Enhancing learning in Primary Education with an Interactive Aquarium

Tomás Pinto dos Santos

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisor: Manuel Fernando Cabido Peres Lopes

Examination Committee
Chairperson: Maria Luísa Torres Ribeiro Marques da Silva Coheur
Supervisor: Manuel Fernando Cabido Peres Lopes
Member of the Committee: Rui Filipe Fernandes Prada

October 2020
Acknowledgments

Firstly, I would like to thank my mother for providing me with everything I needed throughout life, allowing me to further my education up until this point. Thank you for your endless patience, trust, and care.

To my friends: thank you for all the emotional support, for motivating me whenever I lazed about, and for distracting me in times I needed a break.

To my colleagues: thank you for making these last years a blast, for all the things you’ve taught me, and for all the group projects where you pulled your own weight.

Also a huge thank you to everyone at this university, it was a pleasure to be a part of this organisation, where everything is held to such high standards.

Last but not least, a thank you to my supervisor, Prof. Manuel, who always gave me solid advice, not just for this project, but also my future, for his constant availability and feedback, and for sticking with me despite all of my delays and periods of silence.
Resumo

Arte digital é uma prática em constante evolução, cujos últimos desenvolvimentos abriram a porta para novas formas de expressar ideias, como instalações de arte em realidade virtual e arte gerada por computadores. A arte encontra-se agora o mais interativa e acessível de sempre, graças à internet e recentes avanços tecnológicos. Contudo, apesar do seu vasto potencial, as suas contribuições na área da educação não têm sido suficientes. As artes têm sido uma parte essencial da educação primária e secundária de alunos há anos, pois são capazes de desenvolver várias capacidades, promover criatividade e serem integradas com outras disciplinas. Hoje em dia, com um número tão elevado de computadores, tablets e ligações à internet em escolas, deveria haver mais incentivo para inovar a educação com o uso de ferramentas de arte digital. O objetivo deste projeto é demonstrar os benefícios da implementação de arte digital na sala de aula, criando uma instalação de arte interativa que serve como recurso educativo. A solução proposta é um aquário online que corre num site, onde os estudantes podem submeter os seus desenhos de peixes. Aos desenhos também são dados determinados comportamentos sociais e de alimentação. Assim que submetidos, os desenhos habitam o aquário e tornam-se animados, interagindo com o ambiente à sua volta e outros desenhos de acordo com os comportamentos escolhidos. O resultado é uma experiência artística em que os alunos podem participar ativamente e ao mesmo tempo aprender.

Palavras-chave: Arte Digital, Arte Interativa, Segmentação de Imagem, Inteligência Artificial, Ensino
Abstract

Digital art is an ever-evolving practice, and its latest developments have opened the door for new and alternative ways of expressing ideas, such as virtual reality art installations and computer-generated art. Art is now more interactive and accessible than ever, thanks to the internet and recent technological breakthroughs. Despite its vast potential, however, digital art's contributions to the field of education haven't been enough. Arts have been a vital part of primary and secondary education for years, as they can enhance multiple skills, promote creativity and be integrated with other subjects. Nowadays, with such a high number of computers, tablets and internet accesses at schools, there should be more of an incentive to innovate education with the use of digital art tools. The goal of this project is to showcase the benefits of implementing digital art in the classroom, by creating an interactive art installation that serves as an educational resource. The proposed solution is an online aquarium that runs on a website, to which students can submit their drawings of fish. These drawings can be given certain feeding and social behaviours. Once submitted, the drawings inhabit the aquarium and become animated, interacting with their environment and other submitted sketches according to those behaviours. The result is an artistic experience that students can actively participate in and learn from.

Keywords: Digital Art, Interactive Art, Image Segmentation, Artificial Intelligence, Education
Contents

Acknowledgments ................................................................. iii
Resumo ............................................................................. v
Abstract ........................................................................... vii
List of Tables ....................................................................... xi
List of Figures ....................................................................... xiii
Nomenclature ...................................................................... xv
Glossary ............................................................................... 1
1 Introduction ........................................................................ 1
  1.1 Motivation ................................................................. 2
  1.2 Topic Overview ........................................................ 2
  1.3 Objectives ................................................................... 3
  1.4 Thesis Outline ........................................................... 3
2 Related Work ..................................................................... 5
  2.1 Interactive Art Installations ........................................ 5
    2.1.1 Sketch Aquarium .............................................. 5
    2.1.2 Telegarden ....................................................... 8
  2.2 Education ..................................................................... 8
    2.2.1 Arts Education ............................................... 8
    2.2.2 Digital Education ........................................... 9
    2.2.3 Digital Art in Education .................................. 10
    2.2.4 Environment Study ......................................... 11
  2.3 Animation .................................................................... 12
  2.4 Intelligent behaviours ............................................... 14
3 Implementation .................................................................... 17
  3.1 Solution overview ...................................................... 17
  3.2 Image Segmentation .................................................. 21
  3.3 Imgur API ................................................................... 22
  3.4 Firebase API ............................................................. 22
  3.5 Interacting with Unity WebGL .................................... 22
List of Tables

