

**FINAL YEAR PROJECT**

**“FALLING”**

**BY**

**Abrar Ulsy**

**ID: 143-40-168**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in Multimedia and Creative Technology

Supervised By

**Dr. Shaikh Muhammad Allayear**

Associate Professor & Head of the Department

Department of MCT

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**November 2018**



## APPROVAL

This Project titled “**Falling**”, submitted by **Abrar Ulsy** to the Department of Multimedia and Creative Technology, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Multimedia and Creative Technology and approved as to its style and contents. The presentation has been held on **29<sup>th</sup> November 2018**.

## BOARD OF EXAMINERS

---

**Dr. Shaikh Muhammad Allayear**

**Associate Professor & Head**

Department of MCT

Faculty of Science & Information Technology

Daffodil International University

---

**Arif Ahmed**

**Adjunct Associate Professor**

Department of MCT

Faculty of Science & Information Technology

Daffodil International University

---

**Md. Samaun Hasan**

**Lecturer**

Department of MCT

Faculty of Science & Information Technology

Daffodil International University

---

**Professor Dr. Mohammad Zahidur Rahman**

**Professor**

Department of CSE

Jahangirnagar University

**Chairman**

**Internal Examiner**

**Internal Examiner**

**External Examiner**



## DECLARATION

I hereby declare that, this project has been done by me under the supervision of **Dr. Shaikh Muhammad Allayear, Associate Professor & Head of the Department, Department of MCT**, Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**

---

**Dr. Shaikh Muhammad Allayear**  
**Associate Professor & Head**  
Department of MCT  
Daffodil International University

**Submitted by:**

---

**Abrar Ulsy**  
ID: 143-40-168  
Department of MCT  
Daffodil International University



## ACKNOWLEDGEMENT

At first, we would like to express our gratitude to Almighty Allah for enabling us with his blessing to accomplish this final year project successfully. Making a complete and functional hyper-realistic 3rd person action game not only gives us a project to feature in our portfolio but also helps us to make a leap in terms of growing as a 3d artist. This project makes contributions to both Indie Game Development and AAA Game industries.

I'm really grateful and indebted to my project supervisor Dr. Shaikh Muhammad Allayear, Associate Professor and Head of the Department, Department of MCT Daffodil International University. His kind assistance, deep knowledge and a keen interest in the field of video games to carry out this massive project. His kind guidance, patience, wise advice, constructive and helpful criticism at all stage have made it possible and helped me grow through the development cycle of this project.

I would like to thank 3d artists and Unreal Engine developers around the world for the continuous contribution they are making to make the game development process easily accessible for indie developers like us.

I'm indebted to the faculty members and staff of the Multimedia and Creative Technology Department Daffodil International University for their support and aid through this long journey. I need to thank; Sawgat Audin, Anisuzzaman Khan, Aneek Mustafa Anwar, Rahmanur Rashid Hillol, Jabed Bhuiyan, Jesika Jahan Reha, Tanbeen Amin(Sweden) and Dianne Van Giersbergen(Netherlands)

Finally, I must pay my gratitude with due respect to my parents and family members for their constant support and patience all year round.



## ABSTRACT

The goal of this project was to create a hyper-realistic third-person action game based on a historic event from the Second Carnatic War. The game features high-quality models and textures of the characters and environment in order to be able to take the gamers through the historic Siege of Arcot led by Robert Clive. From September 23 to November 14 of 1751 British East India company forces consisting of 500 soldiers fought against 7420 soldiers of French East India Company and the Nawab of Arcot resulting in the victory of the British force that initiated Clive's supremacy and enabled English arms to achieve success after success in India later on. My objective was to create a game consisting of two different levels depicting the battle with 3d characters and environment assets made according to the standard pipeline used in AAA game development. The game can be released in pc version through Steam as well as Xbox One and PlayStation 4 versions in the marketplace. This paper breaks down the process of making the game from storyboarding to the final product. Processes of the 3d asset creation, sculpting, texturing and making them functional for the game and finally setting up everything in the game engine and optimizing the final game for hardware of the target consumers are described here. The paper takes a look at the AAA game titles from top developers leading the video game industry and how they inspire Indie Developers and 3d artists to be courageous enough for creating and shipping ambitious game projects with a small number of resources, low budget and open source tools. The significance of maintaining the quality of game assets within tight time constraint and budget is focused here. The paper also describes the importance of believable animation and enemy A.I along with cutting-edge beautiful visuals to make the game enjoyable and realistic for players. An overview of PBR workflow and its dominance in today's game industry is included here. The contribution of video games to make players learn about historical events through appealing visuals and storytelling is undeniable in the current time and that's the purpose of this project too.



## TABLE OF CONTENTS

| <b>CONTENTS</b>   | <b>PAGE</b> |
|---|-------------|
| Approval.....   | ii          |
| Declaration.....  | iii         |
| Acknowledgement.....  | iv          |
| Abstract.....   | v           |
| <br>  |             |
| <b>CHAPTER</b>  |             |
| <b>CHAPTER 1: INTRODUCTION.....</b>                           | <b>1-2</b>  |
| <br>  |             |
| <b>CHAPTER 2: PREVIOUS WORKS OF OTHERS.....</b>               | <b>3-6</b>  |
| 2.1 Third Person Game.....                                    | 3           |
| 2.2 Inspirations from games with historical settings.....     | 4           |
| 2.3 Open World setting in video games.....                    | 5           |
| 2.4 Unreal Engine and it's accessibility.....                 | 5-6         |
| <br>  |             |
| <b>CHAPTER 3: BREAKDOWN.....</b>                              | <b>7-46</b> |
| 3.1 Software used.....  | 7-8         |
| 3.1.1 Autodesk Maya.....                                      | 7           |
| 3.1.2 Zbrush.....   | 7           |
| 3.1.3 Marvelous Designer.....                                 | 7           |
| 3.1.4 Blender.....  | 7           |
| 3.1.5 3D Coat.....  | 7           |
| 3.1.6 Substance Painter.....                                  | 7-8         |
| 3.1.7 World Machine.....                                      | 8           |
| 3.1.8 Simplygon.....  | 8           |
| 3.1.9 Unreal Engine.....                                      | 8           |
| 3.2 Background Study.....                                     | 8           |
| 3.3 Steps of Development.....                                 | 9-36        |
| 3.3.1 Storyboarding.....                                      | 9           |
| 3.3.2 Concepting.....   | 9-14        |
| 3.3.3 Choosing the best workflow.....                         | 15          |
| 3.3.4 Sculpting & Modeling the high poly of Robert Clive..... | 15-17       |



|   |       |
|---|-------|
| 3.3.5 Retopology and low poly of Clive’s model.....                       | 17-19 |
| 3.3.6 Hair Creation of Robert Clive.....                                  | 19-20 |
| 3.3.7 Sculpting the high poly models of Soldiers and Charlie Marryat..... | 20-22 |
| 3.3.8 Making the Low Poly versions of Soldiers.....                       | 22-23 |
| 3.3.9 Making the haircards of Soldiers.....                               | 23    |
| 3.3.10 Modeling the Enfield Snider Rifle.....                             | 24    |
| 3.3.11 Modeling the Forts & Cannons.....                                  | 25-28 |
| 3.3.12 Packing the UV.....  | 29-32 |
| 3.3.13 Baking the maps.....   | 32-33 |
| 3.3.14 Texturing Process.....   | 34-39 |
| 3.3.16 Creating the terrain in World Machine.....                         | 40-42 |
| 3.3.17 Skinning & Rigging.....  | 42-44 |
| 3.3.18 Animation.....   | 45-46 |
| 3.3.19 Final Check-up.....  | 46    |

**CHAPTER 4: Developing the game inside Unreal Engine.....47-71**

|   |       |
|---|-------|
| 4.1 Creating the project and importing everything in Content Browser..... | 47    |
| 4.2 Setting up the animations for playable character.....                 | 48-49 |
| 4.3 Making the character playable.....                                    | 49-52 |
| 4.4 Setting up the environment.....                                       | 52-56 |
| 4.5 Lighting of the world.....  | 58-59 |
| 4.5.1 Sunlight.....   | 58    |
| 4.5.2 Exponential Height Fog.....   | 58    |
| 4.6 Post Process.....   | 59-62 |
| 4.6.1 Depth of Field.....   | 60    |
| 4.6.2 Bloom.....  | 60-61 |
| 4.6.3 Exposure.....   | 61    |
| 4.6.4 Lens Flare.....   | 61-62 |
| 4.7 Hardware Requirements & Optimization.....                             | 63-66 |
| 4.8 Soldier and Weapon Functionalities.....                               | 67-69 |
| 4.9 Game Hud.....   | 69-70 |
| 4.10 Challenges.....  | 70-71 |
| 4.11 Limitation.....  | 72    |



|   |              |
|---|--------------|
| <b>CHAPTER 5: Preview of The Game &amp; Assets.....</b> | <b>72-85</b> |
| 5.1 In Game Screenshots.....                            | 72-73        |
| 5.2 Characters Presentation.....                        | 74-80        |
| 5.2.1 Robert Clive.....                                 | 74-76        |
| 5.2.2 British Soldier.....                              | 77-78        |
| 5.2.3 French Soldier.....                               | 79-80        |
| 5.3 Weapon Presentation.....                            | 81-82        |
| 5.4 Forts Presentation.....                             | 83-85        |
| 5.4.1 Arcot Fort.....                                   | 83           |
| 5.4.1 Timari Fort.....                                  | 84-85        |
| 5.5 Terrain Presentation.....                           | 85           |
| <br>  |              |
| <b>CHAPTER 6: Discussion.....</b>                       | <b>86-88</b> |
| 6.1 Frustum Culling.....                                | 86-87        |
| 6.2 PBR Shaders.....                                    | 87           |
| 6.3 Mocap data for animation.....                       | 87-88        |
| 6.4 Dynamic Lighting.....                               | 88           |
| <br>  |              |
| <b>CHAPTER 7: Conclusion.....</b>                       | <b>89</b>    |
| <b>References.....</b>                                  | <b>90-91</b> |
| <b>Bibliography.....</b>                                | <b>92</b>    |





## FIGURES

|  |    |
|--|----|
| Figure 1: Total War Attila a strategy game based on the Late Antiquity period.....     | 3  |
| Figure 2: Assassin’s Creed III a third person game set during 18th century America...4 |    |
| Figure 3: Red Dead Redemption 2 Open world map.....                                    | 5  |
| Figure 4: PBR texturing metal-rough workflow.....                                      | 6  |
| Figure 5: British Empire Adventure Stories contains the novel.....                     | 9  |
| Figure 6: Rough Storyboard of Falling.....   | 9  |
| Figure 7: Concept Art of Arcot Battle from web.....                                    | 10 |
| Figure 8: Robert Clive painting from web.....  | 11 |
| Figure 9: Concept of a British Officer from Pinterest.....                             | 11 |
| Figure 10: Concept art of a British Soldier.....                                       | 12 |
| Figure 11: Concept art of a French Soldier.....  | 12 |
| Figure 12: Enfield Snider Reference photo.....   | 13 |
| Figure 13: Various fort references from the web.....                                   | 13 |
| Figure 14: Final reference used for making the Arcot Fort.....                         | 14 |
| Figure 15: Final reference of Timiri Fort.....   | 14 |
| Figure 16: Beginning stages of sculpting Robert Clive.....                             | 15 |
| Figure 17: Robert Clive high poly sculpt inside Zbrush.....                            | 16 |
| Figure 18: Simulated Costume inside Marvelous Designer.....                            | 16 |
| Figure 19: Initial Look of Clive’s Costume in zbrush.....                              | 17 |
| Figure 20: Robert Clive High Poly version in Zbrush BPR render.....                    | 17 |
| Figure 21: Required Facial Topology Flow.....  | 18 |
| Figure 22: Retopologized Low Poly of Clive(140k Triangles).....                        | 19 |
| Figure 23: Hair cards texture image.....   | 20 |
| Figure 24: Robert Clive’s Hair cards in Blender.....                                   | 20 |
| Figure 25: Bust sculpt of British Soldier, Charlie and French Soldier.....             | 21 |
| Figure 26: High Poly Sculpt of the British Soldier.....                                | 22 |
| Figure 27: High Poly of the French Soldier.....  | 22 |
| Figure 28: Low Poly of French Soldier’s model.....                                     | 23 |
| Figure 29: Texture for French Soldier hair cards.....                                  | 23 |
| Figure 30: Hair cards Placed with texture on French Soldier’s head.....                | 24 |
| Figure 31: High poly mesh of Enfield Snider Rifle.....                                 | 24 |



|   |    |
|---|----|
| Figure 32: Low Poly version of the Enfield Snider (3.9k Triangles).....         | 25 |
| Figure 33: Block out of Timari Fort.....  | 25 |
| Figure 34: Block out of Arcot Fort.....   | 25 |
| Figure 35: High poly sculpt of Timari Fort Modules.....                         | 26 |
| Figure 36: High Poly Sculpt of Arcot Fort Modules.....                          | 27 |
| Figure 37: Pieces of Beekeeper house and one of the torn gates.....             | 27 |
| Figure 38: Pieces of destroyed building structures.....                         | 28 |
| Figure 39: 36 Pounder Cannon based on trophy version.....                       | 28 |
| Figure 40: 36 Pounder Cannon based on the regular version.....                  | 29 |
| Figure 41: Swivel Gun High Poly mesh.....                                       | 29 |
| Figure 42: Face, Hands and eyes in one UV set for Robert Clive.....             | 30 |
| Figure 43: One single UV set for the Coat of soldiers.....                      | 30 |
| Figure 44: French Soldier UV packing inside 3D Coat.....                        | 31 |
| Figure 45: UV set packing of Enfield Snider in 3D Coat.....                     | 31 |
| Figure 46: Fort Side Gate Group UV set of Timari Fort.....                      | 32 |
| Figure 47: Round Tower UV set of Arcot Fort.....                                | 32 |
| Figure 48: Texture Map Baking options in Substance Painter.....                 | 33 |
| Figure 49: ID Map of Enfield Snider baked in Maya.....                          | 34 |
| Figure 50: Texturing Clive's Model inside Substance Painter.....                | 34 |
| Figure 51: Look development of Clive in Substance using it's iray renderer..... | 35 |
| Figure 52: Look Development of British Soldier.....                             | 35 |
| Figure 53: Layers put to paint and use maps for texturing the rifle.....        | 36 |
| Figure 54: Smart Masks in Substance Painter.....                                | 36 |
| Figure 55: Smart Material library inside Substance Painter.....                 | 37 |
| Figure 56: Painting drips and moss in Substance Painter.....                    | 37 |
| Figure 57: Modules of Timari fort in Substance Iray renderer.....               | 38 |
| Figure 58: Modules of Arcot fort in Substance Iray renderer.....                | 38 |
| Figure 59: Realtime characters of Falling with textures.....                    | 39 |
| Figure 60: Realtime model of Enfield Snider Rifle with textures.....            | 39 |
| Figure 61: Realtime models of Arcot and Timari forts with textures.....         | 39 |
| Figure 62: Image of LiDAR data from Atkins Global.....                          | 40 |
| Figure 63: Nodes used inside World Machine to generate my terrains.....         | 41 |
| Figure 64: Landscape Material network inside Unreal Engine.....                 | 41 |



