Téssara: Developing of a game to teach about the fourth dimension and 4D objects

Samuel Isaac Camargo Consuegra
Universidad del Norte
Barranquilla, Colombia
sicamargo@uninorte.edu.com

Juan José Bolívar Velandia
Universidad del Norte
Barranquilla, Colombia
jbolivarj@uninorte.edu.com

Juan Diego Rojas De La Hoz
Universidad del Norte
Barranquilla, Colombia
rojasdelahoz@uninorte.edu.com

Margarita Rosa Gamarra Acosta
Universidad del Norte
Barranquilla, Colombia
mrgamarra@uninorte.edu.com

Fatima Castro Castro
Universidad del Norte
Barranquilla, Colombia
fatimac@uninorte.edu.com

Eduardo Angulo Madrid
Universidad del Norte
Barranquilla, Colombia
edangulo@uninorte.edu.com

Juan David Borja Cantillo
Universidad del Norte
Barranquilla, Colombia
jborjad@uninorte.edu.com

.

Abstract — **One of the main difficulties when it comes to understanding the fourth dimension is the lack of concrete examples of it in 3D space; there is an absence of any physical reference to higher dimensions, which makes it harder to conceptualize how 4D objects would behave. However, there exists an useful resource in the form of the cross-sections analogy, which defines each dimension as a set of "cuts" from a lesser dimension, stacked together. Téssara seeks to implement this analogy in the Godot game engine with the objective of allowing people to experiment with this concept in a didactic manner, and generating a deeper understanding of shapes in the fourth dimension. Moreover, the analogy is implemented through the rendering technique named Ray Marching, in which the surface of each shape is defined using a mathematical Signed Distance Function. The results obtained from this project reinforce the relationship between didactic teaching methodologies and learning. As such, the push for more strategies of this type is to be highlighted for the sake of teaching complex topics in the form of games.**

Keywords - 4D objects; 4D; Godot, Didactic; Ray Marching; SDF; hyper-shapes.

I. INTRODUCTION

Games have been a fundamental tool for the purpose of exploring and understanding the world around us. In video-games, those that present a simulation of reality allow the user to experiment with concepts, strategies and possibilities from the real world. Through this, individuals are provided with a path to discover elements that contribute to their formation [1]. Moreover, in mathematics, games can play a crucial part in the development of new theories and the comprehension of complex phenomena. This much is demonstrated by Moya, Rodríguez and Saldías [2], study in which they present the implementation of a game that simulated the properties of geometric shapes for students between the ages of 13-14. Such strategy resulted in a higher percentage of correct answers when the students were asked about the topic.

When proposing a video-game for the teaching of 4D, it is said that the conceptualization of what a dimension even is means a pretty significant challenge. This is due to the fact that it is a concept with diverse use cases often lacking a precise definition [3]. Also, the addition of higher dimensions than those perceived makes it harder to understand. For example, the concept of "dimension" is often associated with a "perspective" from which a phenomenon is observed [3], instead of a geometric description. For the purposes of this paper, we make use of "dimension" applied to geometric spaces, in which each dimension represents an axis of freedom perpendicular to all the other ones.

In this context, Téssara seeks to offer a unique opportunity to explore this concept in an accessible and didactic manner. When playing, the user will be able to manipulate 4D objects, visualize complex shape transformations, and develop an intuitive sense over this concept that would otherwise be harder to attain. Also, the game includes a color blindness correction filter with the purpose of reducing visual difficulties associated with his condition. Téssara possesses great educational potential as it helps in the understanding of a complex geometric topic, and it demonstrates the effectiveness of didactic strategies when teaching such subject matters.

In contrast to the traditional 3D or 2D game development workflow, working in 4D presents unique challenges for the making of the game. This is because traditional game engines are not equipped with architecture that supports the simulation of 4D objects or spaces. As such, complex mathematical concepts are to be applied for the rendering and simulation of such objects.