4.1 Fish type distribution. ................................................. 34
4.2 Feeding behaviour distribution. ................................... 34
4.3 Social behaviour distribution. ..................................... 35
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>teamLab’s Sketch Aquarium.</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Painting a Sketch Aquarium drawing.</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Scanning a Sketch Aquarium drawing.</td>
<td>6</td>
</tr>
<tr>
<td>2.4</td>
<td>Touching a fish.</td>
<td>7</td>
</tr>
<tr>
<td>2.5</td>
<td>Touching the virtual food bag.</td>
<td>7</td>
</tr>
<tr>
<td>2.6</td>
<td>Telegarden’s online-controlled robotic arm.</td>
<td>8</td>
</tr>
<tr>
<td>2.7</td>
<td>Number of portuguese schools with interactive white boards [10].</td>
<td>9</td>
</tr>
<tr>
<td>2.8</td>
<td>Average number of students per computer [10].</td>
<td>9</td>
</tr>
<tr>
<td>2.9</td>
<td>“Zoo Escape” - A 2D Interactive game from Leya’s “A Grande Aventura” [11].</td>
<td>10</td>
</tr>
<tr>
<td>2.10</td>
<td>A digital schoolbook’s page on how animals eat [15].</td>
<td>12</td>
</tr>
<tr>
<td>2.11</td>
<td>A schoolbook’s page about animal lifestyles and characteristics [16].</td>
<td>12</td>
</tr>
<tr>
<td>2.12</td>
<td>X. Tu and D. Terzopoulos’s spring-mass dynamic fish model.</td>
<td>13</td>
</tr>
<tr>
<td>2.13</td>
<td>In-game footage of ABZŮ.</td>
<td>13</td>
</tr>
<tr>
<td>2.14</td>
<td>Vertex animation in ABZŮ, as shown at a GDC Talk [19].</td>
<td>14</td>
</tr>
<tr>
<td>2.15</td>
<td>ABZŮ blend shapes for biting motions.</td>
<td>14</td>
</tr>
<tr>
<td>2.16</td>
<td>Separation [21]</td>
<td>15</td>
</tr>
<tr>
<td>2.17</td>
<td>Alignment [21]</td>
<td>15</td>
</tr>
<tr>
<td>2.18</td>
<td>Cohesion [21]</td>
<td>15</td>
</tr>
<tr>
<td>2.19</td>
<td>X. Tu and D. Terzopoulos’ generic intention generator.</td>
<td>16</td>
</tr>
<tr>
<td>2.20</td>
<td>X. Tu and D. Terzopoulos’ schooling behaviour routine.</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Website’s default page.</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Flowchart of a drawing submission.</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>Entering the aquarium.</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>Raster Images.</td>
<td>21</td>
</tr>
<tr>
<td>3.5</td>
<td>Vector Images.</td>
<td>21</td>
</tr>
<tr>
<td>3.6</td>
<td>Image before background removal.</td>
<td>21</td>
</tr>
<tr>
<td>3.7</td>
<td>Converted image with image segmentation script.</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>Forward direction before axis correction.</td>
<td>23</td>
</tr>
<tr>
<td>3.9</td>
<td>Forward direction after axis correction.</td>
<td>23</td>
</tr>
<tr>
<td>3.10</td>
<td>The aquarium’s bounds.</td>
<td>24</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>.NET</td>
<td>.NET Framework</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>Computer Graphics</td>
<td></td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
<td></td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
<td></td>
</tr>
<tr>
<td>PUN</td>
<td>Photon Unity Networking</td>
<td></td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call</td>
<td></td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>Secure Sockets Layer/Transport Layer Security</td>
<td></td>
</tr>
<tr>
<td>STEM</td>
<td>Science, Technology, Engineering, and Mathematics</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
<td></td>
</tr>
<tr>
<td>WebGL</td>
<td>Web Graphics Library</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Arts have been part of primary and secondary education for decades now, and despite being cut at times (in terms of budget and time allocation), in favour of more STEM-oriented curricula, research has shown that children exposed to arts programs do better academically, and develop better social, emotional and cognitive skills than those who aren’t [1] [2]. It is therefore imperative to maintain and find new ways of integrating arts in our school programs.

Arts have, however, undergone fundamental changes ever since computer art broke into the scene. At the time, computer art creators were mostly scientists and engineers, as computers were still hard to find in places other than universities and research laboratories. Computer art’s reception wasn’t entirely positive, especially within the artistic community, who resisted and criticised its use for years to come [3], but it paved the way for what is now known as digital art - an art piece or process, that involves digital technology in its creation or presentation.

Artists now use graphics tablets for drawing, digital brushes for painting, and artificial instruments in music production. They’re no longer limited by traditional tools or materials, and their works are more accessible than ever thanks to the internet. Movies, video games, and television shows can be entirely animated with computer-generated imagery, and new technologies and software are created every day with digital art in mind.

Interactive art installations (art projects in which the spectators actively influence the outcome) are some of the most innovative pieces to come out of this evolution, as they broke away from the norm of being site-specific, and now utilise a multitude of platforms such as the web, mobile devices and even virtual reality to create highly immersive experiences.

With this rising number of new tools, mediums, artists and ideas, it would be a waste not to cross them over to fields that have integrated art in the past, like education. Especially when considering issues like educational stagnation and the need for distance learning. But how has education adapted to technology?

In this digital era, classrooms, much like art, have seen significant benefits from technological advancements. Digital schoolbooks, computers, tablets, and smart boards are available at high numbers [4], helping teachers create classes that are engaging and easy to comprehend, while also developing
Several disruptive innovations for education models have also started to gather support, making use of:

- Massive Online Open Courses (MOOCs) - to provide free online courses, materials, discussions and more, to any student;

- Artificial Intelligence - to automate tasks and provide personalised tutoring without human supervision (Intelligent Tutoring Systems);

- Gamification - to motivate students to study using game elements such as points, badges, leaderboards and more.

Unfortunately, despite all of this, education has been slow to adopt many of these ideas, with common barriers being the costs involved, and the need for training of staff and teachers. Due to this, digital art hasn’t made many contributions to the field, so its influence on student learning remains a question mark. As such, research and experiments regarding this crossover should be done in the future, along with more digital art resources for teachers and students to use, just like this project’s interactive aquarium.

1.1 Motivation

As technology becomes evermore prevalent inside classrooms, its integration in school subjects should become a priority, in order to increase engagement and promote students’ learning experiences.

Arts, in particular, can be combined with other subjects to break down difficult concepts into visuals, which are easier to retain and evoke strong emotional responses on students. Art is also known to improve motor, social and cognitive skills, academic results, and student creativity.

While some efforts have been made to integrate technology in other subjects, digital art is still very underused, despite its vast potential. Not only does it provide new and engaging teaching tools, but it can also help students who struggle with conventional teaching methods, teach them valuable skills that can lead to future jobs, and facilitate learning.

1.2 Topic Overview

This thesis will study if digital art can be used to teach primary school students and the benefits of its integration, using an interactive art installation (a web aquarium with user-generated fish) with a follow-up questionnaire.
1.3 Objectives

This project set out to achieve the following goals:

- Creating an interactive digital art installation;
- Showcasing digital art’s benefits when applied to education (promoting visual learning, creativity, multiple skills...);

1.4 Thesis Outline

Chapter 2 covers work that's related to this project: relevant interactive art installations, the current state of education, especially in regards to digital technology and art, and techniques often used for animating and programming digital fish.

Chapter 3 describes the chosen approach for meeting the project's objectives - the creation of an educational interactive art installation - and goes into detail about the way it was implemented.

Chapter 4 presents the conditions in which the tests took place, and an analysis of the results obtained from the experience.

Chapter 5 concludes the thesis with a mention of its achievements and possible future work.
Chapter 2

Related Work

This chapter covers existing work that is related to the project. It starts off with the interactive installation that inspired it, along with other interactive installations of interest. Afterwards, the work that has been done in the field of education - common digital teaching methods, and what is taught about sea life at primary schools. Lastly, it studies how physical images can be turned to digital ones, and ways of animating and programming them, once converted.

2.1 Interactive Art Installations

2.1.1 Sketch Aquarium

This project was heavily inspired by an already existing art installation from teamLab [5], entitled Sketch Aquarium [6].

teamLab is a Tokyo-based digital artist collective, comprised of professionals from multiple fields - programmers, artists, CG animators, mathematicians, and architects. Their works have been exhibited at multiple venues worldwide, and consist of interactive art installations that promote creativity, collaboration and new experiences.