|   |    |
|---|----|
| Figure 65: Arcot Terrain overview in Unreal Engine.....                   | 42 |
| Figure 66: Timari Terrain Overview in Unreal Engine.....                  | 42 |
| Figure 67: Placing Bones of the skeleton.....                             | 43 |
| Figure 68: Checking out the deformations.....                             | 43 |
| Figure 69: Painting weights with Maya’s weight painting tool.....         | 44 |
| Figure 70: Setting up the control Rig.....                                | 44 |
| Figure 71: Mixamo animation library.....                                  | 45 |
| Figure 72: Editing the animation montage.....                             | 46 |
| Figure 73: Autodesk Maya Time Editor.....                                 | 46 |
| Figure 74: Content Browser of Falling.....                                | 47 |
| Figure 75: Previewing Animation in Unreal Engine.....                     | 48 |
| Figure 76: List of animations.....  | 48 |
| Figure 77: Blend Space for Idle, Walk and Run.....                        | 49 |
| Figure 78: Animation Graph of the Brit Soldier Blueprint.....             | 50 |
| Figure 79: British Soldier Locomotion.....                                | 50 |
| Figure 80: Firing Animation montage.....                                  | 51 |
| Figure 81: Aim Poses.....   | 51 |
| Figure 82: Event graph of British Soldier Blueprint.....                  | 52 |
| Figure 83: Arcot Fort from different angles.....                          | 52 |
| Figure 84: Timari Fort Placement.....                                     | 53 |
| Figure 85: Checking out the Terrain with colors.....                      | 53 |
| Figure 86: Populating terrain with grass from the landscape material..... | 54 |
| Figure 87: Field Grass Material.....                                      | 54 |
| Figure 88: Field Grass Diffuse.....                                       | 54 |
| Figure 89: Field Grass Output.....  | 55 |
| Figure 90: Foliage tool contents.....                                     | 55 |
| Figure 91: Billboard material of ScotsPine.....                           | 55 |
| Figure 92: BogMyrtle Output.....  | 56 |
| Figure 93: Options that can be tweaked for each foliage.....              | 56 |
| Figure 94: Procedural Foliage Simulation tool.....                        | 57 |
| Figure 95: Arcot Terrain populated with grass & foliages.....             | 57 |
| Figure 96: Directional Light settings.....                                | 58 |
| Figure 97: Exponential Height Fog parameters.....                         | 58 |



|   |    |
|---|----|
| Figure 98: Arcot Terrain with Sun and Exponential Height Fog enabled..... | 59 |
| Figure 99: Post Process Volume enabled with Infinite Extent.....          | 60 |
| Figure 100: Depth of Field parameters.....                                | 60 |
| Figure 101: Bloom Parameters.....   | 61 |
| Figure 102: Exposure Parameters.....                                      | 61 |
| Figure 103: Lens Flare parameters.....                                    | 62 |
| Figure 104: Light Propagation Volume parameters.....                      | 62 |
| Figure 105: Timari Terrain with Post Process features enabled.....        | 63 |
| Figure 106: Box colliders for Arcot Fort.....                             | 65 |
| Figure 107: British Soldier Blueprint.....                                | 67 |
| Figure 108: Sparks Particle controls.....                                 | 68 |
| Figure 109: Smoke Particle controls.....                                  | 68 |
| Figure 110: Rifle Smoke and Bullet Trace in Blueprint.....                | 69 |
| Figure 111: GameHUD widget designer window.....                           | 69 |
| Figure 112: GameHUD widget blueprint.....                                 | 70 |
| Figure 113: In Game Screenshot of Arcot Fort Front.....                   | 72 |
| Figure 114: In Game Screenshot of Arcot Terrain Trees.....                | 72 |
| Figure 115: In Game Screenshot of Timari Fort Front.....                  | 73 |
| Figure 116: In Game Screenshot of Timari Terrain grass & foliage.....     | 73 |
| Figure 117: Robert Clive presentation Marmoset.....                       | 74 |
| Figure 118: Robert Clive front.....                                       | 74 |
| Figure 119: Robert Clive left.....  | 75 |
| Figure 120: Robert Clive right.....                                       | 75 |
| Figure 121: Robert Clive back.....  | 76 |
| Figure 122: Robert Clive Posed.....                                       | 76 |
| Figure 123: British Soldier Posed.....                                    | 77 |
| Figure 124: British Soldier Front.....                                    | 77 |
| Figure 125: British Soldier right.....                                    | 78 |
| Figure 126: British Soldier left.....                                     | 78 |
| Figure 127: French Soldier Presentation.....                              | 79 |
| Figure 128: French Soldier Left.....                                      | 79 |
| Figure 129: French Soldier right.....                                     | 80 |
| Figure 130: French Soldier front.....                                     | 80 |



|   |    |
|---|----|
| Figure 131: Enfield Snider front.....   | 81 |
| Figure 132: Enfield Snider back.....  | 81 |
| Figure 133: Enfield Snider butt.....  | 82 |
| Figure 134: Enfield Snider muzzle.....  | 82 |
| Figure 135: Arcot Fort wide shot.....   | 83 |
| Figure 136: Arcot Fort mid shot.....  | 83 |
| Figure 137: Arcot Fort rear shot.....   | 83 |
| Figure 138: Timari Fort wide shot.....  | 84 |
| Figure 139: Timari Fort side shot.....  | 84 |
| Figure 140: Timari Fort rear long shot.....                                     | 84 |
| Figure 141: Timari Fort overview.....   | 85 |
| Figure 142: Timari Terrain view from the ground.....                            | 85 |
| Figure 143: Arcot Terrain view from the ground.....                             | 85 |
| Figure 144: Without frustum culling (left) and with frustum culling(right)..... | 87 |
| Figure 145: Frustum culling of Horizon Zero Dawn.....                           | 87 |



## **CHAPTER 1**

### **Introduction**

Video Game industry is a huge part of the entire entertainment industry worldwide. People nowadays play video games not only to have fun or enjoy multiplayer competitive gaming but also to experience a good story through immersion in the world of the game itself. The increasing quality of video game graphics with both realistic and stylized games make the developers push for both in-game content and the story to provide an overall cinematic experience for players. The collective output from a strong storyline, proper art style according to the story, high-quality 3d assets of characters, hard surface, environment and props, realistic animations, fun and responsive gameplay and the advancement of modern game engines play the role to turn a game into a complete experience. The market is getting bigger and developers try to innovate and push for improvement in every step to stay ahead of the competition. The games that feature important historic events attract the attention of most gamers due to having a strong and true storyline.

Modern games that feature history through advance gameplay mechanics and realistic graphics generally receive a very good critical response and market value. Assassin's Creed series, Ryse Son of Rome, Total War series, Kingdom Come Deliverance are some of the recent triple-A titles in this genre. In my project, I tried to look at a part of history that is untouched by any game developer to this day. The Siege of Arcot by Robert Clive's British East India company force which was a part of the Second Carnatic War at India was the first battle that cemented Clive as a force to fear for the Indian Nawabs and French East India Company. This Battle also proved pivotal to the British domination at India afterwards. The goal is to deliver an experience from that particular event for the players while also showcasing my ability to make high-quality game ready 3d characters and assets.

A game must have an overall art style that completely aligns with the story. So that the environment and characters carry the fragment of the timeline and events that are being



shown. The art style dictates how the models will be made, the scene compositions, the colour palette, props, setting and even the animations. As my game takes place in the 17th century to support its historical setting I looked at the artworks of the British soldiers and India from that time to decide how my game will look, what the setting will be, how the characters and models will be made and how I can create the atmosphere of Arcot from that time. The stylized look was not the best option to tell a story about this event so I chose a hyper-realistic look for my game.

During my course of Compression Technique where we are taught about the complete pipeline from Maya and Zbrush to substance painter in order to create high-quality game ready assets. I grew a lot like a 3d artist and I was able to practice and develop to be efficient enough in creating game ready models to be used in video games. The goal of creating hyper-realistic models is not to make them look like photos but to make them realistic in accordance with the art style of the overall game. The models can be made by taking real-world references and inspiration. Hyper-realistic models are the perfect fit for my game to make players experience the battle of Arcot.

The aim of this project is to develop a game with high-quality visuals and gameplay that tells how Robert Clive and his force fought the French to take over Arcot. The players can experience an important part of the history of our subcontinent and they will get to know Robert Clive in a different way while they have fun through the third-person open world shooter experience. This indie game can be published in the steam store and other marketplaces.

In the following chapters, I extensively described the method, the pipeline and workflow, the software packages that have been used, the inspirations as well as some investigation and discussion about modern games and their technical aspects.

## CHAPTER 2

### Previous work of others in the same area

#### 2.1 Third Person Game

Third person action games generally provide the player with the playable character visible on screen while playing. These games mostly are of shooting. Some of them feature sword fighting and fighting with different kinds of weapons. Even though there were games of this genre back in the 1980s and 90s it became widely popular and mainstream since 2000. A lot of games that want to tell the gamers compact and intense stories tend to take the path of third-person action. Even though many history based games are RTS(Real Time Strategy) and Turn Based Strategy games modern developers have focused on the third person action genre mostly to make players involved with one or two key characters they can play with. This helps the gamers to feel more connected to the story and character development while becoming more personal with the playable character.



Figure 1: Total War Attila a strategy game based on the Late Antiquity period



## 2.2 Inspirations from games with historical settings

If we take a look at the likes of Assassin's Creed series, Call of Duty series, Battlefield series, Ryse Son of Rome, Medal of Honor series we can have great examples of intense storytelling, real events, strong historical background, high-quality visuals and audio to take the gamer through the story and most importantly a complete immersive experience in an important historical setting.



Figure 2: Assassin's Creed III a third person game set during 18th century America

As I took inspirations from these aforementioned games I analyzed and studied them to find out the perfect method and workflow in order to achieve my goal. Sculpting the high poly models in zbrush to bake the details into low poly model, the physically based rendering workflow for textures, using motion capture data for animating the characters and making the game in Unreal Engine for it's advanced lighting to achieve the kind of look I was targeting were the key steps I put in my workflow after studying the top video game titles.

### 2.3 Open World setting in video games

The sandbox or open world setting is increasingly popular among the developers and gamers these days as this setting allows the players to explore the game more, have more content along with the key objectives of the game. As my game takes place around the area of Arcot and Timiri I took the opportunity to create a beautiful world that provides the sandbox experience for players to roam as well as play the main missions. It offers versatility in gameplay unlike linear level based games that is rather one dimensional in terms of focusing only on the main missions and key objectives.



Figure 3: Red Dead Redemption 2 Open world map

### 2.4 Unreal Engine and it's accessibility

Optimizing the game for most pc users and make it affordable and available for mid-range pc users is a key priority of the pc version. The Unreal Engine provides fantastic tools for optimizing and profiling the game with detailed models and high-resolution textures as well as a huge world filled with assets and foliage. Most modern engines



made the optimization easier so developers can push for more high-quality visuals while maintaining the system requirements reachable for most pc users. For the Xbox One and PlayStation 4 versions optimization becomes easier when the pc version is ready. Since the consoles come with their stock configuration it is easier to do profiling and making the game playable at desired frame rate than in the pc version.

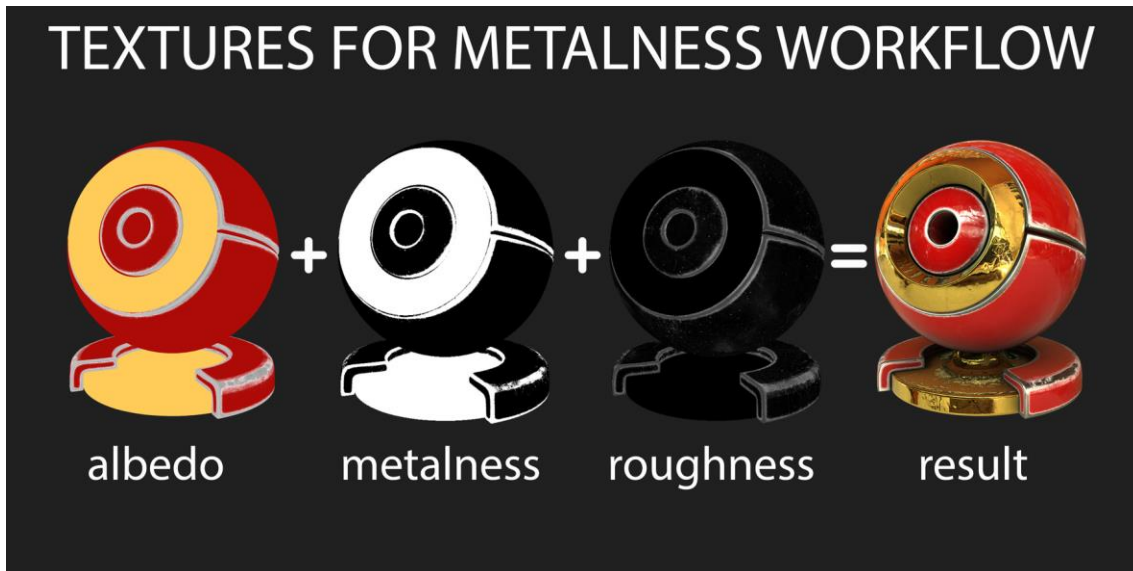


Figure 4: PBR texturing metal-rough workflow



## CHAPTER 3

### Breakdown

#### 3.1 Software used:

The process of developing a 3rd person action game is huge and intensive. It required a total of 9 software packages including Unreal Engine 4.

3.1.1 Autodesk Maya: Autodesk Maya or Maya is a 3d Computer Graphics application created by Alias Systems Corporation and later owned by Autodesk. It is used to create 3d products, 3d animation shorts, video games, film, visual effects and 3d models of any kind.

3.1.2 Zbrush: Zbrush is a digital sculpting software that combines 3D and 2.5D modelling. It has poly modelling tools but its best use is the brushes and tools for sculpting. This software is widely popular for creating highly detailed models for 3d printing, games and movies.

3.1.3 Marvelous Designer: Marvelous Designer is a 3D Cloth simulation software that allows creating cloth with realistic draping and simulation at ease really fast. It can virtually replicate fabric textures and physical properties to last button, fold and accessories. It was used to create the costume of Robert Clive.

3.1.4 Blender: Blender is an open source 3D Application. I have used this to create the hair cards for my characters.

3.1.5 3D Coat: 3D Coat is a 3D application from Pilgway designed to create organic and hard surface 3D Models. The tools to sculpt, add polygonal topology, create UV maps, texturing and rendering are available here. In my project, I used 3D Coat's advanced and simple UV Unwrapping and packing tools to get the best UV for my models. I also used 3D Coat to clean up the artefacts from the normal and ambient occlusion maps of my models.

3.1.6 Substance Painter: Substance painter is a 3d texture painting software that allows users to paint and texture 3d assets that have specific UV layouts. This software uses PBR workflow of texturing. It also has a real-time rendering feature in its viewport for working on the look development of assets. The uses of smart masks, smart materials,



brushes, alphas, procedurals, generators, filters of this software really enhanced the quality of my assets.

3.1.7 World Machine: World Machine is a powerful terrain creation software. It has tools and a node-based workflow to create terrains procedurally. The software allows users to export height and splat maps in very high resolutions to the game engines.

3.1.8 Simplygon: Simplygon is a 3D application used for optimizing 3D Models through remeshing and reducing polygons. This is a very useful tool for creating LODs to be used in the game.

3.1.9 Unreal Engine: Unreal Engine is a game engine developed by Epic Games. I've used this engine to bring all the assets together, set up the scene and environment, set up the animations, game mechanics, enemy AI and to do all the other tasks in order to create a final build of my game. Although the engine was primarily used for creating first-person shooting games back in the early 2000s later it has contributed to the development of games across a variety of genres. The code is written in C++ and because of its advanced and easy to adapt features and tools it is being widely used by game developers worldwide. The most recent version is Unreal Engine 4.

## 3.2 Background Study

Before taking on the project I researched on the Siege of Arcot event to have a solid understanding of the battle that took place, its background and its outcomes. So I studied "With Clive in India" by GA Henty from British Empire Adventure Stories novel collection. GA Henty's novel provided me with an in-depth idea of Clive and his force's journey to conquer Arcot. I had to research the artworks portraying the 17th century time of India to understand how the British and French soldiers would look. After that, I decided how the overall art style and setting of the game will be, how the story will progress with the game's objectives and what kind of gameplay I will try to produce. Following the research, I initiated the project and started working on the storyboard.

