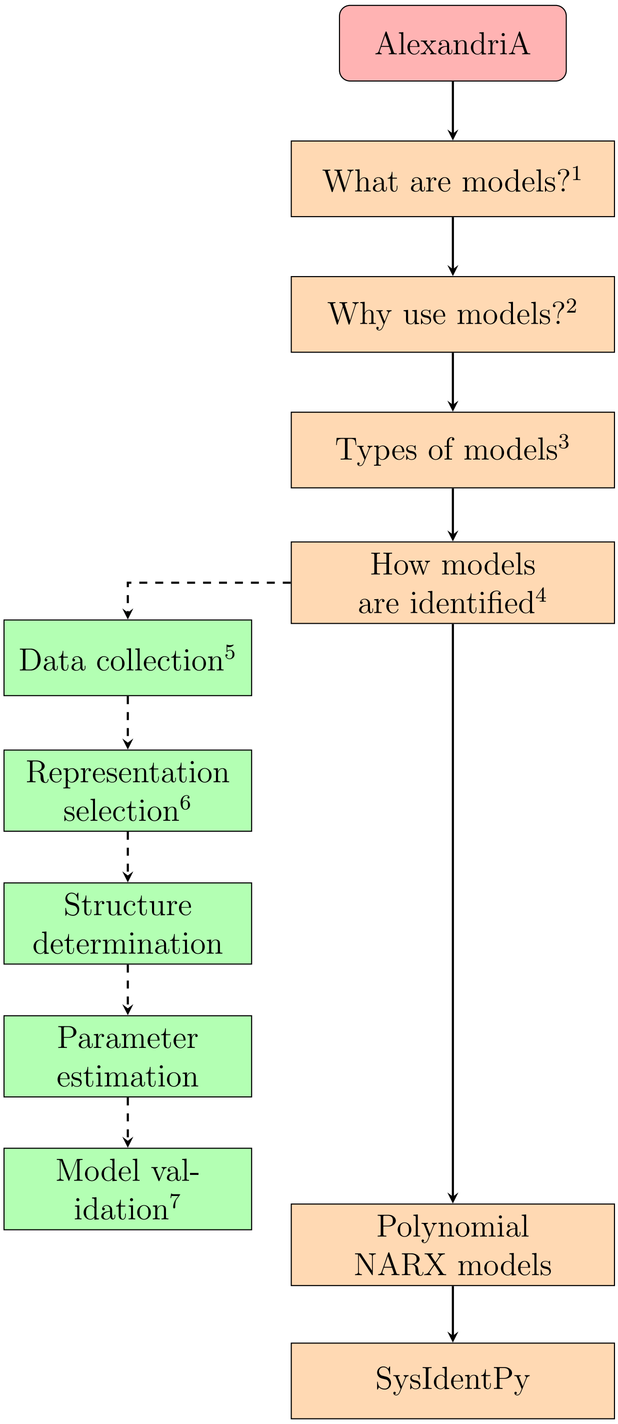

Figure 1.1: Flowchart of the game's storyline.

In the following sections, the methodologies to be applied in order to follow the content sequence of the Flowchart 1 will be described, as well as a description of auxiliary software, characters, storyline sequence, and mechanics.

[1] A deeper focus and elaboration are needed in this section;
[2] Or why we use them;
[3] Parametric or Non-Parametric;
[4] Focus on SYSID and analytical methods;
[5] Focus on instrumentation;
[6] Reference: FT, Sundaresan method, Frequency domain identification, linear models;
[7] This section requires an extremely didactic approach to avoid being tedious.

The references used for the preparation of this document and the game are [3, 9, 13, 1, 15, 2, 14, 7].

Softwares

In addition to the PyGame-CE library, written in Python, other software will be used for the game development. These will be listed below, along with a description of their main functionality.

2.1 BeepBox

BeepBox [11] is an application for creating instrumental music tracks. It is available as open-source, either through an online page or for download. BeepBox will be used in this project to design tracks for the game's levels.