Figure 2.1: teamLab’s Sketch Aquarium.
In teamLab’s Sketch Aquarium, participants are invited to colour drawings of sea creatures, which include animals such as turtles, squids, jellyfish, sharks, seahorses and more. The painting process is done with crayons:

![Figure 2.2: Painting a Sketch Aquarium drawing.](image)

Once coloured, the sea creatures are then scanned and added to the aquarium, where they swim with others:

![Figure 2.3: Scanning a Sketch Aquarium drawing.](image)

After being added, the creatures can be interacted with in two different ways. Touching them directly makes them swim away, and touching a virtual food bag feeds them.
By participating in this experience, visitors are able to nurture creativity, respect for diversity, interest in technology and self-confidence.

As it stands, Sketch Aquarium is a very engaging interactive installation, but it has some limitations, and its concept could be taken further. Participants are restricted by the selection of drawn creatures available, and there’s also a limited number of scanners, tables, and painting materials on-site. The movement and behaviour of the creatures is also very surface-level, and they don’t interact much with the environment.

Considering the skills it nurtures, its collaborative nature, and how engaging the installation is for children, this concept could be applied to a school environment.

To study if interactive art installations like this one can be used to teach, this thesis proposes a similar concept, with the following changes:

- Participants will be able to submit their own drawings, instead of using pre-made ones;
- Submissions will be done online, and users will be able to customise their creatures’ behaviours;
- The submitted creatures will interact with their environment and other drawings, displaying real feeding and social behaviours;
- The installation will display environment study concepts that would otherwise be learned in class.
2.1.2 Telegarden

Within the scope of art installations, a sub-genre of web-based ones exists, sometimes referred to as internet art.

Telegarden was a pioneering art installation of this type: it allowed web users to view and interact with a remote garden filled with living plants. Members could plant, water, and monitor the progress of seedlings through the movements of an industrial robot arm [7]:

![Figure 2.6: Telegarden's online-controlled robotic arm.](image)

Telegarden managed to garner a large user base (over 10000 users, and more than 100000 visitors) during its lifetime, which ended up creating a small, yet very attached, community.

This art piece can be likened to the project that will be implemented, since they’re both web-based, and have their participants observe “living” creatures grow. Telegarden also combined old technology (agriculture) with new (the Internet), just like this project’s doing with handmade drawings and digital art.

The comparisons stop there, as the projects have very different goals: Telegarden required responsible and frequent visits from its users, as they were dealing with actual living beings. Its goal was not to teach or allow artistic expression, but to create a community who worked towards a goal: making the garden survive.

2.2 Education

2.2.1 Arts Education

Arts education has been a topic of discussion for years now, as its importance in national curricula is often overlooked (in favour of subjects with standardised tests), and its teaching approaches often vary: some think arts should be taught for recreational purposes only, while others argue that they can enhance learning in other subjects.

Despite this, studies have shown that arts education has multiple benefits for students. A good example is a recent study [1] by professors Daniel H. Bowen and Brian Kisida, which investigated the causal effects of arts education experiences. Using data collected by the Houston Education Research Consortium, which covered over 10000 students and 40 schools, the study concluded that more exposure to arts-learning experiences resulted in better student behaviour, compassion and writing ability.
The European Union also believe in arts education, shown by their funding of e-ARTinED [8], a project that created didactic tools to enable teachers to use multiple forms of art to teach curricular subjects.

To avoid an arts education decline, which has been evident in places like the USA [9], now, more than ever, empirical evidence should be accrued showing why arts education is still crucial to student development. More didactic tools, such as the ones created for e-ARTinED, should also be supplied to teachers, so that they can easily integrate them into their classes.

### 2.2.2 Digital Education

Access to technology in the classroom, in the form of computers, interactive white boards and mobile devices has increased to very respectable numbers in recent years. In Portugal, where this project took place, reports show that over 50% of schools have interactive white boards in their classrooms, and the average number of students per computer with an internet connection is approximately 5:

<table>
<thead>
<tr>
<th>T2.4 – Escolas com quadros interativos</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Valores absolutos e distribuição percentual, Portugal continental)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>Escolas com quadros interativos</td>
</tr>
<tr>
<td>Escolas sem quadros interativos</td>
</tr>
</tbody>
</table>

Figure 2.7: Number of portuguese schools with interactive white boards [10].

<p>| G1 – Evolução do número médio de alunos por computador e do número médio de alunos por computador com ligação à internet, por natureza do estabelecimento de ensino |
| (Ensino básico e secundário, Portugal continental) |</p>
<table>
<thead>
<tr>
<th>2013/14</th>
<th>2014/15</th>
<th>2015/16</th>
<th>2016/17</th>
<th>2017/18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluno/Computador</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
</tr>
<tr>
<td>Aluno/Computador - Ensino Público</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
</tr>
<tr>
<td>Aluno/Computador com ligação à Internet</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
</tr>
<tr>
<td>Aluno/Computador com ligação à Internet - Ensino Público</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Figure 2.8: Average number of students per computer [10].

This level of availability opens the door for new teaching methods that involve digital learning resources. Teachers have started to employ multimedia presentations in their classes, using animations, slide presentations, games, and more to turn their classes more interactive, and easier to understand.

As for students, those with digital textbooks have access to similar resources online, in case they want to complement their classes. Below is one of those resources:
Figure 2.9: “Zoo Escape” - A 2D Interactive game from Leya's “A Grande Aventura” [11].

In this game, children are asked to catch animals that fit a description provided through audio cues. In this case, the goal was to catch animals categorised as reptiles.

Resources like “Zoo Escape” are useful, but are also lacking in terms of interactivity and depth, so they quickly get repetitive and end up underused.

There are numerous benefits in the use of smart boards and digital schoolbooks, so the goal of this project is not to replace them. There is, however, a lack of online and collaborative digital learning resources that are readily available to students and teachers. This project provides such a resource, as it involves the active participation of several students simultaneously, and teaches the importance of coexistence. It also offers a more engaging experience than most digital resources, as the students are involved in its creation and can see their colleagues’ contributions.

The installation can also be integrated with both technologies: it can be projected on the smart board for the entire class, or added as a resource to a digital schoolbook.

2.2.3 Digital Art in Education

Despite the increasing number of digital tools at teachers’ disposal, there's still a lack of digital art resources, which makes research on its benefits scarce. Nevertheless, there have been studies and experiments with its integration, that came with promising results.

In [12], Y. Gan studied how student-generated digital drawings could help students increase their understanding of the concept of light, using an online multimedia environment, Knowledge Forum (an educational software which supports knowledge building).

In this forum, students were encouraged to come up with ideas, illustrate them, and discuss them online with their peers over time, which ultimately resulted in a better collective understanding of the topic.

The study concluded that the students’ digital drawings fostered:

- Understanding of content area knowledge;
- Idea improvement and conceptual change;
• Progressive scientific inquiry for problem solving;

• Theory-building and modeling.

As the teachers have to be involved in this type of activity, strategies have also been proposed to help them decide which tools to use, when to use them, and how to evaluate their impact on students [13]. These resources are also important to consider, as many teachers lack training with digital tools and aren’t sure of how to integrate them in class.