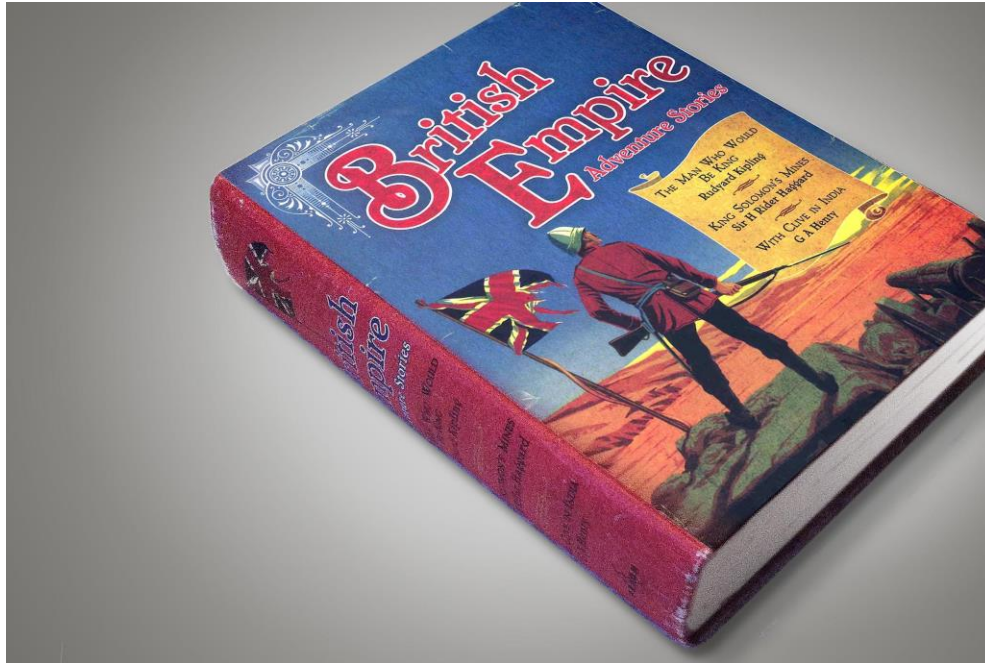


Figure 5: British Empire Adventure Stories contains the novel

### 3.3 Steps of Development:

3.3.1 Storyboarding: Ready the storyboard including gameplay sections and how the objectives will be laid out based on the novel's "Siege of Arcot" Chapter. The storyboard includes how the story will be told along with the gameplay.



Figure 6: Rough Storyboard of Falling

3.3.2 Concepting: Based on the information from the book and arts found in Pinterest and web the concepts of characters, environment and weapon were done. The gameplay mechanics were planned based on the description of the battle in the novel.

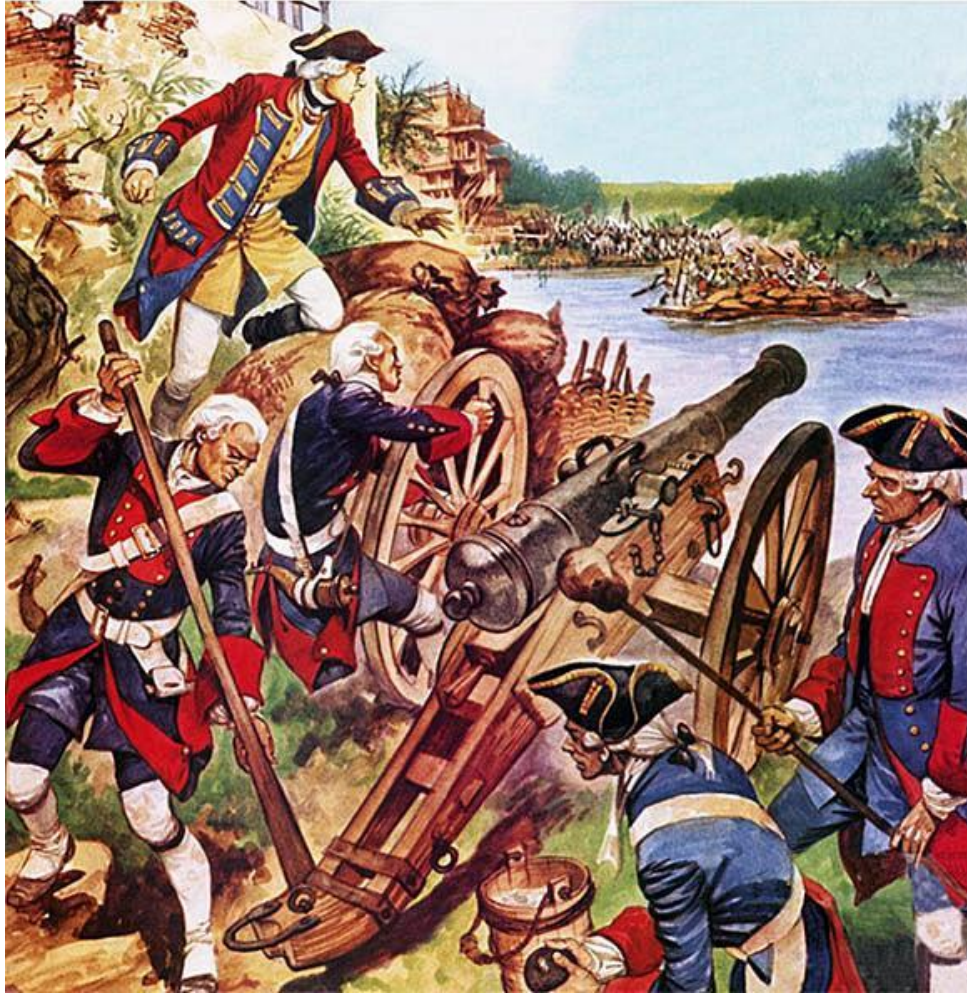


Figure 7: Concept Art of Arcot Battle from web

In order to figure out how I want Robert Clive to look I needed to look at his available portraits from internet. I needed a much younger Clive and unfortunately there's no portrait of him to be found from that period. So I needed to collect concepts of officers and their costumes and imagine a young Clive wearing those. I decided to build up the young Clive's bust based on the available portraits and my assumption. So I had a lot of sculpting to do for iterating the various passes of his face before I could achieve the final look.



Figure 8: Robert Clive painting from web



Figure 9: Concept of a British Officer from Pinterest





I went through concepts of British East India Company soldiers and their costumes from different periods between 1700s and 1800s. The signature red coat, the East India Company logo, side bag, the cap and boots were the most common things to be found in all of them.



Figure 10: Concept art of a British Soldier

The French soldiers had almost similar costume minus the red color in their coats. Instead of red they had blue coats that resembles the color from their flag. With the French force I decided to add a backpack instead of a side bag. This gave me variety as well as some distinguishable feature for them.



Figure 11: Concept art of a French Soldier

For weapons I've selected a number of weapons to be used across the game including Enfield Snider Rifle, Rail Gun, Pistol, 36 Pounder and 18 Pounder cannons and Swivel Gun all based on the novel and how it has described the arsenal of both British and French fronts. Fortunately there are plenty of references available in the web for the weapons used during that period.



Figure 12: Enfield Snider Reference photo

The two forts I wanted to show in the game don't exist anymore. It's also not possible to find any drawing or photo that shows what they looked like. So I had to come up with random references and ideas to figure out what kind of forts would suit my environment as well as the story and it's timeline.



Figure 13: Various fort references from the web



Figure 14: Final reference used for making the Arcot Fort



Figure 15: Final reference of Timiri Fort



3.3.3 Choosing the best workflow: With the experience of more than two years of 3D Modeling the most suitable and standard PBR workflow was chosen for creating PBR enabled models. Since it's a game developed for PC and consoles Unreal Engine was picked because it's the most user-friendly game engine with lots of resources and materials available on the web. For the animation of characters, the rigging and skinning in Maya was a must but keyframing every single animation was not needed due to the availability of Mixamo's motion-capture animation library. Due to the time constraint and the requirement of quality animations, it wasn't possible to manually keyframe so many animations for character movements. Based on the storyboard and concepts the overall art style of the game was decided and after thorough research, I was able to decide how to proceed for modelling, animation and putting everything inside game engine to prepare the game.

3.3.4 Sculpting & Modeling the high poly of Robert Clive: All of the modelling works were done with Maya and Zbrush. Zbrush was mostly used to sculpt the high poly versions of characters, assets, props and weapons. Maya was useful to create the block out, the retopology work and finally creating the game ready low poly versions.

The beginning stages of sculpting Robert Clive was figuring out how his bust will look. I needed to make sure he looked young enough to be recognized from that period. After going through some passes during sculpting I was able to reach a point where the bust started to resemble Robert Clive in some ways.

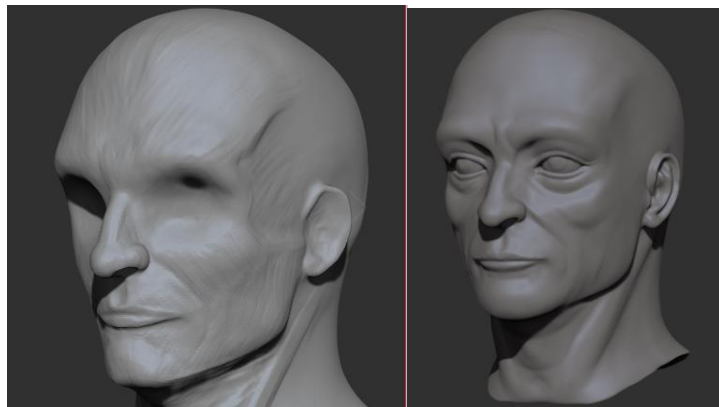


Figure 16: Beginning stages of sculpting Robert Clive

After the bust was done I was able to attach it with the rest of the body. The body covers a strong and stout figure of our young Clive.

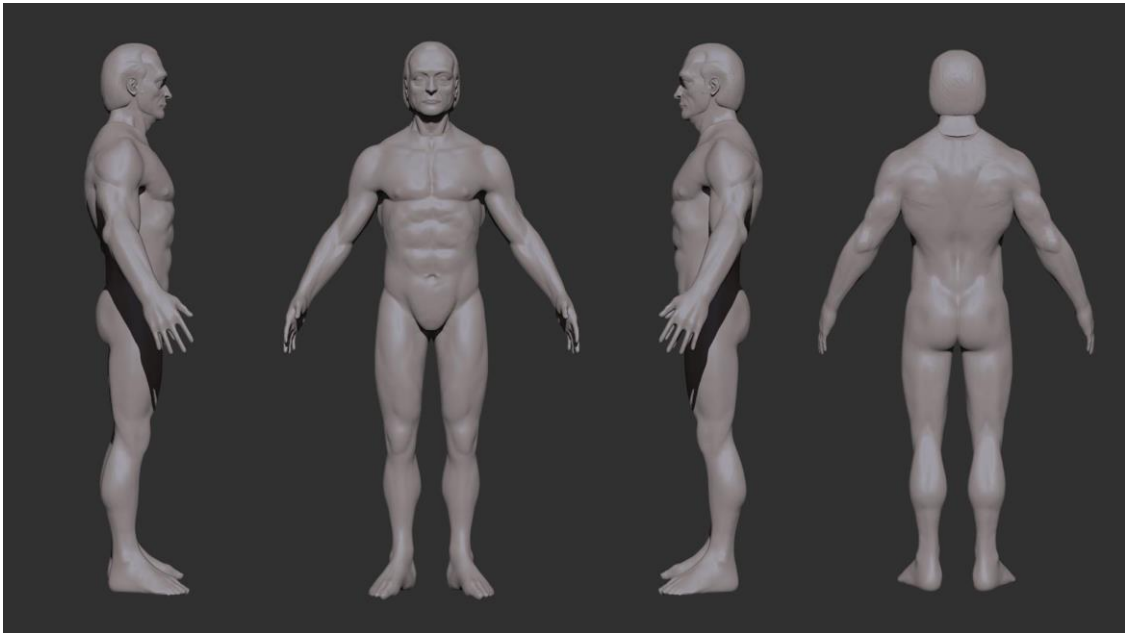


Figure 17: Robert Clive high poly sculpt inside Zbrush

The decimated version of Clive's high poly body was taken to Marvelous Designer to exploit its fantastic cloth simulation tools to simulate Clive's shirt, pant, coat and hat. Marvelous gave me a good base by simulating the cloth according to the properties of material (e.g. polyester, cotton, silk etc) I set for each part of the costume.



Figure 18: Simulated Costume inside Marvelous Designer

The props including the belt, buttons, side bag and leather shoes are made in Maya and further developed through sculpting inside zbrush. The retopo of marvelous meshes from

Maya were taken into zbrush and main marvelous meshes details were projected into them while I had to subdivide them multiple times for getting clean projection. Then they were sculpted further to get better wrinkles, position, stitches and more details. Finally all the high poly models came alive together inside of zbrush and I had an idea how Robert Clive is going to look in the game.



Figure 19: Initial Look of Clive's Costume in zbrush



Figure 20: Robert Clive High Poly version in Zbrush BPR render

3.3.5 Retopology and low poly of Clive's model: The retopology process of the entire character, it's costume and props were pretty straightforward. I had to create the low poly version of everything inside maya with it's modeling tools and brilliant Quad Draw which works as a fantastic retopology tool for me. Thankfully I could create the final low



poly versions of props from the blockouts to save a good chunk of time. The marvelous meshes had to be retopologized before taking into zbrush for further sculpting. Those were used as the final low poly version for costumes. The highest subdivision version of them with all details were to be used as final high poly mesh. I had to do a clean retopology of the hands and face from the high poly counterparts using quad draw by following the proper topology flow in order to make my character skinning and animation friendly. The low poly of costumes also needed clean topo flow and extra edge loops at areas where the character will deform during animations.