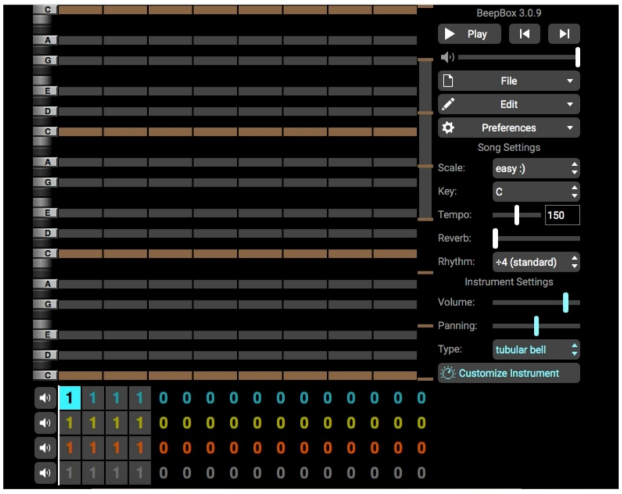

Figure 2.1: BeepBox Interface.

2.2 Px Editor

Px Editor [5] is an open-source software for creating pixel art. With its minimalist design, it allows for intuitive and quick creation of drawings. Px Editor will be used for creating illustrations of the characters, scenes, and objects in the game.

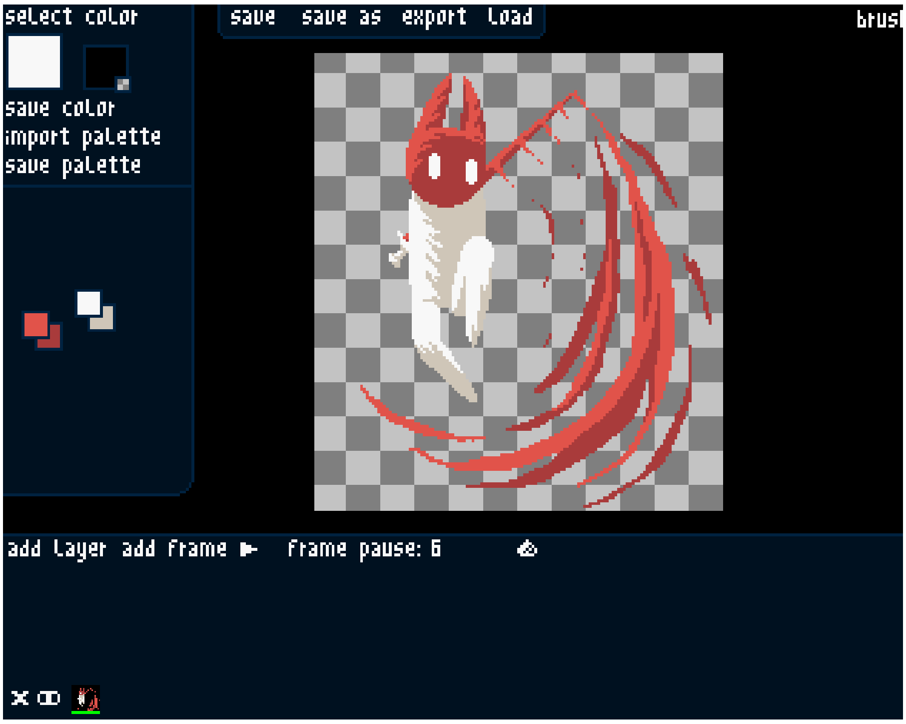

Figure 2.2: Px Editor Interface.

2.3 BFXR

BFXR [8] is an open-source software for creating SFX tracks. It is an enhanced version of the SFXR project [12], and will be used for creating audio tracks for actions such as movement, interaction with the environment, and others.

