Developing A Voxel Building System to Help Foster Engagement In The Arts

Swaiba Faisal
Department of Computer Science and Creative Technologies
University of the West of England
Coldharbour Lane
Bristol, UK
Swaiba2.faisal@live.uwe.ac.uk

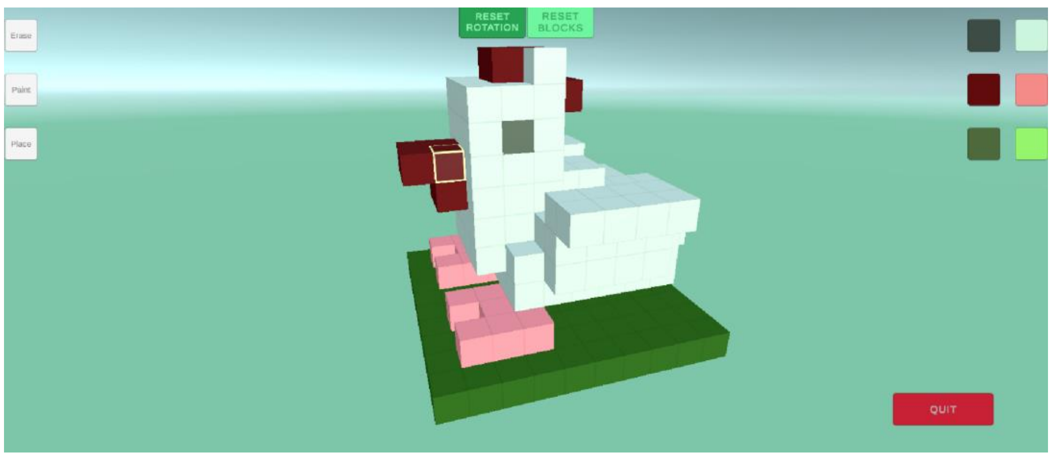

Figure 1. Building a chicken using the final prototype of the building system

Abstract

In response to a client's request regarding whether Games Technology can be used to foster engagement in the arts, research was conducted into The Arts, building systems in Video Games and Voxels. A voxel building system was then developed (*Figure 1*), and an evaluation of its success carried out based on technical functionality and theoretical principles.

Author Keywords

Voxels; Unity; STEAM; Creativity; Games Technology.

Introduction

Science, Technology, Engineering, Arts and Mathematics (hereafter STEAM) is described as "a more interdisciplinary approach to STEM education" through the addition of The Arts into the aforementioned areas of study (Wajngurt and Sloan, 2019). In response to a client's inquiry as to whether Games Technology can be used to foster engagement in STEAM, this project aims to conduct research into video games' implementation of voxel systems, with the intention of developing a building system that aids in fostering engagement within The Arts.

This report delves into the research, planning and implementation of the proposed artifact, as well as an evaluation of its success in answering the client's question. Suggestions are then made for expanding upon this project's foundations.

Background Research

The increase in consumption of interactive digital content within younger demographics is notable. Figures published by The Association for UK Interactive Entertainment (UKIE) highlight that 72% of 15-24 year olds across key markets in Europe and the UK play video games (UKIE, 2021).

In particular, Minecraft (Microsoft, 2011) boasts over 300 million copies sold (Nichols, 2023). Studies find that the game can be instrumental in fostering creativity and divergent thinking through its building system (Diaz *et al*, 2020); despite this not being the core focus of the game. Using said system is described as "analogous to playing with real life Lego blocks", with players able to place Voxels anywhere on a matrix – like grid to build complex models (*Figure 2*). While it is true that this study's focus group was Undergraduate students, the conclusion still stands when considering that they were not familiar with the game before playing.

The term "Voxel" is attributed to a cube-shaped object, often referred to as a "three-dimensional instance of a pixel" (Hughes *et al.*, 2014). While Minecraft is rendered using polygons, its building system is arguably still voxel based due to its placing of voxel shaped "blocks" onto a three-dimensional grid (*Figure 2*).

Figure 2. A screenshot from Minecraft, showing off the building gameplay (Microsoft, 2011)

A research review of The Arts, published by the Department of Education (2023), asserts that a key principle of the subject is "Practice", with secondary school education focusing on generating creative output alongside technical theory. Taking this into account, focus was placed onto developing an artifact that facilitates this output.