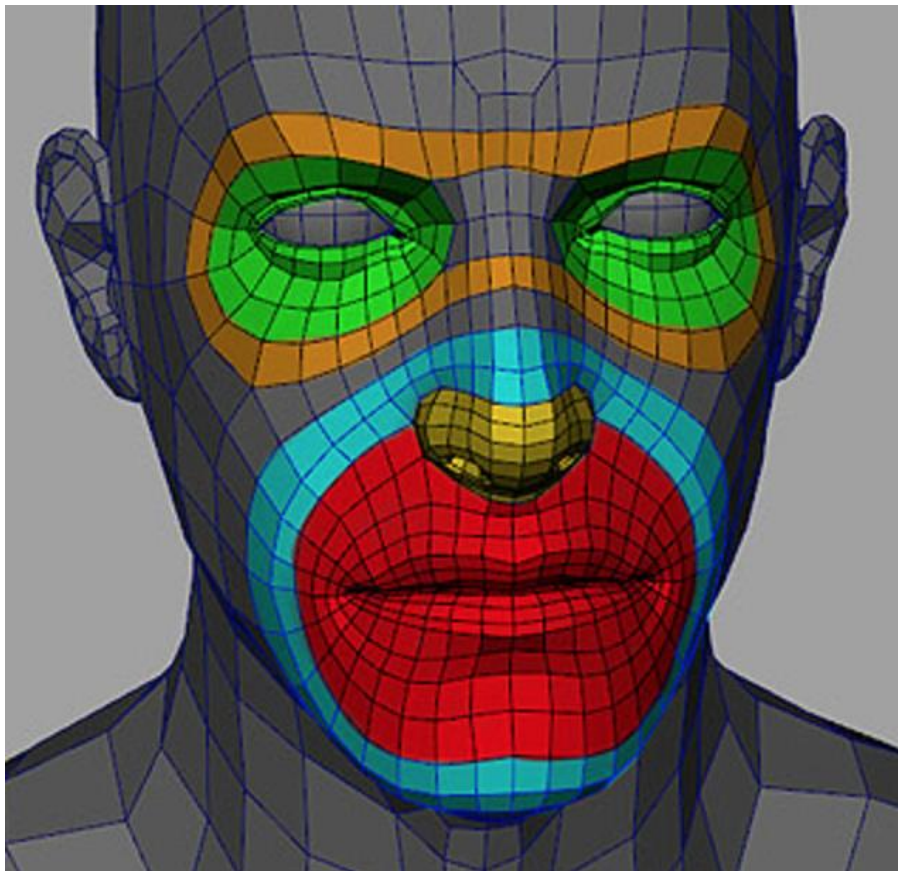


Figure 21: Required Facial Topology Flow

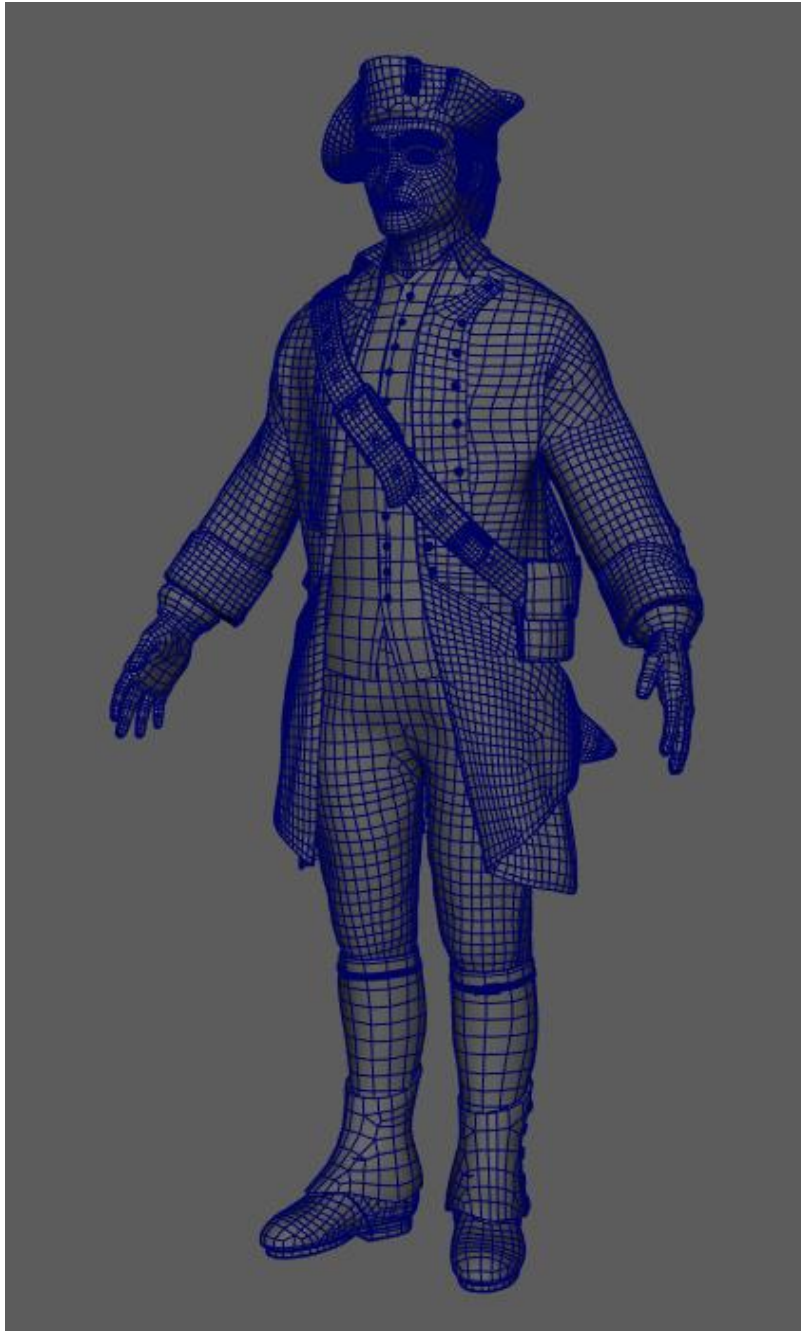


Figure 22: Retopologized Low Poly of Clive(140k Triangles)

3.3.6 Hair Creation of Robert Clive: Robert Clive's hairwork was entirely done inside blender. With it's particle hair first rendered for placing the haircards according to their UV. Then I placed all the haircards on Clive's head made by bezier curve. Every single haircard has the exact same UV and the UV contains texture for four different type of



hair cards. The hair cards are placed on the head in a way we can have variety and the hairstyle looks believable.



Figure 23: Hair cards texture image

Blender Cycles Engine was really useful for me to do the look development of the hair. I switched between different colors to get the final look of Clive's hair. Realtime viewport rendering of cycles helped me to iterate really fast and get quick results.



Figure 24: Robert Clive's Hair cards in Blender

3.3.7 Sculpting the high poly models of Soldiers and Charlie Marryat: A Base Mesh of a human body was taken from makehuman and I've used this same base mesh to sculpt three different busts of French Soldier, British Soldier and Robert Clive's companion



Charlie Marryat. The hands for British Soldier and French Soldier are the same as Clive's. Only for Charlie I sculpted rest of the body including hands from the base mesh. All the sculpting work has been done inside Zbrush.

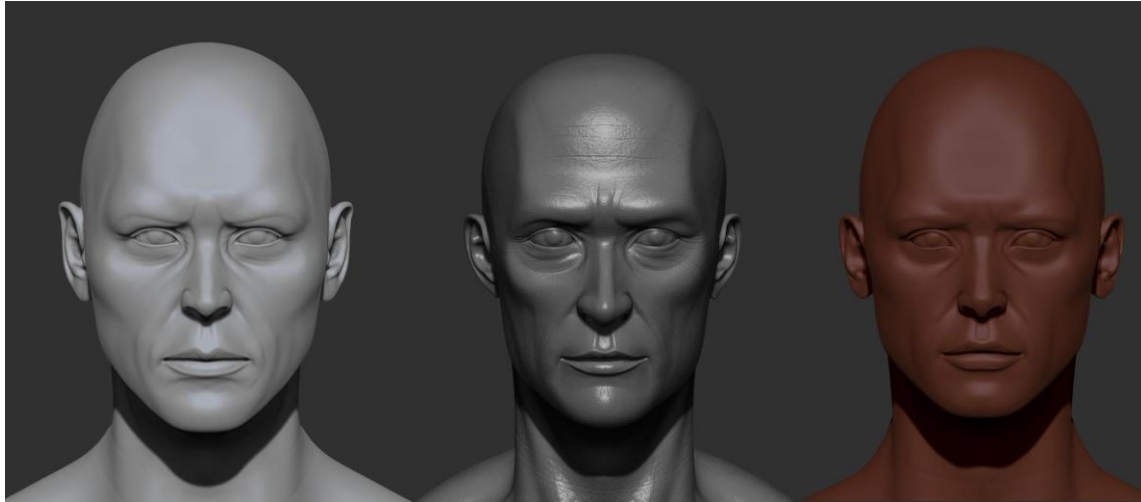


Figure 25: Bust sculpt of British Soldier, Charlie and French Soldier

The costumes and props for my soldiers were made using maya and sculpted in zbrush. In this case I decided not to go to Marvelous designer and stick to zbrush-maya hybrid workflow. The base meshes and blockouts were done in Maya and then I brought them in Zbrush to sculpt all the details. For making the base meshes of costumes first I extracted basic shapes of clothing from the base mesh in zbrush. Then I took them to Maya, did the retopology to figure out low poly versions of coat, shirt, pant and shoes. Then brought them in zbrush, raised subdivision level as I sculpted and put details. For the bag, belt, buttons and hat I followed more or less the same workflow. I felt I needed a bit more research and familiarity with Marvelous before I can use it for making all sorts of costumes. So I went on with the classic modeling and sculpting workflow for the costume of soldiers. In future I would like to be able to work more frequently with Marvelous in order to make some High Quality costumes for Charlie Marryat.



Figure 26: High Poly Sculpt of the British Soldier



Figure 27: High Poly of the French Soldier

3.3.8 Making the Low Poly versions of Soldiers: During the blackout phase all the models were already created in lower polycount. I had to edit them with quad draw to make sure they fit properly with their high poly counterparts. The Makehuman base mesh that I sculpted on has amazing topology flow so I could get a very functional face from

the lowest subdivision of my sculpt for each of the soldiers. All of this were put together inside Maya and I got clean low poly version of soldiers. Both of them contain 140k triangles each including haircards.

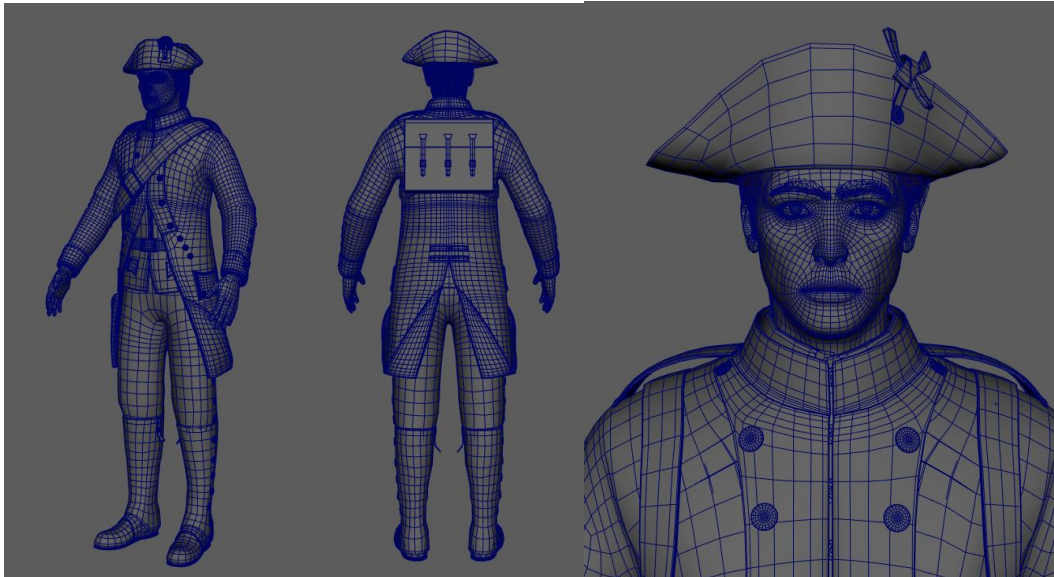


Figure 28: Low Poly of French Soldier's model

3.3.9 Making the haircards of Soldiers: Similarly as Robert Clive's I followed Blender's bezier curve workflow to create the haircards and rendered hair particles by placing them on viewport according to the UV of haircards. When I plug the diffuse and alpha of the rendered texture with placed haircards on character's head they come to life.



Figure 29: Texture for French Soldier hair cards



Figure 30: Hair cards Placed with texture on French Soldier's head

3.3.10 Modeling the Enfield Snider Rifle: Making the enfield snider was about going back and forth between Maya and Zbrush. The blockout sculpt was done in zbrush. Then it was taken into maya to retopologize and give it the hard look. It was brought into zbrush again, subdivided and further details were sculpted. The retopology version in maya was used to build the final low poly mesh. The Low Poly mesh was comprised of 3.9k triangles.

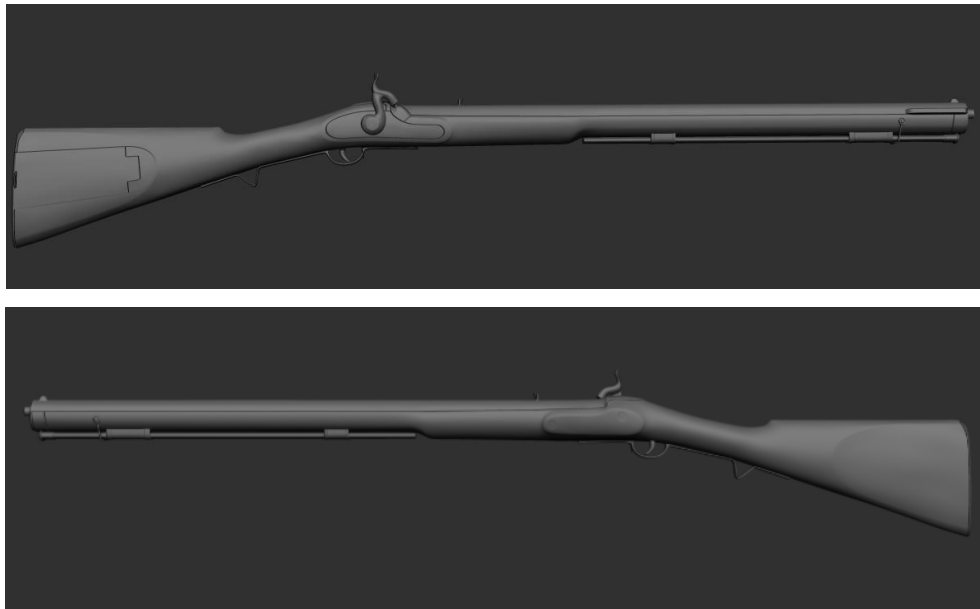


Figure 31: High poly mesh of Enfield Snider Rifle

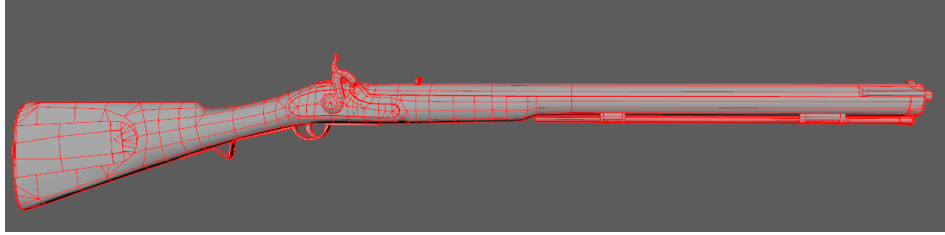


Figure 32: Low Poly version of the Enfield Snider (3.9k Triangles)

3.3.11 Modeling the Forts & Cannons: The modeling of my forts were the lengthiest process in this pipeline. First I had to do the complete block out according to approximate measurements I could come up with the generic mannequin model from Unreal Engine as my reference for a human size. Then I had to figure out how many modular pieces I was going to put based on my reference concept. So I can build each modular piece on its own and then put all of them together with the full block out model as my base.

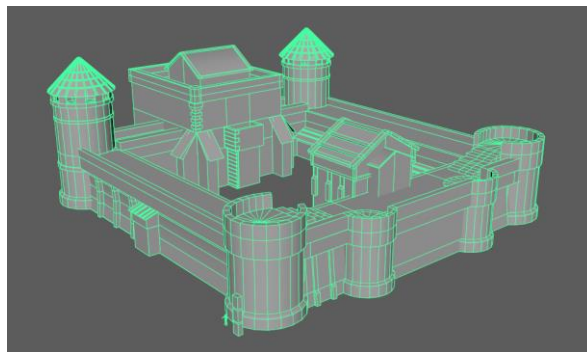


Figure 33: Block out of Timari Fort

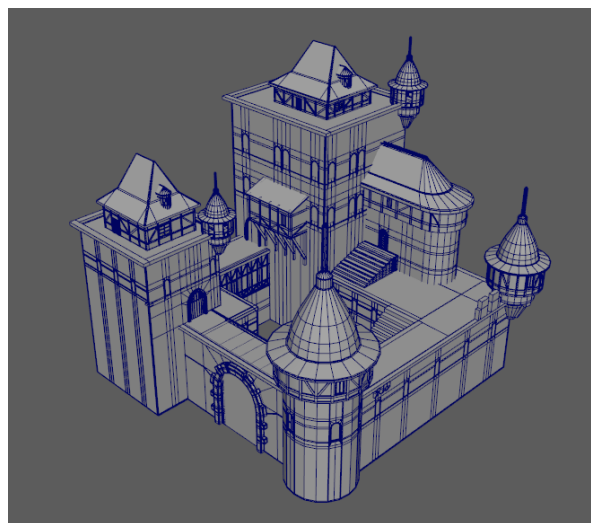


Figure 34: Block out of Arcot Fort

After the blockout was done and I knew how much modular pieces I need and where to put them it was time to work on each one of them. The pieces that will be repeated would be needing to build only once. That is the beauty of using modular workflow. Each low poly mesh of the pieces was made inside maya and then taken inside Zbrush to subdivide to higher polycounts for sculpting. I have used different brushes for sculpting the pieces and make them look torn, old and natural. I was aiming to tell a story through my environment so I needed a lot of wear and tear as well as the beaten up look in my assets. Zbrush has all the tools I needed to sculpt them and bring my desired look.