II. Related Works

Conveniently, there are a few projects that implement techniques that work as a basis for what is being developed:

Firstly, “4D Toys” is a game that allows the manipulation of diverse 4D primitives inside a discrete space, allowing the user to experiment with the behavior and physics of four-dimensional objects [4]. The developer, Marc Ten Bosch, achieves this by implementing the cross-section analogy as described by British author Edwin Abott in his satirical novel “Flatland: A Romance of Many Dimensions” [5]. In this novel and in 4D toys, the world inhabited by the characters is but a lower-dimensional cross-section of a higher-dimensional space. Which is to say, if a 3D sphere were to pass through a 2D plane, beings in the plane would only observe one 2D cross-section of it at a time; they would see the circle- shaped area that intersects with the plane. Clearly, this logic could be extended for higher-dimensional objects. If a 4D sphere were to pass through a 3D space, beings in the space would only observe one 3D cross-section of it at a time; a sphere-shaped region that intersects with the space.

Secondly, Miegakure is a game in development that allows the exploration of and interaction with a 4D world with the objective of puzzle solving [6]. Also developed by Marc Ten Bosch, it makes use of the cross-section analogy to reflect changes in a whole space, instead of only shapes. These works demonstrate the possibility of simulating and illustrating complex topics through computers, and also involving them in narrative and interactive experiences.

Thirdly, 4D Golf is a golfing videogame that uses 4D tracks as a setting [7]. The work elicits a kind of intuitive sense for the comprehension of 4D space allowed only by the medium it uses; being videogames. The player is tasked with transforming the terrain in such a way that allows the ball to reach the end of the circuit, it being a 4D space constructed with 3D tetrahedra. What this means is that while a traditional golfing game would have a 3D circuit constructed from 2D primitives, 4D Golf has 4D circuits constructed from 3D primitives. Effectively, this game's rendering pipeline is the same as any other game's, elevated a dimension higher. This accurately represents a completely 4D space, and the task presented to the players of playing golf is more than enough to make them grow accustomed to it. As such, this is an important precedent for games as a learning tool of complex topics, and also an impressive technological feat for the topic of n-dimensional space representation.

Lastly, 4D Miner is a survival and construction game that takes place in a four-dimensional world [8]. It allows players to explore an infinitely generated map that exists along four axes. Particularly, the player has the option of rotating the 3D cross-section they inhabit with the scroll wheel. Also making use of the cross-section analogy, players come to understand the fourth dimension through the particular challenge of building safe enough houses and battling 4D enemies. As such, it is also a precedent of games as a learning tool, and is particularly illustrative of the flexibility of this medium.

Additionally, Marc Ten Bosch has published a generalization for “N-Dimensional Rigid Body Dynamics” [9]. This work is focused on the physical interactions between hyper-shapes. While being an important precedent for the development of more games focused on 4D space, it is out of scope for this project. As such, its generalizations are not implemented in the developed game.

III. Methodology

The technology employed in the development of the game is one that allows the rendering of 4D shapes in a way that matches what is described in the cross-sections analogy. In this case, a rendering method called “Sphere Tracing” is employed.

As described by John Hart [10], Sphere Tracing allows the rendering of shapes through what are called “Signed Distance Functions (SDFs)”. These functions output the distance between a point in space and the surface of the shape it describes. Each primitive shape has its very own SDF. SDFs return values that approach 0 the closer the point is to the surface. Moreover, the only input data they require is a point in space, and the dimensions of the shape (in most, primitive cases). If it is established that a point in space is represented by the coordinates it is located in, however dimensions there are in that space corresponds to the number of coordinates assigned to the point. What this results in is the possibility of tasking computers with rendering n-dimensional shapes through the description of simple math functions.

This possibility is brought to reality through the algorithm of Sphere Tracing. What it consists of is defining a point from which Rays are shot from, where each Ray corresponds to a pixel on the computer screen (in other words, locating a camera). Then, these rays “march”; they evaluate the SDF of the world in a point, then advance the distance returned by it in the same direction. This process is repeated until the distance returned by the SDF is small enough to consider that the Ray could “hit” the surface of a shape, or when it marches too many steps or a maximum distance is traveled.