2.2.4 Environment Study

One of the school subjects most suited for this project is Environment Study. As its program is already flexible and interdisciplinary, combining it with arts falls in line with its methodology.

As for the topics it covers, the first three years are the most relevant to this project, as that’s when students learn the most about animals and their natural environment.

According to the national program [14], students should be taught the following during those years, regarding living beings:

**First grade**

- Recognise animals and plants in different phases of their lives;
- Recognise different bodies of water;
- Identify colours, sounds and smells of plants, soil, sea, animals...

**Second grade**

- Recognise different environments that plants and animals live in;
- Observe and identify domestic and wild animals;
- Recognise external features of some animals (beak, fur, scales...);
- Collect information about how they live (what they eat, how they reproduce, how they move...).

**Third grade**

- Compare and classify animals according to their characteristics and life styles;
- Recognise the utility of plants;
- Identify factors that condition animal lives;
- Construct simple food chains.
Below are two pages taken from environment study schoolbooks, which show how these topics are usually taught (mostly through exposition or written exercises):

Besides these methods, the subject also encourages students to research and experiment on their own, as it considers practical activities a fundamental process in learning, and ultimately wants students to become active observers in their environment.

### 2.3 Animation

Computer-generated fish animation is commonly done in movies, video games and simulations with underwater environments, so there’s no shortage of references for realistic animated movements.

In [17], X. Tu and D. Terzopoulos propose an animation framework that uses a dynamic fish model with multiple nodal point masses and springs:

In order for the model to move like a real fish, it contracts and relaxes some of its springs (which serve as muscles), so its tail starts swinging. When the tail swings, a volume of water is displaced, generating a normal force that propels the fish forward.

The model also contains simplified versions of pectoral fins, allowing the artificial fish to control its pitching (up-and-down motion), yawing (side-to-side motion), and rolling (motion around longitudinal-axis).

The result is highly realistic, especially when combined with the fish’s behavioural modelling described in 2.4.

If, however, the number of fish were to be much larger (hundreds or thousands), applying forces to every affected node, while maintaining continuous perception of the environment would result in very
high computational costs. In cases like those, an approach like ABZÜ’s is more appropriate, albeit less realistic.

ABZÜ [18] is an adventure game that follows the journey of a diver as she explores the ocean. The game’s underwater environment is rich in terms of plant and animal life, hence the relevance to this project.

Since the game had, at times, thousands of fish on screen, each with up to a hundred joints, updating every joint per frame was too taxing. Instead, the developers decided to use static mesh instancing to reduce the amount of unique data per instance - a technique often chosen to animate grass and other foliage.

Static meshes don’t have joints, so skeletal animation isn’t possible with them. In order to create swimming cycles, vertex animations were done to the materials of each fish instead.

Vertex animation is done with vertex shaders, computer programs that can manipulate properties like position, colour and texture coordinates of vertices. They are a programmable stage of the rendering pipeline (a sequence of steps required to render a 3D image into a 2D screen). In the case of ABZÜ, these animations consisted of: a side-to-side translation, a yaw rotation, and panning rotations along the spine.
Other animations, such as biting and turning, were done with morph target animation, also known as blend shapes. In this type of animation, vertex interpolation is done between multiple deformed meshes:

![ABZU blend shapes for biting motions.](image)

### 2.4 Intelligent behaviours

When trying to artificially recreate an environment with several living creatures, a common approach is to create an agent-based system where each individual is represented by an autonomous agent. In this context, an agent can be defined as a piece of software that runs continuously without the intervention of a user or other program (autonomy), while sensing its environment and reacting to possible changes in it (reactivity) [20].

Craig Reynolds’ Boids [21] was one of the first biological agent-based models to simulate coordinated animal motions, such as bird flocking and fish schooling, which is implemented in this project.

Boids was developed in 1986, and has been extended and integrated multiple times ever since, mostly in computer graphics, but also in fields like robotics. The model consists of three simple steering behaviours which describe how an individual boid (flocking creature) moves based on the positions and velocities of nearby flock mates:
• Separation - steer to avoid crowding local flockmates;

![Figure 2.16: Separation [21]](image)

• Alignment - steer towards the average heading of local flockmates;

![Figure 2.17: Alignment [21]](image)

• Cohesion - steer to move toward the average position of local flockmates.

![Figure 2.18: Cohesion [21]](image)

While this model is great for simulating coordinated movements, it doesn’t capture the complexity of actual fish, and doesn’t take into account factors like fear of predators, hunger, desire for mating, etc.

X. Tu and D. Terzopoulos’s aforementioned model, [17], addresses these issues, by generating intentions based on the fish’s habits (liking brightness or darkness, cold or warmth, schooling or not...), mental state (based on hunger, libido and fear), and sensory information, which trigger behaviour routines accordingly:
The possible behaviour routines are: avoiding-static-obstacle, avoiding-fish, eating-food, mating, leaving, wandering, escaping, and schooling.

For fish who are prey, the chance of being hunted decreases if: they’re far from the predator, behind the predator’s field of vision, or schooling.

The figure below describes how the schooling routine is implemented:

With all of these possible behaviours, the framework manages to showcase highly realistic individual and collective motions of fish, while also capturing the unpredictable nature of their behaviour.
Chapter 3

Implementation

To measure digital art’s value as an educational tool, a web aquarium where participants can submit their drawings of fish, define their feeding and social behaviours, and observe them interacting with their environment was created. This aquarium can be accessed through a website, and is meant to enhance primary school students’ learning by showcasing certain concepts visually and improving creativity and a variety of other skills.

3.1 Solution overview

Teaching goals

This art installation tries to teach the following Environment Study concepts:

- The diversity of sea creatures: in size, shape, colour, habitat;
- What an underwater habitat looks like - thanks to the aquarium and its underwater effects;
- Social behaviours, such as schooling and isolating, and how they impact the environment (isolated animals are easier for predators to catch, for instance);
- Different feeding behaviours - carnivore, omnivore, and herbivore;
- Food chains - plants can be eaten, and herbivores can be eaten as well, so small food chains will be created;
- Life cycles - from birth to death, each creature will have a cycle and will change its appearance and behaviour according to the stage it’s in. After a long period of time without eating, or if the creature is killed - it will die and disappear.

Ideally, by allowing the student participants to play with web technology, actively influence the environment with their own creations and see their colleagues’ drawings, they will be more attentive than normal and pick up the visual cues on display meant to be taught. Creativity, emotional and social skills, and the ability to work with computers are also meant to be fostered with this experience.
**User**

Users are those who actively participate in the experience, by submitting a drawing of their own to the aquarium. For teaching purposes, the intended demographic consists of third grade students or below, who are still unfamiliar with concepts such as feeding behaviours, food chains and life cycles.

**Website**

The website serves as the main user interface. It’s a one-page website where all interactions take place, from submitting pictures to observing the aquarium:

![Website's default page](image)

Figure 3.1: Website’s default page.