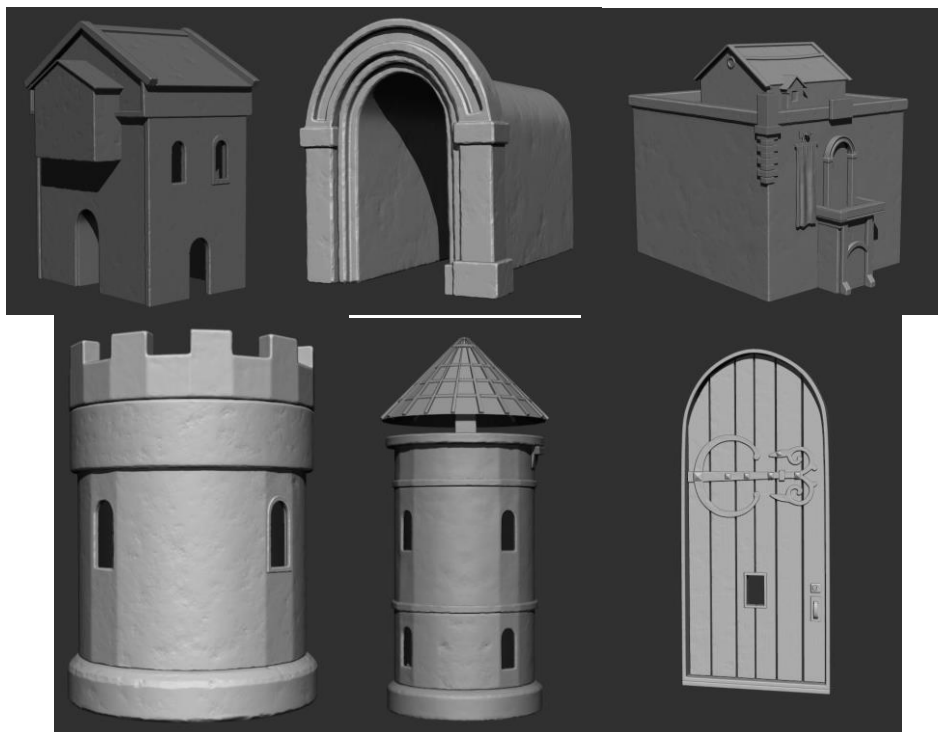


Figure 35: High poly sculpt of Timari Fort Modules

As well as a lot of hand sculpting I've used the noise tool of zbrush to its fullest in order to achieve the natural look on my surfaces. Generally using the noise directly sometimes ruin the volume of the mesh. So I've used mask with noise feature to paint the details with displacement brush or sometimes used the inflate feature from deformation tab for bringing the beaten up details on the surfaces. Since I already built the low poly meshes during the blockout stages of my modules I didn't need to do any retopology at all.

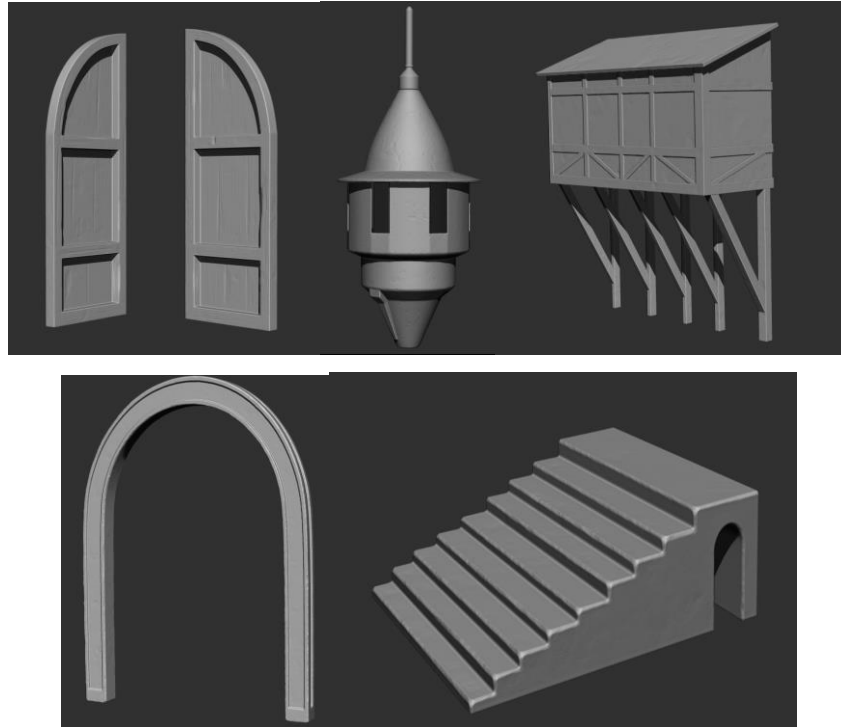


Figure 36: High Poly Sculpt of Arcot Fort Modules

After getting done with the fort modules it was time to sculpt the other environment assets in similar way. I built variety of assets to fill my terrain with. Since I've got two big open terrains I have a lot of room to fill up with structures, objects, settlements. After the forts I made a beekeeper house set, a lumberjack house set, two different types of torn gates and some destroyed building pieces. These assets could be scattered and reused to fill up different parts of terrain according to my overall world composition based on the storyline and game settings.

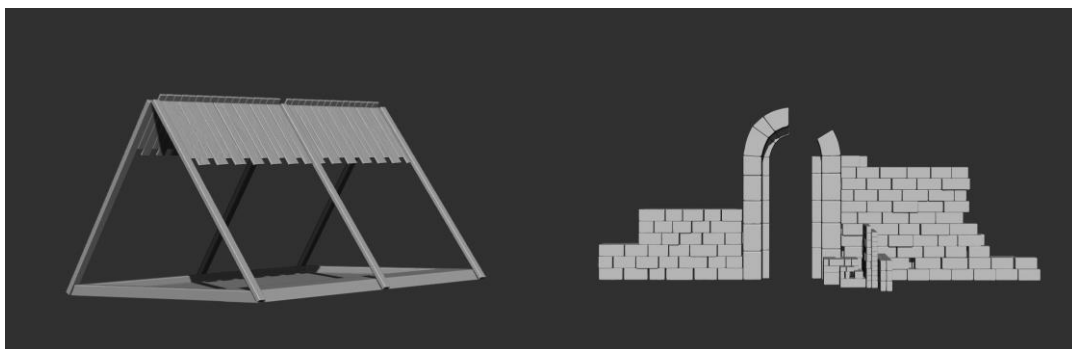


Figure 37: Pieces of Beekeeper house and one of the torn gates



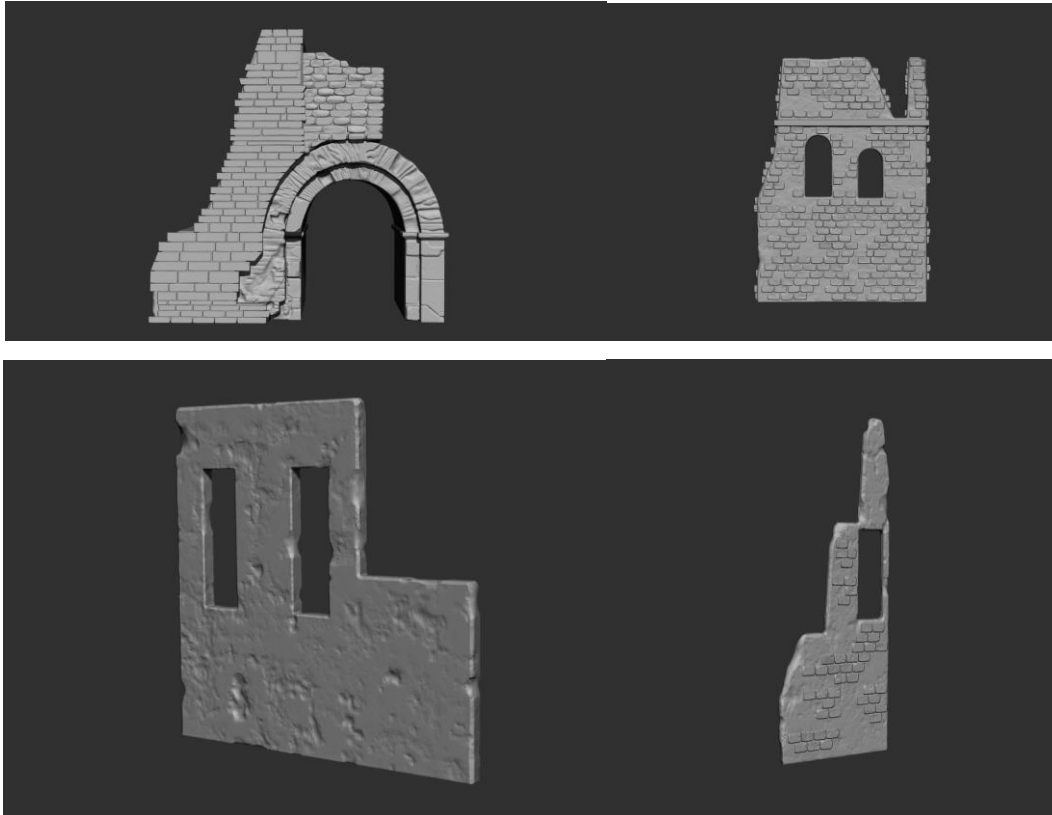


Figure 38: Pieces of destroyed building structures

The Swivel gun and two different versions of 36 pounder cannons were made the way I made my Enfield Snider Rifle. Lowpoly blockout in Maya, blockout taken to zbrush to sculpt details and make final high poly model and then clean up of final low poly model was done in Maya.

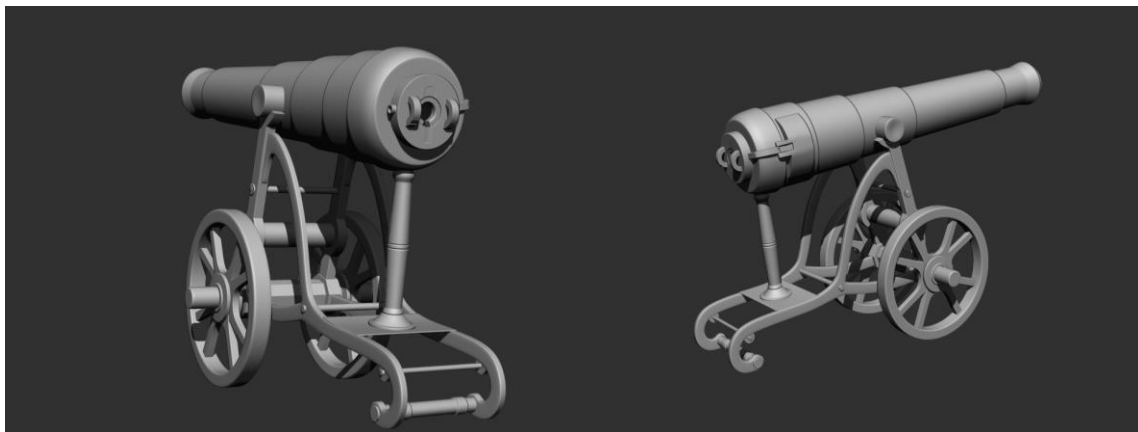


Figure 39: 36 Pounder Cannon based on trophy version

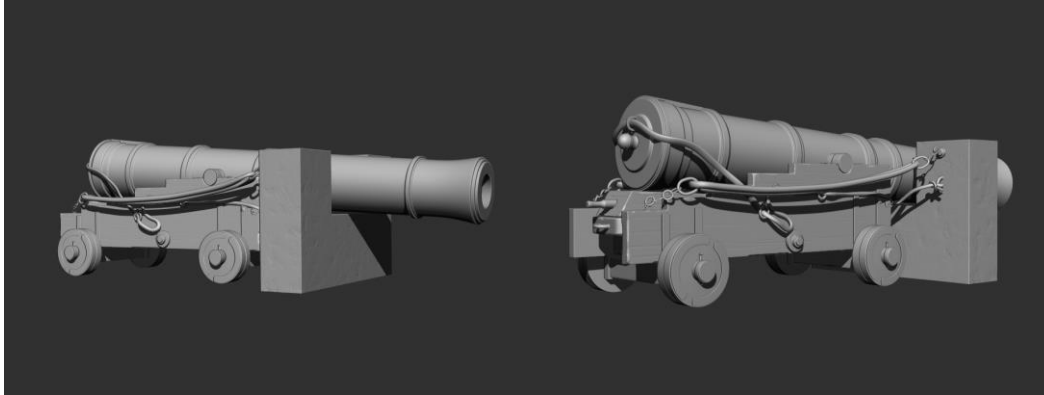


Figure 40: 36 Pounder Cannon based on the regular version



Figure 41: Swivel Gun High Poly mesh

3.3.12 Packing the UV: Every model needed a clean UV layout and that was done in 3D Coat. After having a packed UV layout for each low poly asset the details from high poly geometries were transferred to them through baking the normal, ambient occlusion and thickness maps from Substance Painter. Each character and asset was painted in Substance Painter with textures from Quixel megascan, Polligon and Textures.com libraries. Each asset is comprised of PBR shader. Physically Based Rendering or PBR is the method of realistic shading for models with measured surface values to accurately represent real-world materials. So the visuals of the game with PBR workflow looks very much believable.

For the characters I have multiple UV sets for each so I have a very good amount of texel resolution for my textures. It's true with more UV sets I had to consider more draw calls inside the game engine but I needed a moderate amount of UV sets for all the models so none of the textures look jagged or pixelated. Robert Clive has 9 different UV sets and Soldiers has 11 for each. The haircards make one single UV set while the eyelash and Brow have another different UV set.