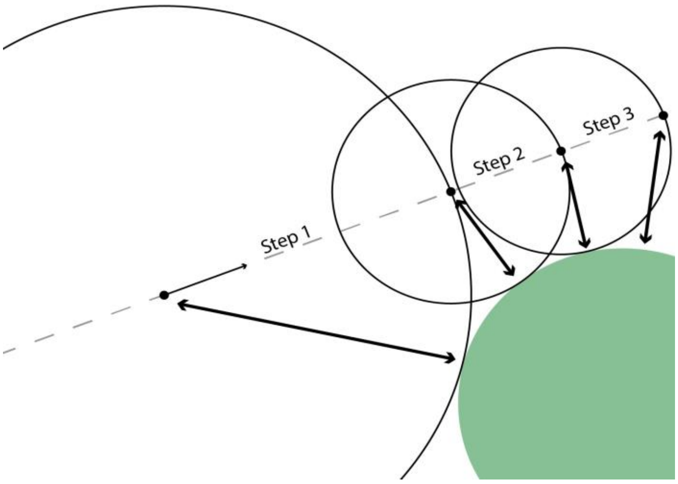

Figure 1. Illustrative diagram of Sphere Tracing. Each “step” is a point from which the SDF is evaluated, after which the Ray advances to the next one Adapted from: [11].

Consequently, as this algorithm defines surfaces through math functions, the amount of dimensions these surfaces have corresponds to the amount of variables the functions consider. In other words, if what is required is to visualize a four-dimensional cube, what needs to be done is to define an SDF with four variables.

The requirements for the program are divided in two categories: functional and non-functional requirements. The execution of these objectives represents the making of the final product.

On one hand, the functional requirements are as follows:

- **4D shape rendering:** The game correctly displays 4D shapes in the user's screen.
- **4D shape transformations:** The game implements a way for the user to interact with the shapes.
- **Puzzle implementation:** The game implements levels that test the user in a way that requires them to reason spatially.
- **Accessibility feature:** The game implements a color correction filter that makes it easier to use for color-blind people.

Additionally, non-functional objectives are defined under the model to classify software quality attributes: Functionality, Usability, Reliability, Performance, and Supportability (FURPS) [12].

- **Functionality:** The program's functionalities are facilitated by implementing data structures that store the information of tetra-dimensional figures in memory. Specifically, linked lists are used to group them. On the other hand, the player's collision handling is stored in a stack that recursively resolves its resulting velocity after collision. Added to the organization of scenes in tree data structures, the program is guaranteed to work.
- **Usability:** The game offers an intuitive and user-friendly interface, allowing users to quickly familiarize themselves with its mechanics and functionalities. In addition, to optimize the user experience, the game introduces the different functionalities gradually. This strategy allows the user to adapt and understand the content at an appropriate pace, thus avoiding an information overload that could be overwhelming.
- **Reliability:** The game demonstrates great reliability, a wide control of the possibilities that the player's freedom allows. In the same way, the program has sufficient exception control for the possible situations the user generates.
- **Performance:** The program requires a high level of computational power to render the figures. This limits its performance on low-end devices, or devices lacking a dedicated graphics card. However, if you have a dedicated graphics card, the game's performance is adequate for its use.
- **Supportability:** Téssara features an introductory tutorial that guides the user step-by-step through the basic game mechanics, as well as comprehensive documentation detailing its features and functionality.

As mentioned above, the Godot graphics engine was selected to develop the software. This engine uses GDScript, a programming language with syntax similar to Python, created specifically for Godot. The choice of Godot is based on its node-based workflow, which greatly simplifies the creation of the game's logic and interactions. Additionally, Godot stands out for its versatility, suitable for 2D and 3D game development. On the 3D side, Godot offers a powerful 3D scene editor, support for various 3D mesh formats, and advanced rendering capabilities.

Ultimately, the agile SCRUM methodology [13] was chosen for the development of the game. The flexibility and adaptability inherent in SCRUM allowed the team to respond effectively to changes and new ideas that arose during the development process. The successful implementation of periodic progress updates facilitated early feedback, allowing the game to adapt to new user needs and preferences.

A. *Class Diagram*

In figure 2 the UML class diagram of the game is shown. In it, the general structure of a Téssara scene is illustrated.