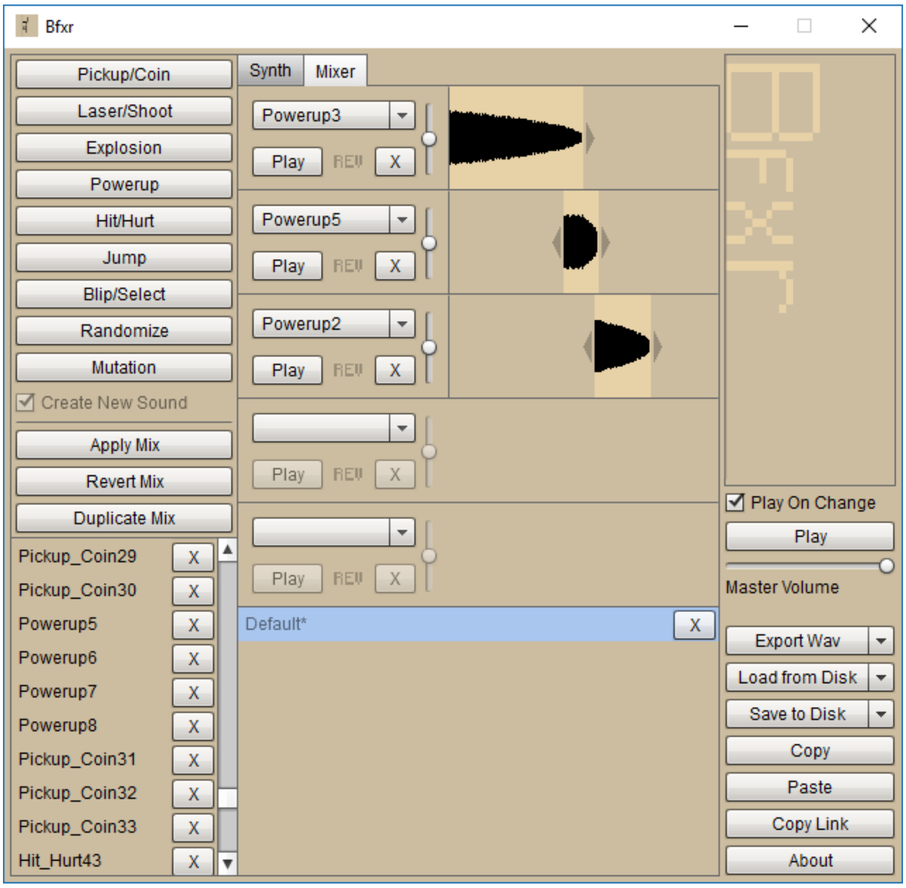

Figure 2.3: BFXR Interface.

2.4 Tiled

Tiled [10] is an open-source software for creating platform game level illustrations. Using this software, the environments where the player can move will be developed.

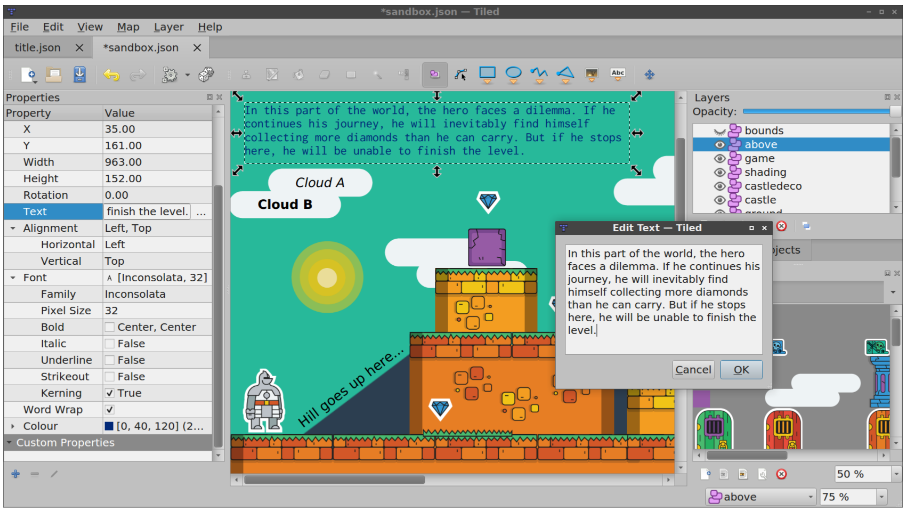

Figure 2.4: Tiled Interface.

2.5 LibreSprite

LibreSprite [4] is an open-source software for creating game sprites (graphic assets). It will be used to create environment objects, character drawings, as well as facilitate the animation of these drawings during the game implementation.