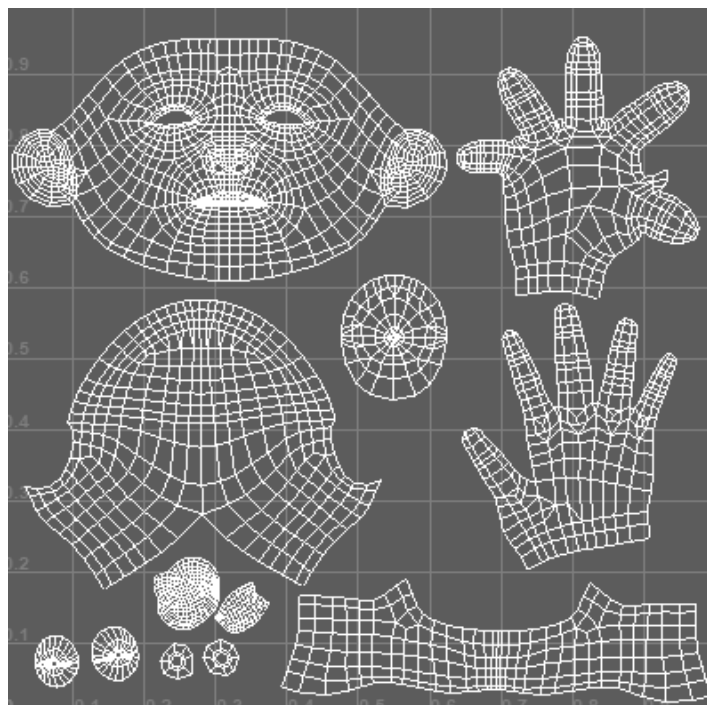


Figure 42: Face, Hands and eyes in one UV set for Robert Clive

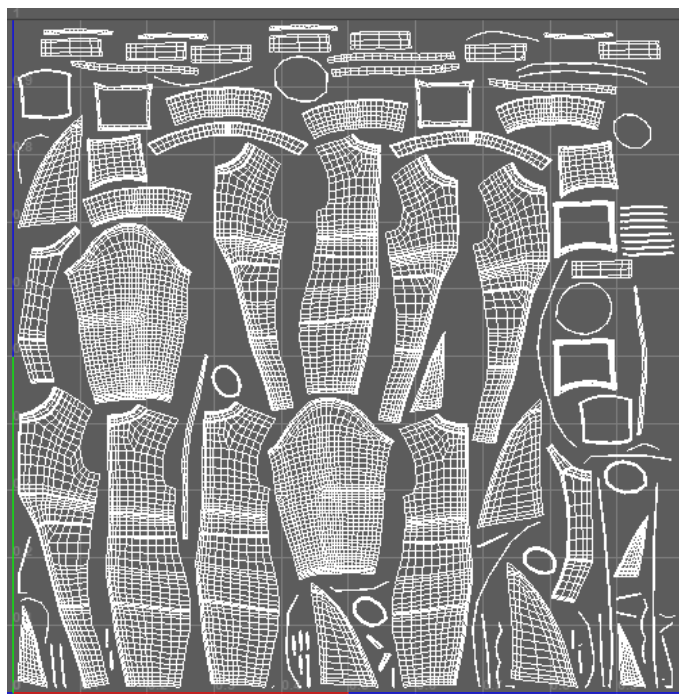


Figure 43: One single UV set for the Coat of soldiers

To create the UV seams and pack the shells with equal scale distribution I've used 3D Coat which saves a lot of time and unlike in Maya I didn't have to cut the seams and place the shells manually with a lot of trial and error.

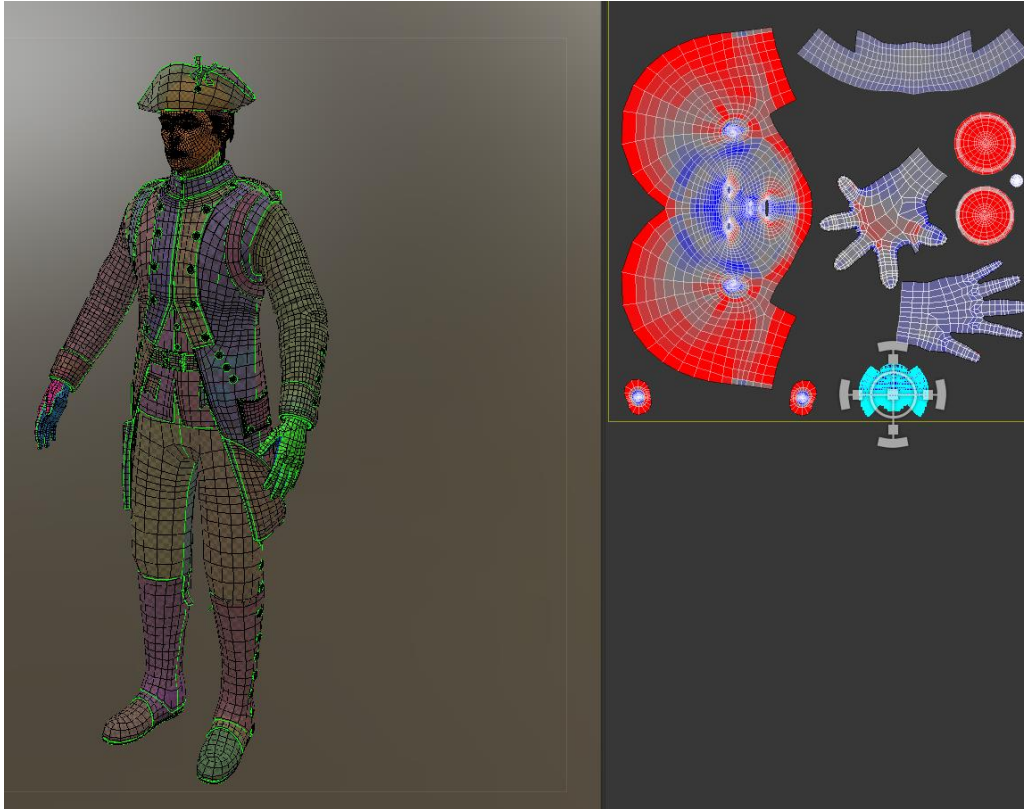


Figure 44: French Soldier UV packing inside 3D Coat

For the Enfield Snider Rifle I just needed one UV set to pack the entire gun. It was a very quick process to seam, pack and scale the shells inside 3D Coat to get the low poly mesh of my gun ready for texturing.

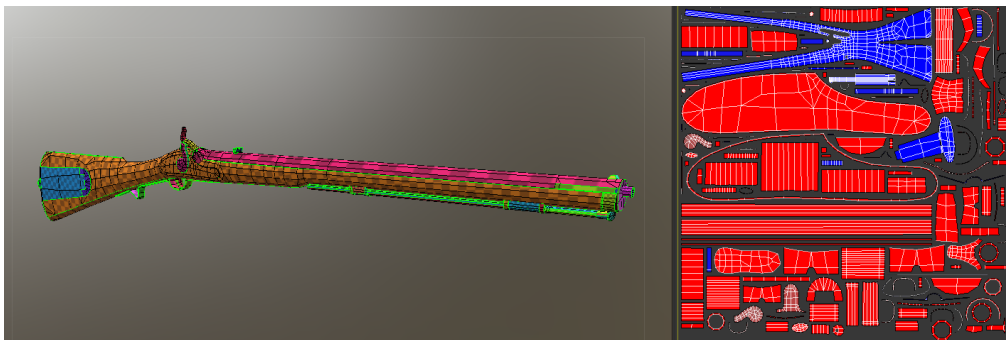


Figure 45: UV set packing of Enfield Snider in 3D Coat

For the Fort Modules I had to manage the UV cleverly so that I don't end up with having to deal with too many draw calls. In some cases one module alone took one UV set, in other I could manage a number of modules to fit together in one set. Some assets have been repeated multiple times so it wasn't a problem if all of them shared a UV set. Timari Fort contains 50 different UV sets in total while Arcot Fort has 13 sets.

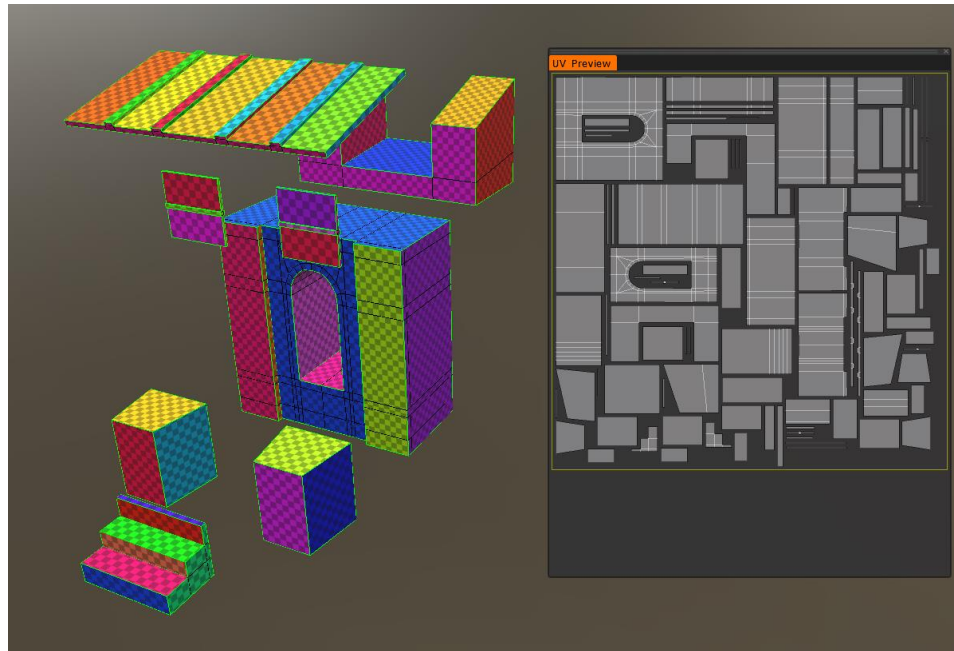


Figure 46: Fort Side Gate Group UV set of Timari Fort

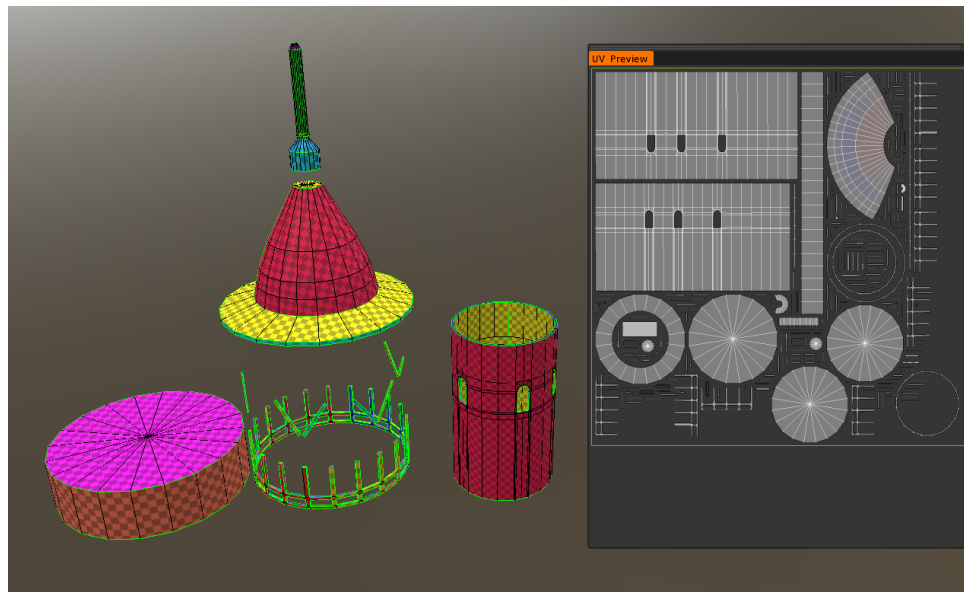


Figure 47: Round Tower UV set of Arcot Fort

3.3.13 Baking the maps: After getting all the UV sets ready for my low poly models I needed to bake the details from high poly meshes into variety of maps. The low poly models would be equipped with baked maps in order to provide similar visuals to their high poly counterparts. The low poly meshes were taken into Substance Painter and the bake mesh option allowed me to put a high poly mesh for each of it's low poly counterpart. Then after tweaking some settings it was just a matter of time to bake the



normal, ambient occlusion, thickness, world space normal, position and curvature maps. For all of the characters I've baked maps using 4k resolution. The maps for Timari fort assets are baked in 2k resolution. Maps for gun and arcot fort modules are baked in 4k resolution. Substance Painter Baking option provides amazing max frontal and max rear distance controls so I didn't need to build extra cage meshes for my models.

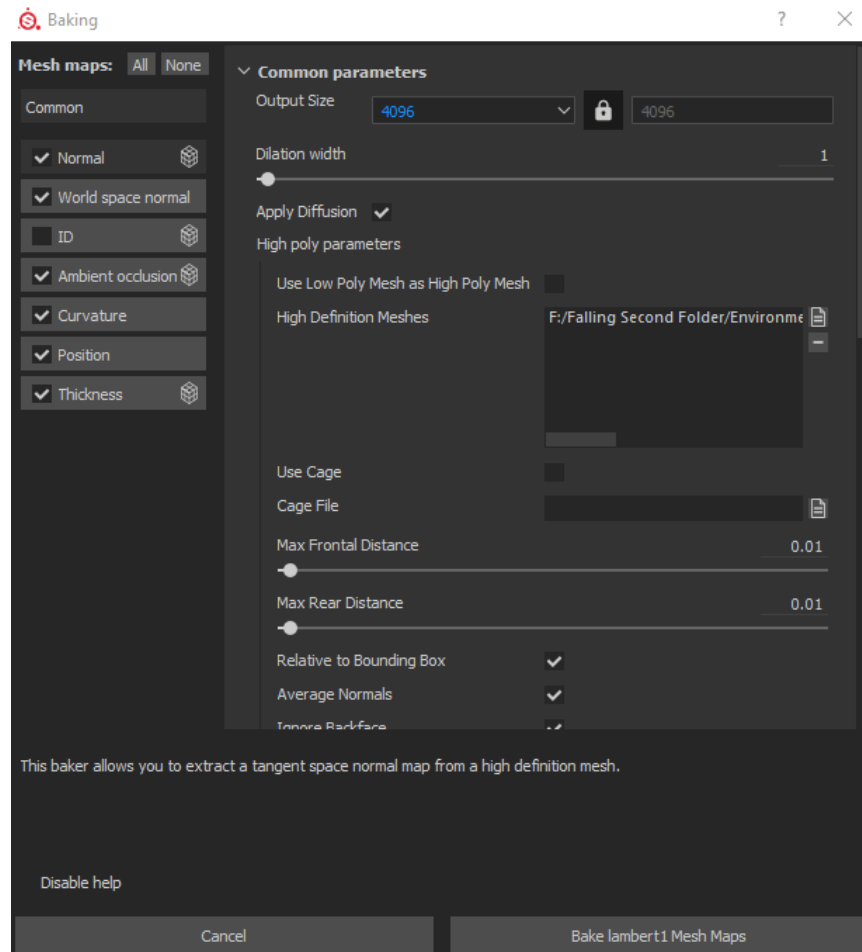


Figure 48: Texture Map Baking options in Substance Painter

Most uv sets are comprised of different meshes so I needed to bake ID maps too. I did this process in Maya's texture map render options. First I put different colored lambert materials in each of the meshes for one group of a uv set. Then I made a copy of that group and put a regular lambert material for the entire group itself. Then I baked the colored ID map from the first group onto the second group.

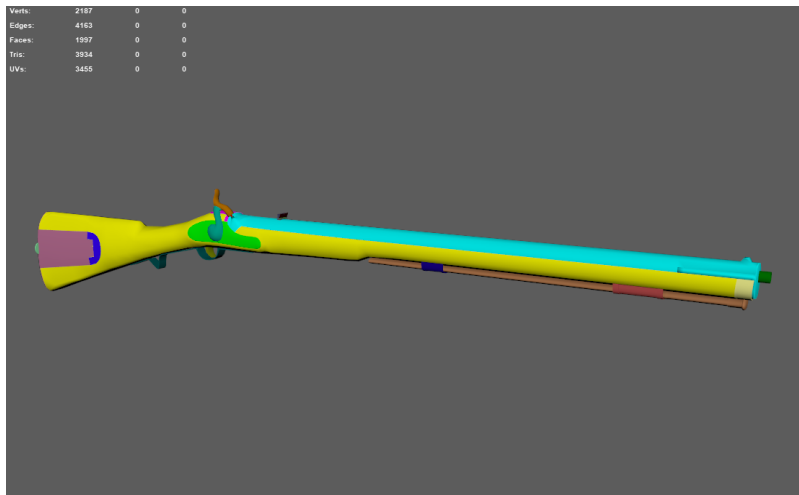


Figure 49: ID Map of Enfield Snider baked in Maya

3.3.14 Texturing Process: Texturing the assets and characters was one of the two most favorite and creative processes of mine in this pipeline. I have used Substance Painter for texturing everything while utilizing texture maps from Quixel Megascan, Poliigon and Textures.com libraries as well as the built in alphas, procedurals, masks and painting tools of Substance Painter.