To avoid setting up and maintaining a server, all functionalities of the website are achieved client-side, with JavaScript. Two APIs are used: Imgur’s (to upload each drawing) and Firebase’s (to store submission data), along with a WebAssembly compilation of ImageMagick (to remove the background colour of the drawings).
Upon successfully submitting a drawing, a preview of the image without its background is shown. Users are then asked a series of questions that define their sea creature’s behaviour:

- Which direction is the drawing facing, left or right?
- Regarding where it lives, is it demersal (lives most of its life at the bottom of the ocean, uses ambushes or camouflage to hunt...) or pelagic (lives in the pelagic zone of waters, is agile and swims long distances)?
- Regarding what it eats, is it a carnivore (only eats meat), herbivore (only eats plants), or omnivore (eats meat and plants)?
- In terms of social behaviour, is it social or solitary?

The first question ensures the fish will swim in the direction it’s facing.

The remaining questions are asked so there’s diversity inside the aquarium (promoting user expression and realism), and to familiarise the participants with some of the concepts that will be represented visually.

Questions are answered by pressing the buttons on the page. Whenever a user presses a button, its colour changes to green and the corresponding object property is set.

Once every question is answered, a "play" button appears on the WebGL player. Pressing it will allow the user to finally join the aquarium:
Unity WebGL Aquarium

The aquarium itself consists of a Unity WebGL application.

Other engine options were considered, such as Godot and Unreal, but Unity ended up being chosen due to familiarity with it, and the large amount of free resources on its asset store.

One of the initial goals of this application was to make it easily accessible, so that anyone with an internet connection could try it - no installation needed - regardless of their location. This would allow for the experience to take place anywhere - at school, at home, at a museum... the only requirement being a device with a compatible browser.

The application runs a 3D scene that represents the bottom of the ocean, decorated with over 20 free assets, including rocks, algae, corals, and more.

3D graphics were chosen for the environment as they added a feeling of depth and realism to the underwater scene that was harder to achieve in 2D.

As mentioned before, the application was built for WebGL - a JavaScript API that renders 2D and 3D graphics on web browsers. Unity WebGL content is compatible with most browsers, but it comes with some limitations: it doesn’t run on every mobile device (due to hardware limitations), and the WebGL graphics API is equivalent to OpenGL ES 3.0 - limiting the use of certain shaders.
3.2 Image Segmentation

Before a drawing is added to the aquarium, it needs to be cropped, resized, and have its background removed, which requires some level of image processing.

If this were to be done manually, software like Inkscape [22] would be a good fit, as its image tracing feature can turn sketches into vector images, and remove background colours automatically.

Image tracing, while not necessary for this project, has an upside compared to segmentation - the resulting vector images can be magnified/reduced freely, unlike raster images, which might have their artefacts exposed.

Below are two raster-to-vector conversions done with Inkscape:

![Figure 3.4: Raster Images.](image1)
![Figure 3.5: Vector Images.](image2)

While extremely convenient, Inkscape’s image tracing cannot be called from the command line, which makes automating the process difficult. For a project such as this one, where multiple images are to be submitted, processing each of them manually would be time-consuming, and participants wouldn’t be able to watch their drawings come to life almost instantly.

To make the submissions as quick and seamless as possible, an image segmentation script was created instead, so that no one has to manually edit them out whenever a user submits a drawing.

The script imports an ImageMagick library [23] as a JavaScript standard module, using its `convert` command [24] to trim the image and remove its background colour.

In the command call, the "-fuzz" operator is used to look for colours that are close to white in terms of colour space. Once found, they’re all replaced with the same colour, and subsequently made transparent, effectively removing the background:

![Figure 3.6: Image before background removal.](image3)
![Figure 3.7: Converted image with image segmentation script.](image4)
3.3 Imgur API

After the background removal, the converted image is then uploaded to Imgur [25], so it can later be downloaded by the Unity application through its URL.

This is done through scripting, by sending an HTTP request to the API’s endpoint [26] with the required authorisation header and image data.

The API returns a JSON response - containing the image’s URL - which is parsed and used to set the property “url” of the JavaScript object that will be sent to Unity.

Imgur was chosen, but other image hosting services with an API could also have been used.

3.4 Firebase API

Since the code is purely client-side, there’s no database where user data can be stored. There is, however, relevant information worth saving from each submission. In order to solve this, Firebase, a BaaS (Backend-as-a-Service), was used for data storage.

With each submitted object to Unity WebGL, a JSON object is simultaneously sent to a Firebase Storage bucket. Each object contains the chosen features on the website, and the image URL associated with them.

Firebase was chosen because its free plan offered strong security and enough storage (5 GB) for this project, but other services could’ve been used instead.

3.5 Interacting with Unity WebGL

For the fish to spawn in the aquarium, its image data, and the characteristics defined by the user have to be sent over to the Unity WebGL app. This is done by calling a function on the Unity instance from JavaScript. The call is as follows:

gameInstance.SendMessage("Launcher","ReadURL",JSON.stringify(obj));

*Launcher* is the scene game object to which the message is sent. *ReadURL* is a method on that object’s script, which creates a *Fish* object based on the JSON string.

*ReadURL* also activates a play button on screen which, when clicked, connects the application to the Photon cloud network and begins the experience.

3.6 Network

For there to be interactions between the users’ drawings, such as eating one another, aspects of the Unity scene have to be synchronised across the network, like transforms, animations, and properties.

To avoid the costs of setting up and maintaining a server, the project uses Photon Unity Network (PUN) [27] and Photon Cloud to create its network, which are part of the Photon framework [28].
PUN is a package which re-implements and enhances the features of Unity’s built-in networking. Photon Cloud, on the other hand, is a software as a service solution (SaaS) compatible with PUN, that takes care of hosting, server operations and scaling.

Since the application is built for WebGL, there are some limitations regarding networking. Due to security implications, JavaScript code does not have direct access to IP Sockets to implement network connectivity. As a result, the .NET networking classes (i.e., everything in the System.Net namespace) are non-functional in WebGL [29].

Because of this, the communication protocol chosen for this project was WebSockets over SSL/TLS. Unity Networking, also known as UNet, was initially considered as the multiplayer solution for this project, but eventually decided against after its deprecation was announced. Lack of support and documentation for WebGL networking were also factors in the decision.

Unity’s Matchmaker feature, which uses relay servers, is not compatible with WebGL, so it was also not an option.

3.6.1 Network Implementation

Whenever a user presses the play button, a connection is made to a Photon Cloud server. If successful, the application looks for rooms with the same app ID. If no room is found, then a room is created instead. PUN’s free plan has a limit of 20 concurrent users, so the maximum number of users per room is 20 as well.

Once a room is joined, three game objects are instantiated by the Photon network: the parent object, which contains all scripts related to fish behaviour, and two “child” objects: one for holding the fish’s Sprite, and another for the fish’s “mouth”. Each of these network objects has a PhotonView component associated with it, which contains an identifier for the object across the network.