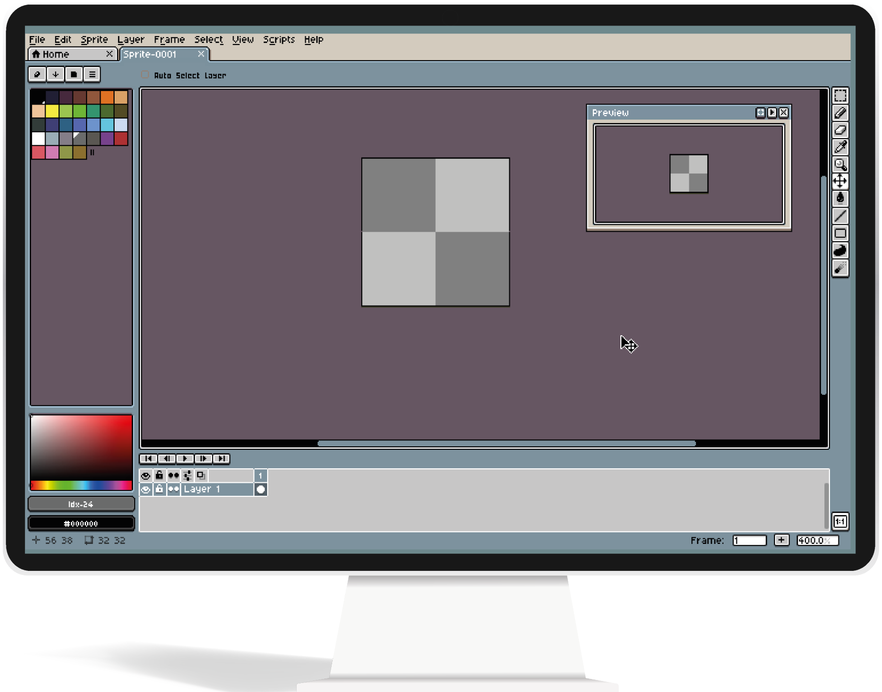

Figure 2.5: LibreSprite Interface.

Mechanics

The game will be developed to follow the platform genre, meaning it will feature a 2D map with simple movement mechanics such as walking side to side and jumping. Additionally, the following mechanics will be implemented:

- Interaction and dialogue with NPCs;
- Interaction with items and objects;
- Inventory for items.

The game will feature quizzes and puzzles, where the player will answer questions or perform some interaction to progress in the game. To create immersion, a life system (or something similar) will be used to introduce consequences for mistakes during gameplay.

Thematic Reference

As previously mentioned, the game has a theme inspired by the Library of Alexandria. Therefore, the sound effects, visual arts and illustrations, NPC names, setting, and puzzles should be based as much as possible on the ancient Egyptian theme and other elements related to the Library of Alexandria.

Characters

5.1 Player

The main player, here referred to as *player* will not have a defined name or gender (throughout this document, the *player* is referred to in the masculine form, but this is only for ease of writing). Options for customizing these characteristics, as well as appearance, will be implemented. The complexity of the appearance customization system will not be defined here and will be entirely dependent on the development of other essential parts of the project.

The character's backstory is as follows:

> A student in the system modeling course is struggling to understand the material. Then, his professor recommends a book in the library that will help him understand the subject but warns that it would be his last chance before failing the course.

5.2 NPCs

Below are the description and backstory of the game's NPCs:

5.2.1 Professor

The professor *Arch* (reference to Archimedes [7]: philosopher and prominent Greek mathematician, who may have studied in Alexandria during his youth) is the instructor for the system modeling course. He is a man in his 50s, with black hair and a thick, graying beard.