Scope and Objectives

The time allocated for the planning, development and evaluation for this research project was roughly 10 weeks (Ersotelos *et al*, 2024). This period was deemed too short for the development of an artifact that successfully fosters engagement in multiple areas of STEAM. Consequently, the decision was made to focus on engaging younger demographics in The Arts specifically.

This project aims to create a voxel-based building system, adjacent to Minecraft's, in which the user can freely practice creating their own voxel models.

Key development objectives for the artifact:

- Create a custom Voxel Object

- Implement a way of selecting Voxels based on mouse position

- Add functionality for "breaking" and "placing" voxels

- Add methods for changing a Voxel's colour

- Implement systems for displaying the Voxel grid and Index

Notably, this voxel system is stripped of the complexities added to it by its other iterations, like in Minecraft or Teardown (2022); there are no physics, or interactions between Voxels. This decision was made due to aforementioned scope constraints but is worth exploring given more time.

Methodology

Engine

The artifact for this project was developed in *Unity 6* (Unity Technologies, 2024). This engine was chosen as compared to other versions due to its increased render performance (Best, 2024), which is necessary for a project that requires effective rendering of multiple Game Objects.

Prefabs

The "Face" prefab consists of a quad mesh and a "place point" transform. A script attached to the parent contains functions for changing the material of the quads.

The "Voxel" prefab contains 6 "face" prefabs, arranged to resemble a cube. The script attached (*Figure 3*) contains references to the aforementioned faces alongside shown functions.

```
public class VoxelFunctionality : MonoBehaviour
{
variables
    [SerializeField] List<GameObject> m_faces = new List<GameObject>();
    1 reference
    public void UpdateVoxelType(VoxelData _newVoxelData)...
    2 references
    public void SelectVoxel()...
    2 references
    public void DeselectVoxel()...
    2 references
    public void UpdateHighlightedMaterial(int _index, Material _material)...
    2 references
    public void UpdateBaseMaterials(Material _material)...
    1 reference
    public void Break()...
    1 reference
    public void Paint(VoxelData _voxelData)...
}
```

Figure 3. "Voxel Functionality" script.

Scriptable Objects (Unity Technologies, 2024) are used to store data specific to the different Voxel types. The Unity Engine Documentation (2024) highlights their ability to save large amounts of data, independent of class instances. Consequently, a "VoxelData" base script was created containing this information (*Figure 4*).

```
public class VoxelData : ScriptableObject
{
    [SerializeField] E_VoxelType m_type;
    public Material m_material;
    0 references
    public E_VoxelType GetBlockType { get { return m_type; } }
    2 references
    public Material GetMaterial { get { return m_material; } }

}
```

Figure 4. “VoxelData” base script

Interactions
“Selection Script” contains the functionality for selection and calling the relevant functions based on the value of the “brushType” variable.

Selection involves performing a Raycast (Unity Technologies, 2024) from the Camera position to the mouse world position and storing the objects hit.

When “OnVoxelClicked” is called the “BrushType” Enumeration (Unity Technologies, 2024) is cycled through, and the relevant functions are called based on this value.

“Breaking” a voxel destroys the “Voxel” prefab entirely.

“Painting” a voxel involves the selection script calling the “UpdateVoxelType()” function inside the “currentVoxelScript”, passing through the selected “VoxelData”.

“Placing” a voxel is handled by the “Voxel Placer” script. The Selection Script will call the “OnVoxelPlaced()” function within the Voxel Placer and pass through the “currentFaceScript” ’s spawn point transform and selected Voxel Data. The following actions are then performed:

1. A “Voxel Prefab” is Instantiated.
2. Its transform values are set
3. It is added to the “Voxel List” list
4. Its “VoxelData” is set

Rotation Functionality
To showcase the artifact’s functionality, a rudimentary rotation system was created. The user can rotate and move all voxels across the x and y axes.

Developing a more complex camera system was considered. However this was not part of the development objectives, and was ultimately deemed outside the scope of this project.

The ”ResetRotation” function resets all voxel’s rotation and transform values. This was added to mitigate issues with setting the voxel’s center of rotation.

Index & Grid system
The final goal listed in the “Scope and Objectives” section is the addition of a grid & index display system. This would display the grid index of a particular voxel on the game screen. The intention behind this, as discussed in the initial progress report video (*Figure 5*), was to encourage users to think mathematically.