The Sprite is held by a child object instead of the parent, because its z-axis does not match the forward direction in which movement is applied. The parent object is therefore used to fix the orientation of the sprite, and holds all movement-related scripts:

![Figure 3.8: Forward direction before axis correction.](image1)

![Figure 3.9: Forward direction after axis correction.](image2)

Since there is no way of passing hierarchy of objects through PUN, a custom implementation of
OnPhotonInstantiate was done, where instantiation data containing the parent’s network ID is passed to each child object, and used to parent them before they’re spawned.

Once each object is spawned and the hierarchy is formed, the birth animation’s trigger and swimming script are activated if the objects belong to the local client.

If the objects belong to other clients, the state of their animations is checked via Photon’s custom properties - a key-values hash table where values are synced and cached on the clients, so they don’t have to be fetched. These properties are also used to store the characteristics of the fish (which direction it’s facing, if it’s social or not, etc..).

Once the animations end and the fish is ready to transition into an adult, a UnityWebRequest is made to download the image drawn by the student and transform it into a Sprite, which replaces the juvenile sprite on the child object.

Other clients are notified of this through the property isSpawning, which is set to false once the animations are complete. Since sprite changes are not synchronised across clients, the change has to be done manually by checking the parent object’s properties (which contain the image URL and direction the drawing is facing), and then downloading and replacing the sprite of its child object accordingly.

### 3.7 Scene

The main scene consists of an aquarium in which the drawn fish spawn and swim in. This area is limited by invisible bounds, preventing them from wandering off, and increasing their number of interactions:

![Figure 3.10: The aquarium’s bounds.](image)

In order to add realism to the underwater environment, a fog effect and a simplex noise shader were added to the camera [30] [31].

![Figure 3.11: Aquarium without noise shader and fog.](image)
The fog effect is part of Unity’s post-processing stack, and is achieved by overlaying colours (in this case, blue) onto objects depending on their distance to the camera.

The simplex noise shader implements an n-dimensional noise function similar to Perlin noise \[32\], resulting in a wobbly effect that creates the illusion of being underwater.

### 3.8 Models and Animation

As the aquarium was designed as a 3D environment, converting the students’ drawings into 3D models was initially considered to maintain consistency.

This type of work has been done before, in sketch-based modelling applications such as Teddy \[33\], where the users’ strokes are interpreted by the software and transformed into 3D polygonal surface models:

![Figure 3.13: Creating a 3D model with Teddy \[33\].](image)

Or with the use of machine learning, like in Nvidia’s DIB-R \[34\], a differentiable rendering framework wrapped around a neural network, which can predict shape, texture, and light from single images.

To achieve it in this project, texture mapping previously made 3D fish models was considered, or having the students draw body parts within certain bounds, and later combining them with depth maps/using extrusion.

Ultimately, these methods were decided against, as they would limit the students’ freedom of expression, potentially deform the original drawings, and move away the focus of the project. For these reasons, the drawings are converted into 2D models instead.

As seen in 2.3, there are several techniques for animating 3D models of fish, such as skeletal animation, morph target animation and procedural animation. For 2D models without rigs, as is the case in this project, there aren’t as many options, but animations were still a key factor in making the experience
engaging for students, making the swimming motions realistic, and breaking down the complexity of a fish’s life cycle.

For frame-by-frame animations like biting, growing, and entering stealth, each animation clip was created with Unity’s in-built system, by changing certain object properties over time, like position, scale, colour, etc.

For the biting animation, since there’s no prior knowledge of the drawings’ mouths, doing morph target animations like in ABZÚ is not possible. Instead, a frame-by-frame animation of a biting motion is played in front of the fish whenever it gets close to its food. Since each drawing will have different dimensions, the biting animation controller’s position (in front of the fish) is found by moving the controller forward by half of the sprite’s width.

The life cycle starts off with a clip called birth, where a hand-drawn egg slowly hatches into a larvae through a series of sprite changes. Following this, the fish is ready to swim, so a growing animation clip begins, where more significant changes happen: the scale of the sprite starts to gradually increase, a material with the swimming motion shader is added, and the larvae sprite is changed into sprites with smaller yolk sacs, until eventually resembling a juvenile fish (where fins are visible and the yolk sac is gone). The cycle plays out in about 2 minutes, which is naturally unrealistic for a process that can take months to complete, but the goal is to simply show some of the most significant physical changes that happen during its course, which the animation does.

Since plenty of demersal fish species use camouflage and ambushes to hunt, whenever a fish of this type gets close to rocks and plants, an animation is triggered which makes the fish fade in and out of them. This is achieved by changing the fish material’s colour (its alpha value, to be specific) with a fragment shader.

To animate the fish’s swimming cycles, the following vertex shader was used:

```cpp
v2f vert (appdata_base v) 
{
    v2f o;
    if (v.vertex.x < _HeadLimit)
    {
        v.vertex.z += sin((v.vertex.x + _Time.y * _Speed) * _Frequency) * _Amplitude * 0.05;
    }
    else
    {
        v.vertex.z += sin((v.vertex.x + _Time.y * _Speed) * _Frequency) * _Amplitude;
    }
    o.pos = UnityObjectToClipPos(v.vertex);
    o.uv = TRANSFORM_TEX(v.texcoord, _MainTex);

    return o;
}
```

In this shader, the vertex position (z-coordinate) is changed by adding a sine wave function to its original value. The wave moves along the x-axis a distance of _Time.y (current time in Unity) * _Speed. The wave’s speed, frequency and amplitude can all be set in the shader within certain ranges. The
if-clause ensures the motion is reduced for vertices located on the head area of the fish ($v$.vertex.x $<$ \_HeadLimit), as most propulsion should come from the fish's caudal fin to make the movement realistic.

In order to synchronise the animations across the network, a PhotonAnimatorView script is added to each object with an Animator. The script is also added as an observable entry to the game object's PhotonView component, so its parameters (like animation triggers, for example) are synchronised discretely. This means each parameter gets sent 10 times a second to other clients, who then pass it to their local Animators.

3.9 Behaviours

3.9.1 Swimming

Once the spawning animation cycle ends, a script is added to the game object which causes the fish (in its larval state) to start swimming.

The script begins by checking what type of fish it's attached to, and sets its speed accordingly. The value is randomly generated within a range, but bottom feeders have a lower maximum speed than pelagic fish, as they are typically slower.

If the fish is social, a flag (canSchool) is also activated so that its movement, instead of randomised, follows the school.

The figure below describes the overall logic of the script:

![Swimming script logic](image)

Figure 3.14: Swimming script logic.