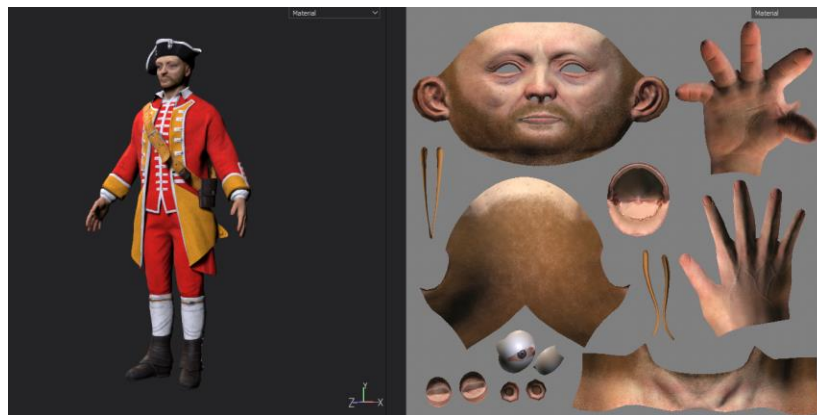


Figure 50: Texturing Clive's Model inside Substance Painter



Figure 51: Look development of Clive in Substance using it's iray renderer



Figure 52: Look Development of British Soldier



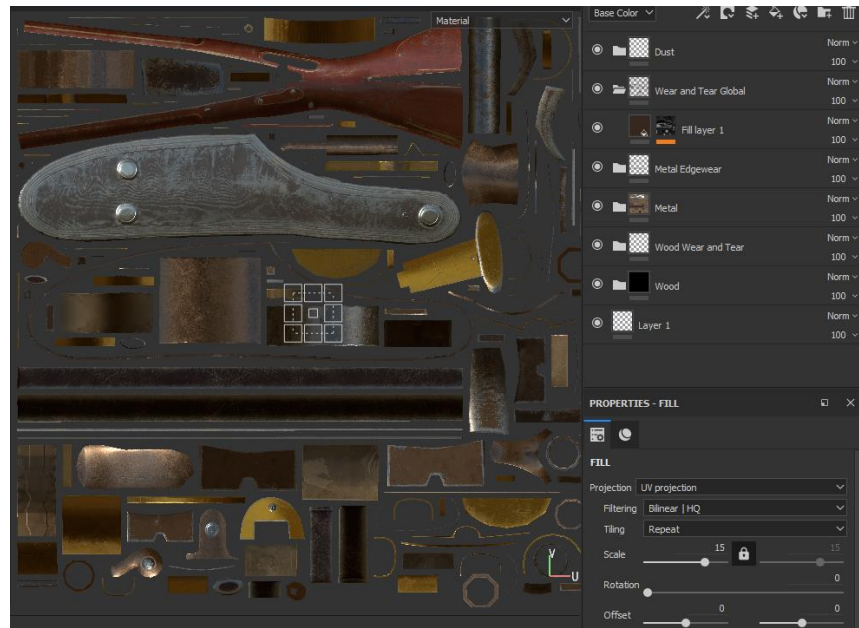


Figure 53: Layers put to paint and use maps for texturing the rifle

In many cases instead of manual painting I could get away by texturing procedurally using the smart masks, procedurals and grunges found inside Substance Painter



Figure 54: Smart Masks in Substance Painter

The different generators and smart masks utilize the baked maps plugged with the model to procedurally generate wear and tear, dirt, dust etc. From the layer stack I can put a smart mask on any layer and it will procedurally generate whatever the mask is aiming to over the layer that sits below. I could also save a total folder of different layers containing masks as a smart material. Whenever I use that smart material on another model it will automatically generate textures using baked maps of the model. That is why they are called Smart Material.



Figure 55: Smart Material library inside Substance Painter

The amazing viewport powered by Yebis and iray renderer inside Substance allowed me to do texture painting and look development at the same time. The particle paint system was useful to paint drips and moss.



Figure 56: Painting drips and moss in Substance Painter

Based on my concepts and the overall color palette I planned each of the modules was painted through lots of iterations.



Figure 57: Modules of Timari fort in Substance Iray renderer



Figure 58: Modules of Arcot fort in Substance Iray renderer

3.3.15 Final Look Check: After painting and generating all textures it was time to put all the modules together and look at the final models as well as the weapon and characters.



Figure 59: Realtime characters of Falling with textures



Figure 60: Realtime model of Enfield Snider Rifle with textures

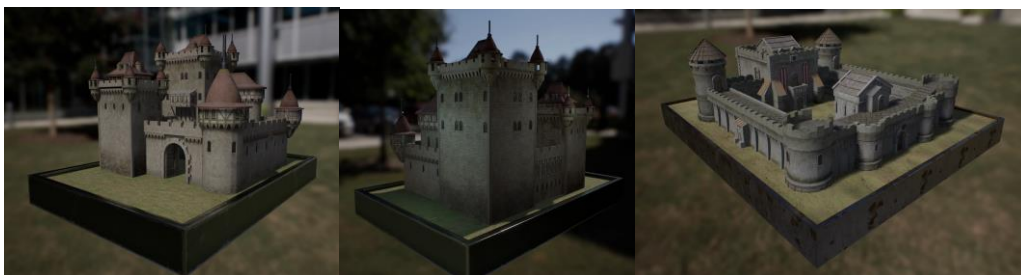


Figure 61: Realtime models of Arcot and Timari forts with textures



3.3.16 Creating the terrain in World Machine: After analyzing the story and opentopography I was able to realize how the map of my game’s world would look. Open topography is an open source of geographical data. It facilitates community access to high resolution, science-oriented, topography data and related tools and resources. Looking at the current map of Arcot and Timari area through open topography I could plan up where to put peaks, hills, erosions, road etc. Lidar or Light Detection and Ranging refers to the remote sensing method for examining the earth surface. With Lidar systems we can achieve data on manmade and natural environments that are accurate, precise and flexible. Due to unavailability of lidar data within that area I wasn’t able to directly generate a height map from the existing open topography data. So I had to manually generate everything in World Machine.

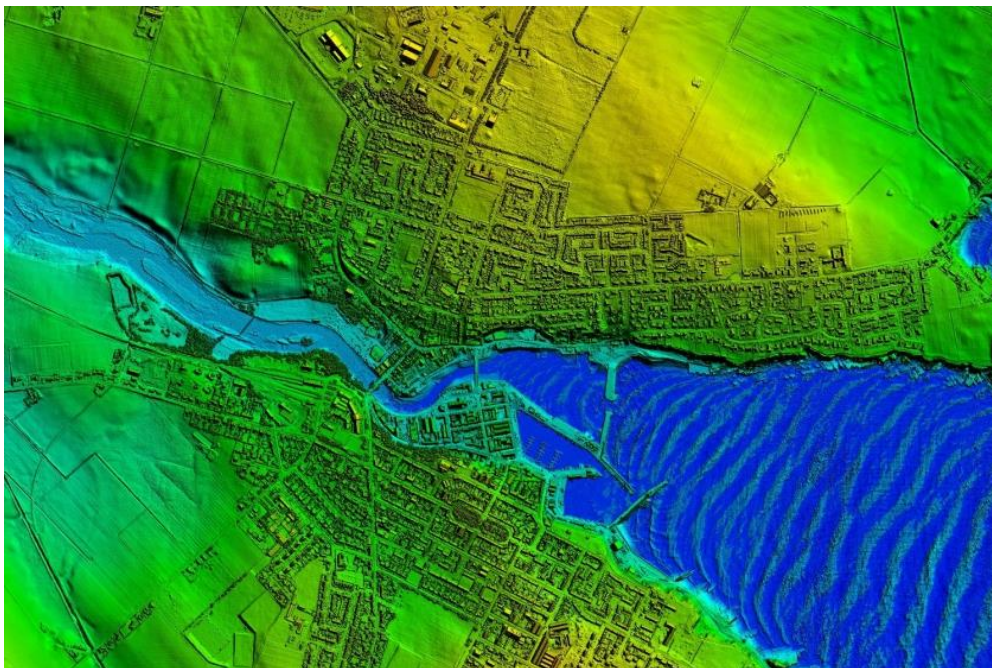


Figure 62: Image of LiDAR data from Atkins Global

I was able to create the terrains of Arcot and Timiri procedurally through the node based, non-destructive workflow within World Machine. The exported height map in RAW16 format is the main map to generate the terrain inside the Unreal Engine. The other weight maps which are exported in RAW format are for blending the landscape layers. Creating a layer blend node in Unreal is the perfect way to set up the terrain with the same colors

as inside World Machine. So it's really important to setup the colors of the terrain inside WM as well as creating the total structure.

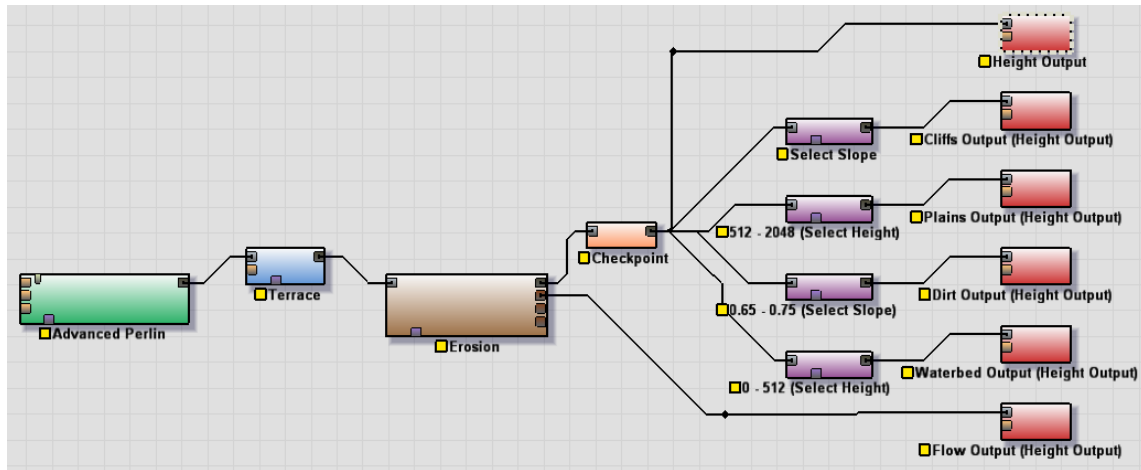


Figure 63: Nodes used inside World Machine to generate my terrains

After getting the terrains ready and exporting the output maps it was time to set them up inside Unreal Engine to have an idea how they are going to look. After using the height output to generate the terrain I quickly setup a material with layer blend node to put the weight maps and check the colors.

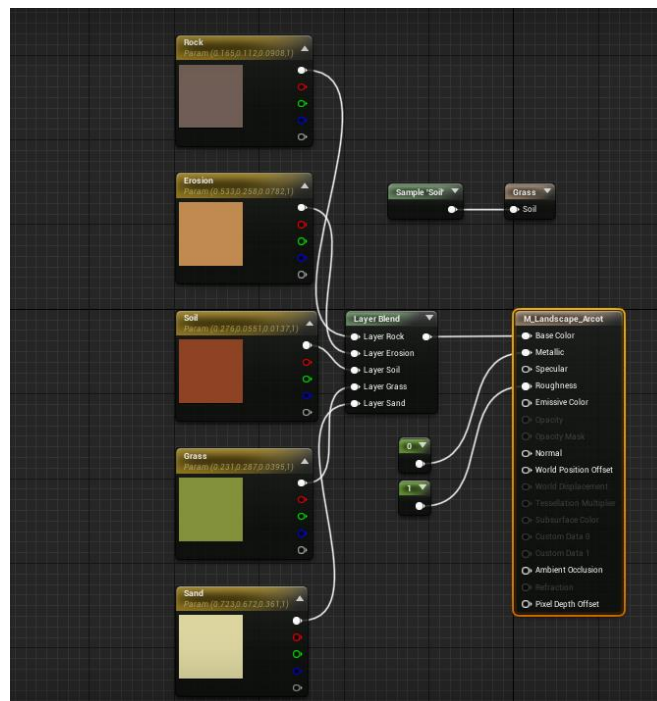


Figure 64: Landscape Material network inside Unreal Engine

Putting this material on the terrain with colors enabled in the weight channels I was able to get the same result as inside of World Machine.

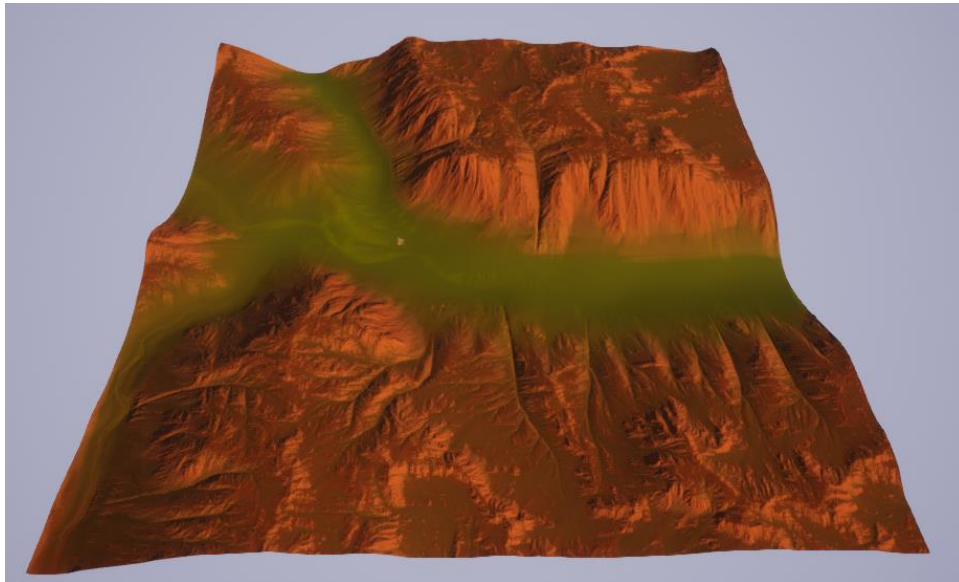


Figure 65: Arcot Terrain overview in Unreal Engine

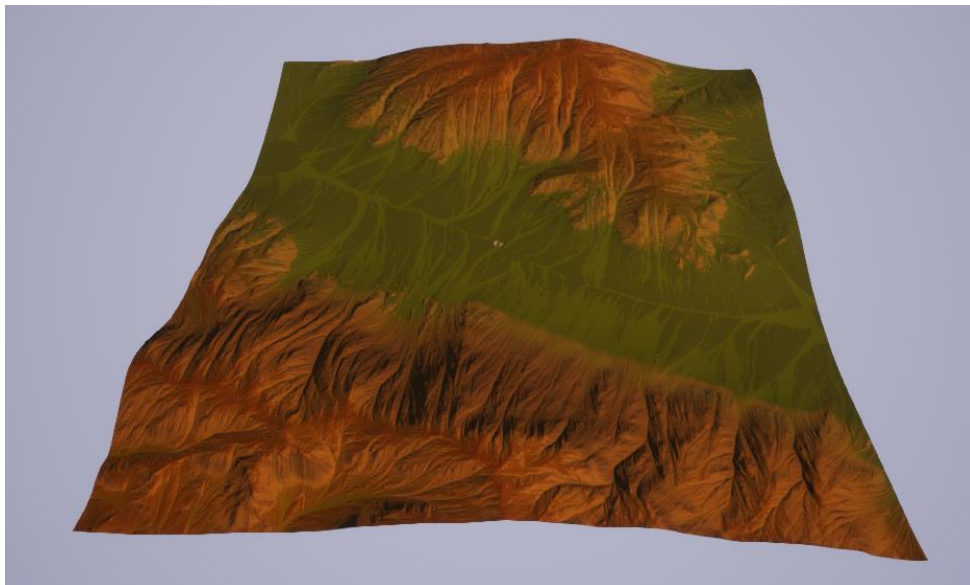


Figure 66: Timari Terrain Overview in Unreal Engine

3.3.17 Skinning & Rigging: Painting skin weights of the characters is a must before rigging any character for animation. The arms, legs, head, belts, bag and all the polygons that will be deformed should be deformed in the appropriate way so the animation looks realistic. In order to achieve the best deformation, I carefully painted the weights within different parts of my characters. I did this with Maya's weight painting tools. First of all I

had to place the bones to have a complete skeleton I can bind my mesh with. Then I had to carefully set orientations of the bones.