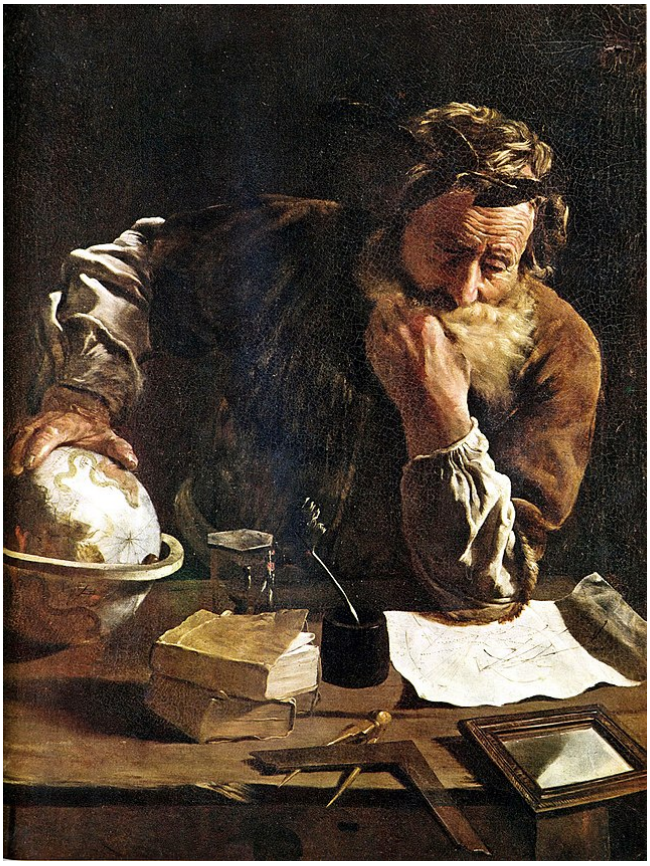

Figure 5.1: Archimedes (Available at: [17]).

The professor's backstory is as follows:

The professor of the system modeling course receives a visit from the *player* who is struggling to understand the material. To help him, the professor tells the *player* to go to the history section of the library and find the book **AlexandriA**. The book transports the reader to an alternate reality, where they must play and complete the levels in order to learn about the content. To use the book, the reader must write their name and the subject they are having difficulty with. Before the student leaves, the teacher warns that this would be their last chance, and if they failed to complete the game, they would be failed.

Notes

This step of writing the name can be included as part of the character creation process.

And a nice touch would be having the name of the teacher and perhaps other personalities included on this page.

5.2.2 Narrator

The narrator is the character who will give instructions to the *player* within the game. His name is **Euclide** and his appearance is similar to reference from the Greek mathematician Euclid [7], who was one of the pioneers of the library:

Figure 5.2: Euclide (Available at: [16]).

It is important to note that all the topics discussed in the storyline are delivered to the *player* by the narrator.

Storyline

The developed storyline must follow the topics shown in the Flowchart 1 and the plotline described below:

6.1 Act 1

The *player* was struggling in the system modeling course: he had performed very poorly on the exam and couldn't grasp the concepts. If the situation did not change, he would fail the course. Determined to change this situation, he goes to the professor's office and pleads for help. The teacher instructs the *player* to look for a certain book in the history section of the library, which he was sure would help: the name of the book is **AlexandriA**. Before the *player* leaves the room, the teacher warns him that if he does not succeed with the book, he will automatically fail the course.

The *player* goes to the library, enters the history section, and looks for the book AlexandriA. Anxious, he opens the book and finds the following note from the author: "Within the pages of this book lies the legacy of the Library of Alexandria: listen carefully to the whispers that have endured through the ages and reveal the infinite truths of the universe to those who dare to understand".

The *player* is intrigued by the sentence, so he turns to the next page and comes across a kind of list with two columns: the name of the person and the name of the subject. The *player* writes his name and the name of the subject "System Modeling", and as soon as he finishes writing, the book begins to flip its pages quickly on its own and glow. Suddenly, the *player* is magically pulled into the book. The *player* finds himself inside the universe of AlexandriA, which reveals itself to be a game designed to help students understand the chosen subject.

Notes

This initial part unfolds as a type of cutscene, where the player interacts only by advancing the dialogue.

6.2 Act 2

Inside the game, the *player* initially learns the mechanics of movement and interaction, as well as how to access the inventory. After the tutorial, the narrator explains the purpose of the game to the *player* and begins the content.

The first part is the definition of models: it is explained that models are a set of hypotheses about the structure or behavior used to represent a physical system. Models can range from mathematical equations describing their real behavior to scale models or mockups of aircraft. The narrator also explains that numerous representations of the same system can exist, even when the same modeling method is applied. The game will focus on showcasing more about mathematical models.

Notes

It is important to include interactive or animated content on the screen during the explanation.