Turning, in this case, means turning away from the aquarium's bounds. These bounds are invisible walls the fish can collide with. Whenever this type of collision is detected, spherical interpolations ("SLERP")s) are done between two vectors: the current rotation of the object, and a rotation towards the centre of the aquarium.

```
transform.rotation = Quaternion.Slerp(transform.rotation,
    Quaternion.LookRotation(Vector3.zero - transform.position), Time.deltaTime);
```

These rotations take priority over all others, to prevent erratic behaviours such as swimming against walls.
If the fish is chasing a target in order to feed itself, and is not close to any walls, its rotation is updated as follows instead:

\[
\text{transform.rotation} = \text{Quaternion.Slerp(transform.rotation, Quaternion.LookRotation(goalPos - transform.position), Time.deltaTime)};
\]

Contact with the object being chased is detected with a ray cast, and triggers a biting animation.

If the fish is not turning, chasing, or schooling, the direction it's facing and speed are randomised every 10 seconds. The range depends on fish type, as bottom feeders are limited to the bottom of the aquarium.

### 3.9.2 Feeding

While the fish has a yolk sac attached to itself (visible in its sprite up until larval stage), there’s no need to hunt for food, as the sac provides enough nourishment. However, once it reaches juvenile stage, the yolk sac has been fully absorbed, so the fish needs to start feeding itself.

Fish that are juvenile or younger do not interact with other fish in this simulation. This is an intentional sacrifice in terms of realism, so that every student gets a chance to see their creation’s complete life cycle. In a real habitat, fish eggs and larvae would be easy prey for predators.

Once a user’s fish becomes an adult, it’s tagged as a potential target for carnivores and omnivores, and a feeding script is attached to it that determines how and when it hunts.

The script starts a `hunger` field and repeatedly invokes two methods: one which increases the field every 4 seconds (`IncreaseHunger`), and a method which finds a valid target within the aquarium every 15 seconds (`FindTargets`). Targets, in this case, are game objects tagged as `Meat` (adult fish from other users) or `Plant` (algae decorating the aquarium). Plants are a limitless resource, whereas the quantity of meat depends on the number of online users. `FindTargets` generates a target list based on fish type and feeding behaviour, as demersal and pelagic fish require different lists, and so do omnivores, carnivores and herbivores. It then proceeds to randomly select one of the list elements as the target to chase (`chasee`). The method is repeatedly invoked, as a new target may have to be selected under certain circumstances: another fish eats the previous target, or the fish gets stuck while chasing, for instance.

After a full minute without eating, fish start chasing their selected target. While moving towards it, `hunger` continues to increase. If, after a minute of chasing, the fish hasn’t made contact with its food, it will die from hunger and disappear from the aquarium.

If the target is a plant and contact is made with it, `hunger` is immediately reset. When the target is a fish, the attack has a chance of failing. If the target can school, this chance increases proportionally to the size of the school, as fish generally school to protect themselves from predators. This prevents carnivores from becoming too oppressive in the aquarium, while enforcing a level of realism, as plenty of herbivores are capable of escaping and defending themselves.

If an attack on a fish is successful, `hunger` is reset, and a RPC (method-call on remote clients in the same room) is made with the target’s network identifier as a parameter. If a remote client’s network identifier matches it, that client’s game object destroys itself locally and remotely.
If the attack fails, the hunter is forced to retreat for some seconds to recover its stamina.

3.9.3 Schooling

Fish defined as "social" by students can engage in schooling behaviour with other fish of the same type. In nature, this behaviour is usually done for the following reasons: to defend against predators, to increase chances of finding food and a mate, and to swim more effectively. In this project, it functions as a self-defence mechanism only (the chance of a successful attack by a predator is reduced proportionally to the size of the school).

This behaviour is integrated into the swimming script. Much like in Craig Reynolds' Boids [21], the fish checks for neighbours within a certain distance in every Update. The major difference is that there's only one possible neighbour for each fish. This neighbour is an invisible network object whose movement is synchronised for all the clients inside the same room, and functions as the "goal position" for the school. A different neighbour exists for demersal and pelagic fish. If the fish is in range of its neighbour, it starts steering towards it and reduces its speed to match the school (alignment). As the goal position is the same for every client in the room, the fish stick close to each other as long as they're within "neighbourhood" distance (cohesion). Fish that get too close to their respective neighbour are repelled to avoid crowding (separation). This method differs from Boids, where each member steers towards the average position of its neighbours.

This approach was taken due to the way the network is implemented, which makes synchronising other clients' speeds and forming groups difficult without cluttering the network with messages and decreasing performance. It comes with two downsides: the fish may get too close to each other as they're only repelled from the goal position, and only two schools are ever formed in the aquarium (one for demersal, and another for pelagic fish).

Figure 3.15: Schooling fish.
Chapter 4

Results

This chapter details how the first tests of the project went, with an actual class of students, along with an analysis of the results.

4.1 Conditions

The tests were conducted at a local school, with a class of 23 students in 3rd grade. They took place over the course of three days, and students had to be divided into small groups (2-6) so that classes weren’t disrupted too much.

To speed up the process, students were asked to prepare their drawings before the days of testing, not knowing what would happen to them. They were asked to draw a single fish in a white background and to colour the fish at their own discretion.

The drawings were then scanned by their classroom teacher and sent out to be reviewed. It’s important to note that some of the drawings didn’t comply with the rules - some had coloured backgrounds and multiple fish. In those cases, the drawings were manually edited.

On the days of testing, a room would be prepared with multiple laptops (around 9) with two pages loaded: one with the website, and another with a collection of all the submitted drawings, so students could pick theirs.

Groups of students would then be called and given instructions on how to use the website and run the application. Most students were able to do it on their own, as they had obtained experience with computers either at home or in their programming and robotics classes. There were, however, some students who still struggled with reading and writing. For those students, individual assistance was given.

4.2 Method

To assess if the interactive installation managed to meet its goals, a questionnaire was handed out to each student after their experience: [A]. Students were left to answer it on their own, and only given help
if they struggled reading the questions or writing their answers.

4.3 Analysis

The first question addressed the usability of the website. The goal was to have students be able to navigate and interact with the website independently after given instructions. Most of the students (91.30%) said it was easy to do so. The remaining answers were non-applicable (left blank).

![Figure 4.1: Question 1 results.](image1)

If a large number of participants had struggled with the website, then the project would have no practical use for primary school students.

The second question was asked to gauge the emotional response of students to their drawings turned digital. It was done with a rating scale from 1 (hated it) to 5 (loved it):

![Figure 4.2: Question 2 results.](image2)

Out of the 23 students, 21 said they loved it, 1 felt indifferent, and another hated it.

Previous studies have shown that positive emotions improve learning performance and satisfaction [35], so this high of an emotional response indicates that the project could be used in that sense. Students ended up very focused on what was happening on screen during the experience, as they felt attached to their own (and to an extent, their colleagues’) drawings.

Questions 3-6 were theoretical and used to evaluate if students learned from the experience.
Question 3 consisted of a multiple-choice question with four possible life cycles for a fish. The results were as follows:

![Question 3 results](image)

The students in question had yet to learn at school how a fish’s life cycle plays out, so one of the goals of the experience was to teach them that with its animations. The results were positive, with 18 of the 23 students having correct answers. Those who got it wrong mixed up the larvae and juvenile stages of the fish, where the biggest differences are the size of the yolk-sac and the appearance of small fins. Since the visual cues were small, the transformation might’ve been less memorable for those students.