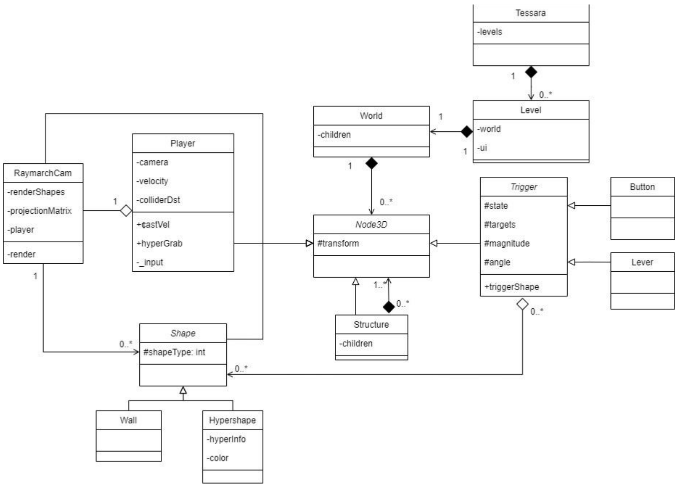

Figure 2. UML Class Diagram.

The classes represented in this diagram follow the Composite design pattern. This allows the treating of data as if it belonged to leaves and branches in a typical Godot tree scene. In this case, each Structure can contain another generic Structure, or a concretely defined class that extends Node3D (Player, Hypershape, RaymarchCam, etc).

The main class represents the game (Téssara), which contains each level. Each level then stores a World (physical interaction) alongside the UI elements (direct interaction) available to the user. World then branches out into Node3Ds. The program is arranged in such a way that Structures work as "folders" for clusters of Shapes or Triggers, whilst the Player is a leaf by itself.

The Shapes present in a World are one of two types. Walls are regular 3D shapes that are to be rendered with Godot's in-house graphics, whilst Hypershapes are 4D Shapes that have their rendering handled by the raymarcher.

IV. RESULTS

Development results are as follows:

Figure 3. Scene rendered through raymarching.

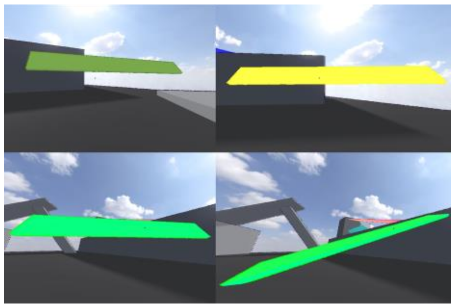

Figure 4. Hypershape transformation performed by the player.

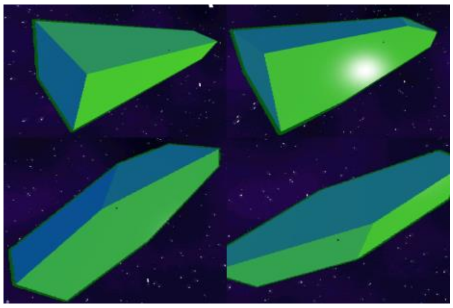

Figure 5. Hypershape rotation performed by the player.

The usability poll results are as follows. The poll was performed on a group of 17 people, aged 17-21. In each question 0 is "Not at all" and 5 is "Quite a lot":

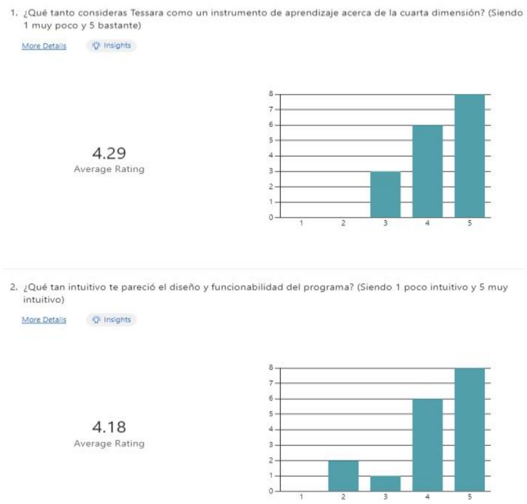

Figure 6. Questions 1 and 2. "How effective a teaching instrument for the fourth dimension do you consider Téssara to be?" and "How intuitive and functional do you consider the program to be?", respectively.