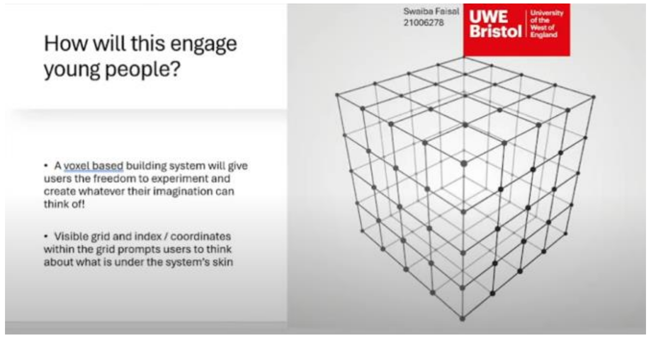

Figure 5 Slide from the initial progress video discussing the index system. (Faisal, 2024)

Early versions of the project featured both (*figure 6*). However, due to the reasons discussed in the following paragraphs, they were eventually cut.

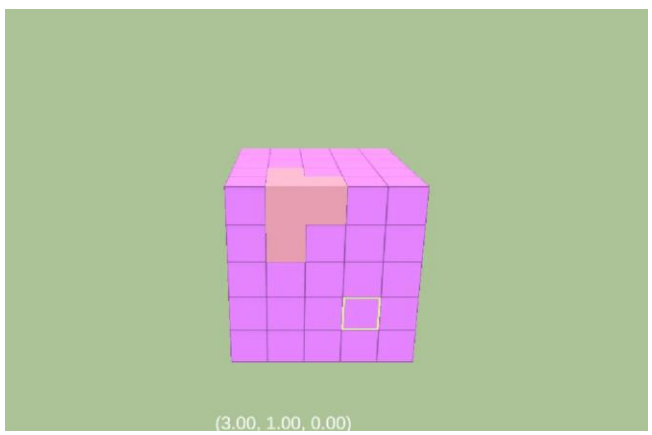

Figure 6. Older version of the artifact, showing cut systems

The visible grid was possible due to how engaging with voxels was previously implemented. "Breaking" voxels initially involved changing their base material to a transparent outline. "Placing" voxels involved switching the material to the voxel's new "VoxelData". This caused issues with the Raycasting, as the user was not able to access voxels that were obstructed by the transparent ones above them. It was also not viable to simply turn the colliders off, or disable the object, as there was then no way to access the object again to "place" it. A workaround could involve multiple Raycasts and Layer Masks for the different functions, but the additional time and complexity this would add to development is too large for the scope of this project. Consequently, the interaction system was altered, and the grid system was discarded.

The decision to remove the index system was done partly due to its lack of value, as no other systems relied on it and the user is able to build without it, alongside Moron's assertion that the amount of creativity fostered by a game was inversely proportional to its level of complexity (2010).

Outcomes

As evidenced in *Figures 1, 2 and 3*, the outcome of this project is a system that allows the user to build freely. UI buttons on the left side of the screen alter the brush type - see "Methodology- Voxel interactions" -, and buttons on the right side set the selected VoxelData. The "Grid Dimensions" and "VoxelSize" variables are also adjustable in editor, allowing for even more creative control.

It can be argued that this project was also successful in fostering engagement in The Arts as it allows for a variety of different models to be made through the placement of cube-shaped blocks. These can be traditional voxel models (*figure 8*) or "pixel art" style models (*figure 7*). The user is also able to modify the size of the initial voxel grid, as well as the voxels themselves, giving them further control over what they would like to create.

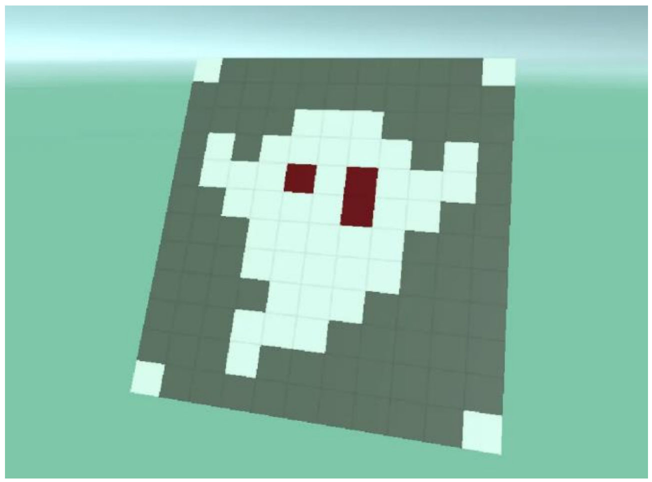

Figure 7 Pixel Art style ghost, created in artifact

Figure 8 3-Dimensional ghost, created in artifact

However, while the user can freely and effectively manipulate the model's form, they are restricted in how they choose to color their creations. Another key aspect of Arts education is Color Theory (Montés *et al*, 2024). Without a variety of colors, the user is unable to effectively engage with this, reducing the artifact's overall effectiveness in fostering engagement in the subject. A way to improve this, given more time, would be the implementation of a color wheel and palette system. This would give the user full control over what colors / combinations they use in their creations, also allowing them to more effectively practice Color Theory.