For question 4, students were asked what happened to the size of the fish as they grew, and all students chose the right answer. This was visually represented with animated augments in size every time a fish reached a different stage of its life.

![Question 4 results](image)

Question 5 is the hardest to analyse, since the definition of “pelagic fish” was displayed on the website and then reinforced visually in the aquarium. Students might’ve simply remembered the definition, but they were not encouraged to memorise anything, and were not aware of the questionnaire. Some students even made their choices without reading the definitions. The idea was to make them remember the type chosen on the website, the area in which their drawing was swimming, and then deduce the answer based on that.
Question 6 - “Why do fish school?” - was the hardest question of the questionnaire, as it didn’t just require good memory, but also deductive reasoning. While higher protection from predators was implemented in code for schooling fish, its visual effect would hardly be noticeable if the schooling fish numbers were small, which was the case during these tests. As such, the answer would have to come from observing schooling fish go their separate ways whenever they felt hungry, instead of securing food together. The results were surprisingly positive:

Note: Questions 5 and 6 were only possible because the number of social and pelagic fish within each group was high. Below are the distributions, as recorded in Firebase’s storage:

<table>
<thead>
<tr>
<th></th>
<th>Demersal</th>
<th>Pelagic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>Percentage</td>
<td>21.74%</td>
<td>78.26%</td>
</tr>
</tbody>
</table>

Table 4.1: Fish type distribution.

<table>
<thead>
<tr>
<th></th>
<th>Carnivore</th>
<th>Herbivore</th>
<th>Omnivore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>7</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Percentage</td>
<td>30.43%</td>
<td>30.43%</td>
<td>39.13%</td>
</tr>
</tbody>
</table>

Table 4.2: Feeding behaviour distribution.
Lastly, since this was also an art project, students were asked if the aquarium, when decorated with their drawings, looked aesthetically pleasing, which the majority agreed with:

![Figure 4.7: Question 7 results.](image)

Besides the seven obligatory questions above, two optional open-ended questions were also in the questionnaire, to gather feedback about what the students liked and/or disliked from the experience.

Very few complaints were left. Two of the students didn't like the assets used to decorate the aquarium. The vast majority of the participants commented on how they disliked seeing their fish die, which once again shows how attached they felt to their creation. As for what they liked, the predominant answer was seeing their fish eat their colleagues'. This reinforces the idea that students are highly interested in multi-user experiences, something that very few digital resources currently offer.

Despite how positive the results were (approximately 86% of the theoretical answers were correct), this small of a sample size makes the data unreliable, so general statements cannot be made regarding the effectiveness of interactive digital art when used for teaching. Is it as effective as watching a video or reading a manual? Did the students retain what they learnt weeks or months after? For more conclusive analysis on this topic, more tests would have to be made in the future, at different schools, along with periodic re-tests.

There were, however, clear takeaways from the experiment, which successfully highlighted the benefits of implementing digital art in class:

- Students were able to express themselves creatively;
- Students became more tech-savvy while using the website;
- Students enjoyed the social aspect of the activity, something that is easily implemented in digital art projects;
- Students were highly immersed during the activity, meaning it's suitable for learning;

<table>
<thead>
<tr>
<th>Social</th>
<th>Non-social</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>15</td>
</tr>
<tr>
<td>Percentage</td>
<td>78.26%</td>
</tr>
</tbody>
</table>

Table 4.3: Social behaviour distribution.
- Students with learning difficulties were still able to enjoy and learn from the experience;
- Students expressed interest in revisiting the web aquarium from home;
- Students expressed strong emotional responses to what happened on screen.
Chapter 5

Conclusions

This dissertation has discussed how digital art is currently unappreciated and underused in the field of education, and how its integration can bring innovation to classrooms and multiple benefits to students.

In order to showcase this, an interactive digital art solution was created with the intent of teaching primary school students about sea life, and a questionnaire was handed out to them afterwards to measure their experience.

Students were very engaged throughout the entire experience, and the questionnaire’s results were very promising for a first sample. Not only were most of the theoretical answers correct, but the majority of students also found the website easy to use and loved observing their fish and the aquarium.

5.1 Achievements

The most notable achievements of this work were:

- Successfully creating an engaging interactive art installation, which can be used as an educational resource, or for artistic purposes;
- Combining knowledge of web design, game design, artificial intelligence and animation into a single project;
- Successfully teaching the majority of students who participated in the experience;
- Showcasing some of the benefits of digital art implementation in classrooms.

5.2 Future Work

While the project had overall positive results, there were still mishaps in its implementation which could be fixed in the future.

The first and most glaring issue is the lack of a self-hosted server. The use of a networking framework and purely client-side script code implies there’s no control over other users’ submissions and that
there’s restrictions in place regarding network usage (a maximum number of concurrent users). Without
the limitation on concurrent users, online collaborations between different classes/schools would be
possible. By setting up a server, the Imgur and Firebase APIs would also no longer be necessary,
since images and data could be stored server-side. The framework also made it difficult to implement
the schooling algorithm (an adaptation of Boids), as synchronisation of speed and other properties of
neighbours had to be done manually, and required an excessive amount of messages back and forth
between clients.

The website is currently in portuguese only. It should be changed into a multi-language website in
the future, so it can reach more schools and students.

Most of the implemented behaviours have room for improvement. The notions of libido and fear,
present in [17], for example, are not present in this project. Their addition, along with corresponding
mating behaviours and self-defence mechanisms, would significantly improve realism and allow students
to learn more about reproduction.

The project could also be expanded in order to include other aquatic animals besides fish, such as
turtles or jellyfish, or even a different habitat altogether.

As mentioned in 4.3, more testing should be done in the future to ensure proper statistical analysis
and the ability to generalise the project’s findings.
Bibliography


39


Questionário para alunos

1. Foi fácil utilizar o site?
   ○ Sim
   ○ Não

2. O que achaste do peixe?
   1 2 3 4 5
   Detestei ○ ○ ○ ○ ○ Adorei

3. Escolhe a ordem de crescimento certa:
   ○ Larva Ovo Alevino Juvenil Adulto
   ○ Larva Alevino Ovo Juvenil Adulto
   ○ Ovo Larva Alevino Juvenil Adulto
   ○ Ovo Alevino Larva Juvenil Adulto

4. O que aconteceu ao tamanho do peixe enquanto crescia?
   ○ Aumentou
   ○ Diminuiu

5. Os peixes pelágicos nadam, geralmente:
   ○ No fundo do mar
   ○ Em mar aberto

6. Peixes que nadam em conjunto (sociais):
   ○ Ficam mais protegidos de predadores
   ○ Partilham a comida

7. O aquário estava bonito?
   ○ Sim
   ○ Não

8. O que gostaste mais?

9. O que gostaste menos?