A key point to highlight is the fidelity level of the model (e.g., whether the model of resistance depends on temperature).

Next, the motivations for using models are presented. Some of these motivations include predicting behaviors, ensuring safety when operating the real system, modeling control systems, and more.

Notes

A good example to include is having the *player* interact with an industrial plant, which then explodes, demonstrating the need for studying the system's model for decision-making.

In the third part of this act, the types of models are described. Some examples that can be covered here include circuit simulations, scale models and mockups, FEM simulations, process flow diagrams, cellular models, CAD drawings, and others.

Notes

A substantial number of examples is important, but it is necessary to bring greater focus in this section to mathematical models, also highlighting the types of mathematical models. Additionally, it should be shown that simulations are COMPLETELY dependent on mathematical models.

Including some puzzles/quizzes to encourage the *player*'s interaction with the game is recommended.

6.3 Act 3

The third part of the storyline addresses the process of obtaining these models. Initially, the distinction between parametric and non-parametric models is presented. Then, focusing on parametric models, the *player* is introduced to the types of modeling (black-box, grey-box, or white-box) with interactive examples. Subsequently, the narrator emphasizes that the game focuses on SYSID.

Notes

While examples of the topics covered are necessary, it is crucial to maintain focus on following the basic structure of the Flowchart 1. Additional examples and interactions can be added later.

This act covers the largest number of concepts, but despite this, it should not be overly extended in a way that feels tedious.

After this introduction to the act, the narrator presents the five main steps of SYSID and then explains each one individually:

6.3.1 Data Acquisition

The narrator introduces the concept of data acquisition, highlighting the importance of acquiring data under the best possible conditions. He explains that reliable data is essential for building accurate models and that the quality of acquisition depends on various factors, such as the proper selection of sensors and the acquisition system.

> **Observations**
>
> Focus on instrumentation here, highlighting the need for a reliable acquisition system.
>
> Some interesting interactions to include involve choosing the acquisition frequency based on the expected process and handling signal noise (e.g., illustrating filters).

Quantization issues are also addressed, emphasizing the need for the sensor output format to be compatible and interpretable by the respective computational system.

> **Observations**
>
> At this point, it would be interesting to add a puzzle similar to fix wiring (`https://static.wikia.nocookie.net/among-us-wiki/images/e/ef/Connected_Wiring.png/revision/latest/scale-to-width-down/902?cb=20240329182838`) from the game Among Us, where the *player* would have to connect the processes in the correct order.

6.3.2 Choice of Representation

The narrator explains that with the data in hand, it must be decided how to adjust the data so that it can represent the system, or in this case, choose the appropriate mathematical representation. Some types of representations are presented, such as linear models (straight lines, quadratic functions, or polynomial combinations, for example), neural networks, Fourier Transforms, identification through frequency domain, exponential and logarithmic functions, etc.

> **Observations**
>
> In this part, a possible puzzle could be to present a set of data and have the player choose which representation they think would best fit the dataset.

6.3.3 Determining the Structure

In this part, the narrator shows the need to choose, based on the selected representation, which terms are necessary to form the structure of the mathematical model. The Akaike Information Criterion and its use cases are presented, along with its importance in the fields of statistics and SYSID.

> **Observations**
>
> This part is VERY abstract, so the focus should be on making the learning curve as smooth as possible. Provide examples such as polynomial degrees and the harmonics of the Fourier series.

6.3.4 Obtaining the Parameters

The narrator shows that in addition to determining the structure, it is important to compose the model with parameters (which may or may not be constants) that determine how much each component influences the final value returned by the model.

Observations

An example that could be interesting is the weights in each neuron of a neural network, which, if not properly adjusted, can propagate an error all the way to the last layer.

Other examples, related to daily life, are tuning instruments or cooking recipes (where the parameters are the ingredients). It is important to relate the content to things that are tangible for the player.

6.3.5 Model Validation

The narrator presents the ways to assess the quality of a model. He explains and demonstrates validation metrics, such as squared errors and prediction errors. He emphasizes that the model will rarely be a perfect representation, but as long as the metrics are satisfactory for modeling the system's phenomenon, the model is considered adequate.