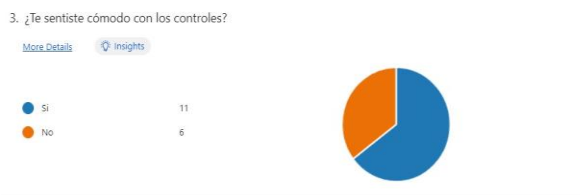

Figure 7. Question 3. "Were you comfortable with the controls?"

The results obtained here help identify and solve usability or accessibility problems, with the objective of bettering the learning experience of the user.

Figure 6 illustrates how valued Téssara is as a teaching instrument (4.28/5), and how intuitive and functional it is deemed to be (4.18/5). Lastly, figure 7 shows that most people felt comfortable with the controls. This indicates its effectiveness as a teaching tool that is easy to use and understand.

V. Conclusions

According to the obtained results, Téssara has been demonstrated to be a useful tool to familiarize the users with the concept of the fourth dimension. Evidence suggests that turning a challenging or complex topic to a dynamic game has favorable effects when it comes to teaching about that topic.

Téssara satisfies all functional requirements proposed in this paper: it renders hyper-shapes with accuracy, allows users to interact with them, possesses puzzles to challenge the user, and promotes learning of the topic through them. Also, it employs a color correction filter to be used by color-blind people.

That being said, the development of this game possessed a plethora of challenges. The computational cost of ray-marching cannot be understated. It requires at least some form of dedicated GPU to be playable. Also, there are accuracy problems when it comes to colliding the user with the hyper-shapes, which does get in the way of the user experience.

This leaves open the possibility of further exploration of the topic at hand. Not only in the form of possible optimizations for the rendering and simulation of 4D shapes, but also, in a more general sense, the making of more games with the purpose of education. When done right, games are a powerful tool to reach multitudes of people and teach them about any topic.

Referências Bibliográfica

[1] Núñez-Barriopedro, E.; Sanz-Gómez, Y.; Ravina-Ripoll, R. Los videojuegos en la educación: Beneficios y perjuicios. Revista Electrónica Educare 2020, 24, 240–257. Publisher: http://creativecommons.org/licenses/by-nc-nd/3.0/, https://doi.org/10.15359/ree.24-2.12.

[2] Gonzalez, O.; Rodríguez, E.R.; Saldías, P.V. Implicaciones de la gamificación en educación matemática, un estudio exploratorio. Revista de Educación a Distancia (RED) 2021, 21. Number: 68, https://doi.org/10.6018/red.485331.

[3] Páez, J.; Orjuela, C.; Rojas, C. El concepto de dimensión: errores y dificultades. Funes 2008.

[4] Bosch, M.T. 4D Toys. An interactive toy for 4D children. Available online: https://4dtoys.com/.

[5] Abbott, E.A. Flatland : a romance of many dimensions; London : Seeley and Co., 1884.

[6] Bosch, M.T. Miegakure [Hide and Reveal]: A 4D puzzle-platforming game. Available online: https://miegakure.com/.

[7] CodeParade. How Do You Build 4D Worlds? - 4D Golf Devlog #3. Available online: https://www.youtube.com/watch?v=dbq9uX_MycY, 2022.

[8] Mashpoe. 4D Miner: A 4-Dimensional Survival Sandbox Game. Available online: https:// mashpoe.com/4d-miner.

[9] Bosch, M.T. N-dimensional rigid body dynamics. ACM Transactions on Graphics 2020, 39, 55:55:1–55:55:6. https://doi.org/10.1145/3386569.3392483. 227

[10] Hart, J. Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces. The Visual Computer 1995, 12. https://doi.org/10.1007/s003710050084.

[11] Cole, J. Signed distance fields. Available online: https://jasmcole.com/2019/10/03/signed-distance-fields/, 2019.

[12] Capturing Architectural Requirements. Available online: https://web.archive.org/web/20201112020231/http://www.ibm.com/developerworks/rational/library/4706.html#N100A7, 2020.

[13] About Scrum.org | Scrum.org. Available online: https://www.scrum.org/about.