Evaluation

In summary, this project was successful in its attempt to create a voxel building system in which the user can place, break and paint voxels freely. Although the Index and Grid Display system were discarded, this was due to technical issues and a reevaluation of their compatibility and use within the project.

While the artifact was also successful in fostering engagement in the arts through its facilitation of complex model creation, the level of success it achieves is limited by the low variation in color choices. Given more time, a more complex color selection system would greatly improve the success level of this project.

While this conclusion is theoretically sound, the lack of user testing reduces its solidity. Given more time, and an increase in scope, it would be beneficial to conduct user tests to more effectively evaluate the outcomes of this project. Through analyzing the collected data, a more informed decision can be made regarding the project's future direction.

References

Best. M (2024) *Unity 6 is here: see what's new.* Available from: https://unity.com/blog/unity-6-features-announcement

Department for Education, Training and Skills (2023) *Research Review Series: Art and Design* [Online] England: Department for Education. Available from: https://www.gov.uk/government/publications/research-review-series-art-and-design/research-review-series-art-and-design#pedagogy

Diaz D.M. Saorin J.L, Carbonell-Carrera C, Cantero J.T, (2020) *Minecraft: Three Dimensional construction workshop for improvement of creativity*. Technology, Pedagogy and Education [online]. 29:5. Pp. 665-678. [Accessed 2 December 2024]

Ersotelos.N, Savickas.L (2024) Advanced Technologies: Module Handbook Bsc Games Technology. [online]. Available from: https://blackboard.uwe.ac.uk/ultra/courses/_365617_1/cl/outline

Hughes, J.F., Akeley, K., Foley, J.D., Sklar, D.F., McGuire, M., Van Dam, A. and Feiner, S.K. (2014) *Computer Graphics Principles and Practice*. 3rd edition. [online]. 1, Addison-Wesley Professional. [Accessed 12 November 2024].

Microsoft Studios (2011) *Minecraft* [Video Game]. Microsoft Studios. Available from: https://www.minecraft.net/en-us [Accessed 1 November 2024]

Montés, N., Barquero, S., Martínez-Carbonell, A., Aloy, P., Ferrer, T., Pantaleón, D.R., Millan, M. & Arturo del, S.S (2024). Redefining STEAM to STEAM ∀H (STEAM for All Humanity) in Higher Education*. Education Sciences* [online] 14(8) pp. 888. Available from: https://www.proquest.com/artshumanities/docview/3097896987/520E59E6331648E5PQ/10?accountid=14785&sourcetype=Scholarly%20Journals

Moron. (2010) UN PRINCIPIO DE INTERVENCIÓN EDUCATIVA: EL JUEGO Y LOS JUGUETES EN EDUCACIÓN INFANTIL. 10, pp. 1–9.

Nichols, S.(2023). *Minecraft Live 2023: the recap!* Available from: https://www.minecraft.net/en-us/article/minecraft-live-2023--the-recap- [Accessed 2 December 2024].

Wajngurt, C. and Sloan, P. (2019) *Overcoming Gender Bias in STEM: The Effect of Adding the Arts (STEAM)*. *InSight: A Journal of Scholarly Teaching* [online]. 14, pp. 13–28. [Accessed 1 November 2024]

Tuxedo Labs (2022) *Teardown* [Video Game]. Tuxedo Labs. Available from: https://teardowngame.com [Accessed 1 November 2024]

Unity Technologies (2024) *Unity Engine Documentation*. Available from: https://docs.unity3d.com/ [Accessed 1 November 2024].

Unity Technologies (2024) *Unity 6* [Computer program]. Available from: https://unity.com/releases/unity-6

UkieColm (2021) Player Diversity and Demographics. Available From: https://ukiepedia.ukie.org.uk/index.php/Player_Diversity_%26_Demographics#:~:text=73%25%20of%206-10%20year,of%20people%20aged%2045-64.