6.4 Act 4

In the fourth act, Polynomial NARX models will be discussed: a type of nonlinear modeling widely used. Their characteristics, advantages, and use cases are described. Delving deeper into the topic of NARX models, the Python library **SysIdentPy** is introduced: a toolbox for system identification using NARMAX modeling (a particular case of NARX models). The library's functionalities, the advantages of its use, as well as some use cases, are also presented.

Observations

In this act, it is important to always highlight why computational tools should be used for SYSID.

At the end, if the *player* has completed all the game levels without running out of lives, the Narrator congratulates him and says that he has absorbed all the knowledge the library had to offer at that moment.

6.5 Final Act

If the *player* manages to complete the objectives, he is transported out of the book. After that, he receives approval in the subject.

Notes

This act is just a closing cutscene.

Bibliography

[1] L.A. Aguirre. *Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, Belo Horizonte, 2007.

[2] Marcos Silvano Orita Almeida and Flávio Soares Corrêa da Silva. *A Systematic Review of Game Design Methods and Tools*, pages 17–29. Springer Berlin Heidelberg, Brasil, 2013.

[3] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley and Sons, Inc, Nashville, TN, USA, 2013. Includes bibliographical references and index.

[4] David Capello. Libresprite. Disponível em `https://libresprite.github.io`, 2021. Ferramenta para criação de sprites para jogos. URL: `https://libresprite.github.io/`.

[5] DaFluffyPotato. Px editor. Disponível em `https://dafluffypotato.itch.io/px-editor`, 2020. Ferramenta para criação de *pixel art*. URL: `https://dafluffypotato.itch.io/px-editor`.

[6] Pygame Community Edition Developers. Pygame-ce: Pygame community edition. Disponível em `https://pygame-ce.org`, 2024. Biblioteca para desenvolvimento de jogos e multimídia em Python. URL: `https://pygame-ce.org`.

[7] Kafr Ferreira. A biblioteca de alexandria e a casa da sabedoria: um breve panorama histórico. *Encontro Nacional De Educação Matemática*, 12:1–9, 2016.

[8] increpare. Bfxr. Disponível em `https://www.bfxr.net/`, 2011. Ferramenta para criação de faixas de SFX. URL: `https://www.bfxr.net/`.

[9] Wilson Lacerda, Luan da Andrade, Samuel Oliveira, and Samir Martins. Sysidentpy: A python package for system identification using narmax models. *Journal of Open Source Software*, 5(54):2384, October 2020.

[10] Thorbjørn Lindeijer. Tiled. Disponível em `https://www.mapeditor.org/`, 2008. Ferramenta para criação de níveis de jogos de plataforma. URL: `https://www.mapeditor.org/`.

[11] John Nesky. Beepbox. Disponível em `https://www.beepbox.co/`, 2019. Ferramenta para criação de música instrumental. URL: `https://www.beepbox.co/`.

[12] Tomas Pettersson. Sfxr. Disponível em `https://www.drpetter.se/project_sfxr.html`, 2007. Ferramenta para criação de faixas de SFX. URL: `https://www.drpetter.se/project_sfxr.html`.

[13] Luis Otoni Meireles Ribeiro, Maria Isabel Timm, and Milton Antonio Zaro. Modificações em jogos digitais e seu uso potencial como tecnologia educacional para o ensino de engenharia. *RENOTE*, 4(1), July 2006.

[14] Adam M. Ross, Matthew E. Fitzgerald, and Donna H. Rhodes. Game-based learning for systems engineering concepts. *Procedia Computer Science*, 28:430–440, 2014.

[15] Mario Gonzalez Salazar, Hugo A. Mitre, Cuauhtemoc Lemus Olalde, and Jose Luis Gonzalez Sanchez. Proposal of game design document from software engineering requirements perspective. In *2012 17th International Conference on Computer Games (CGAMES)*. IEEE, July 2012.

[16] Wikipedia contributors. Euclides — Wikipedia, the free encyclopedia, 2024. [Online; accessed 20-January-2025]. URL: `\url{https://pt.wikipedia.org/w/index.php?title=Euclides&oldid=67787842}`.

[17] Wikipedia contributors. Archimedes — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Archimedes&oldid=1270309076`, 2025. [Online; accessed 20-January-2025].