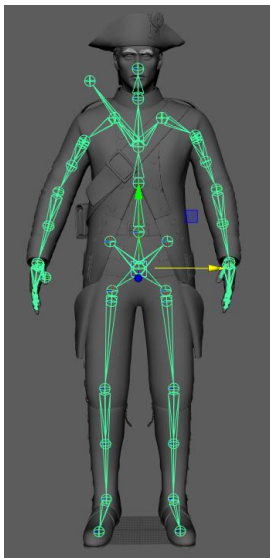


Figure 67: Placing Bones of the skeleton

After binding the skeleton with my character mesh it was time to set up some poses in different keyframes to check how the deformations are working. This is the first step before approaching to paint weights in different parts of the character.

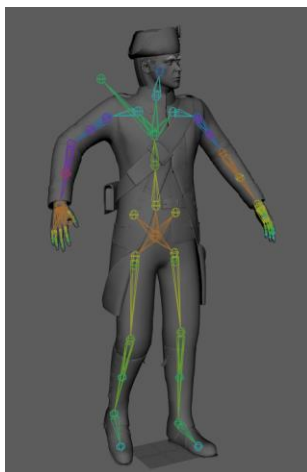


Figure 68: Checking out the deformations

With different poses setup across different keys I started to paint weights for every part of the mesh. Any part of the mesh that will be deformed is correspondent to the bone that is responsible for the deformation. For these particular bones I had to add or remove weights to be able to deform the coat, shirt, pant, hat etc appropriately.



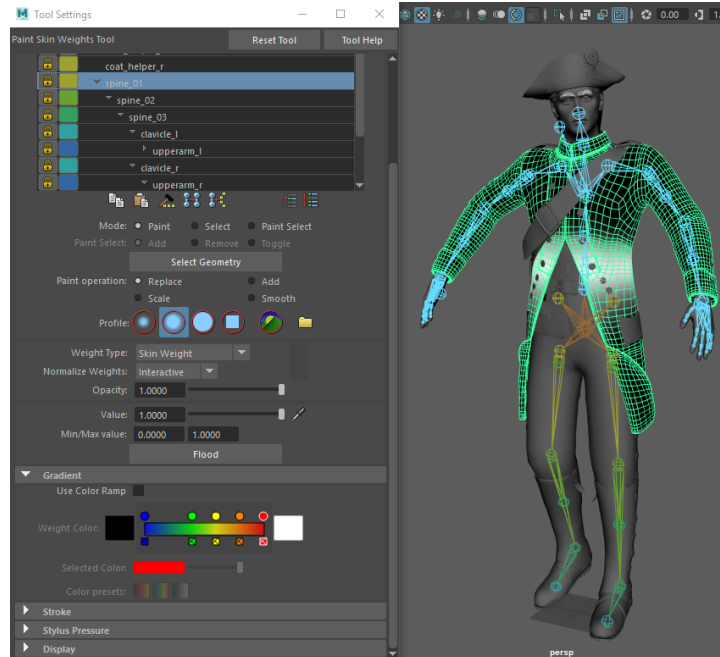


Figure 69: Painting weights with Maya's weight painting tool

The rigging was done with Maya's Human IK option. From the human IK tab I just had to set the character to my skeleton definition and the source had to be control rig so I have all the control curves to animate my character.

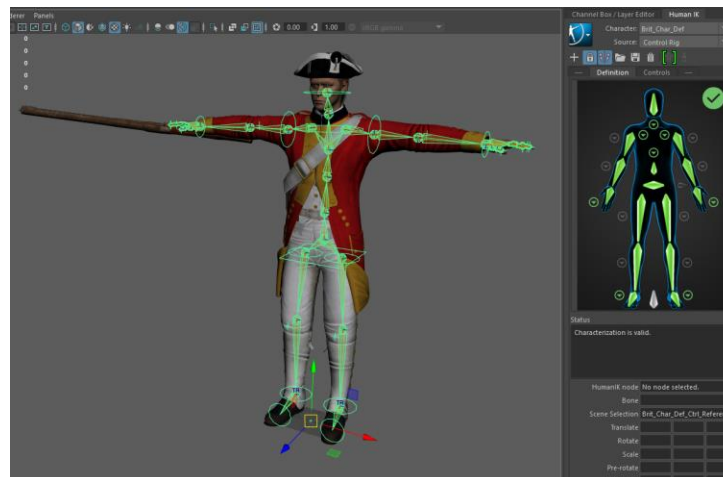


Figure 70: Setting up the control Rig

Since my animation approach was based on mocap/animation data from Mixamo and Unreal Engine I had to place the mixamo rig attaching with Human IK as well so I was able to retarget animations from both Mixamo and Unreal onto my character. I would also be able to edit the animations using the control curves I set before. So I brought the mixamo rig and set it up with T pose for Human IK.

3.3.18 Animation: The motion capture library of Mixamo was very useful to find the animations of movement and action I needed for the characters, idle, walk, run, aim and fire with the gun all these animations were taken from mixamo, retargeted on my characters and then heavily edited by some keyframing within maya. It was essential to edit the animations with gun accurately because the animations from mixamo don't match properly with the Enfield Snider rifle my characters are equipped with.

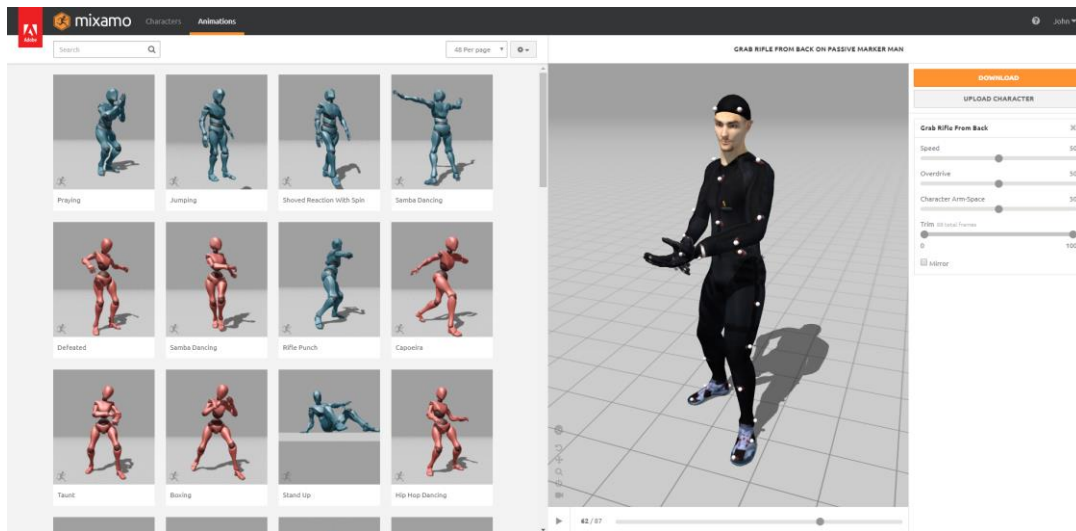


Figure 71: Mixamo animation library

After importing the necessary animations from Mixamo in Maya first I had to put all the animations together in the time editor so I had all of them within a clip. That would make editing process a lot easier. Then I set the Mixamo rig in T pose and had it enabled for Human IK. Now I could set my character definition for character and the mixamo definition for source and the animations from mixamo had been retargeted onto my character. Then I baked the entire animation to my control rig and I was able to edit the keyframes using control curves. I had all sorts of control over the animations now. I could add or remove keyframes, modify movements, place the arm, fingers differently to match the gun animations with my rifle as well as having the rifle animated during fire, recoil, reload etc.



Figure 72: Editing the animation montage

When I was satisfied with all the animations I baked it again onto the skeleton. Now my character has all the animations attached to it. Then I cut the entire clip in different animations using time editor and exported them one by one for taking them into Unreal Engine.



Figure 73: Autodesk Maya Time Editor

3.3.19 Final Check-up: After the animations were exported, I checked everything in my file hierarchy in case any cleanup was needed. The texture maps, assets, animations, characters everything was made sure to be organized in a tidy file management system I've maintained through entire pipeline. The total size of all the files together was 266 gigabytes. Since I had a clean organization it was an easy process to set them up inside the contents folder of my Unreal Project I was about to create.



## CHAPTER 4

### Developing the game inside Unreal Engine

So everything was ready and the cooking within Unreal Engine 4 began. The terrain, assets, characters, animations, textures everything was brought in and integrated with each other. The material editor was used to utilize the textures properly. The blueprint of Unreal Engine was used to create the gameplay mechanics and enemy AI. The storytelling is done using still images of Clive's 3d render with texts appearing on the screen. The loading screens with images and texts provide the background of the story and mission objectives sequentially. The third person movement of the game is very much inspired by the likes of Gears of War, Red Dead Redemption 2 and Metal Gear Solid V The Phantom Pain. The lighting, post-processing and colour grading were done with the fantastic tools Unreal Engine provides us with.

#### 4.1 Creating the project and importing everything in Content Browser:

After creating a new project inside Unreal Engine I brought all the necessary things I needed inside the content browser. The Content browser allowed me to maintain my nice hierarchy and keep everything in a tidy way.

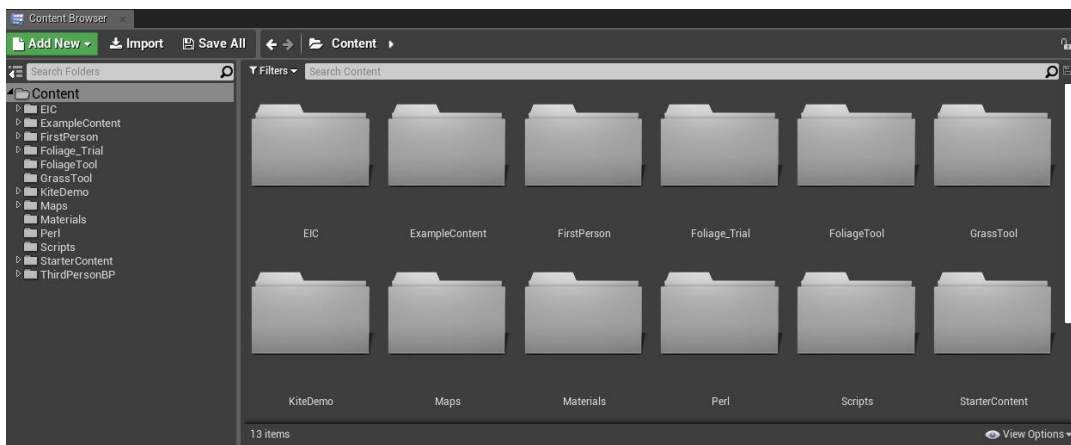


Figure 74: Content Browser of Falling

To fill up my environment I needed some extra assets so starter content, foliage tool, grass tool and kite demo assets were also brought in.

## 4.2 Setting up the animations for playable character:

To set up the overall character blueprint I needed to put the animations with my character first to have my locomotion ready.



Figure 75: Previewing Animation in Unreal Engine

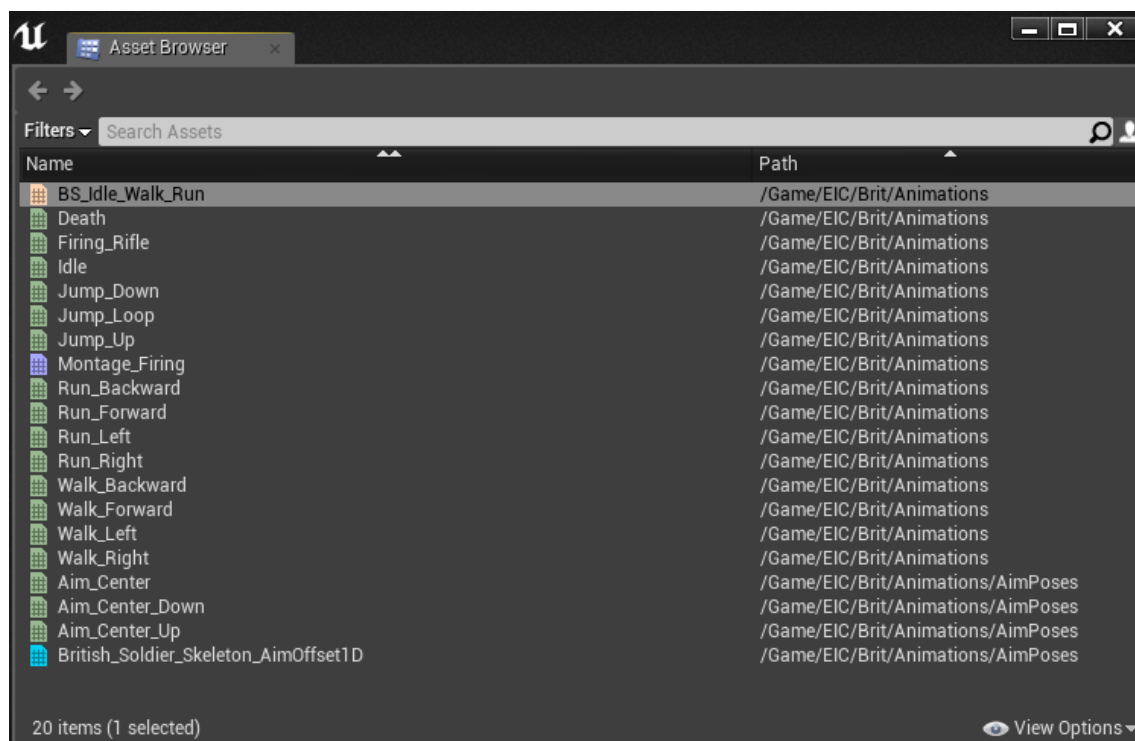


Figure 76: List of animations

The blueprint system in Unreal Engine is a visual scripting node based workflow. With the use of nodes all scripting work in Unreal Engine can be utilized by Blueprint which is powered by C++ language. I had to create the animation graph and event graph for character blueprint that are responsible behind all the animation functioning within my

game. After importing the animations I created a blend space to blend the walk and run animations towards left, right, forward and backward directions. It was necessary to start the character walking first from idle before he starts to run otherwise it would look very unrealistic and not convincing. Based on the character's speed and direction the animations are being switched in the blendspace.



Figure 77: Blend Space for Idle, Walk and Run

### 4.3 Making the character playable:

I created the anim graph and set up my locomotion that contains animations from Blend Space with the jump animations. This locomotion is also the state machine responsible for the movements of my character. It also has the death animation. Based on variables it runs the different animations. This variables are controlled from Event Graph that is going to be explained later.

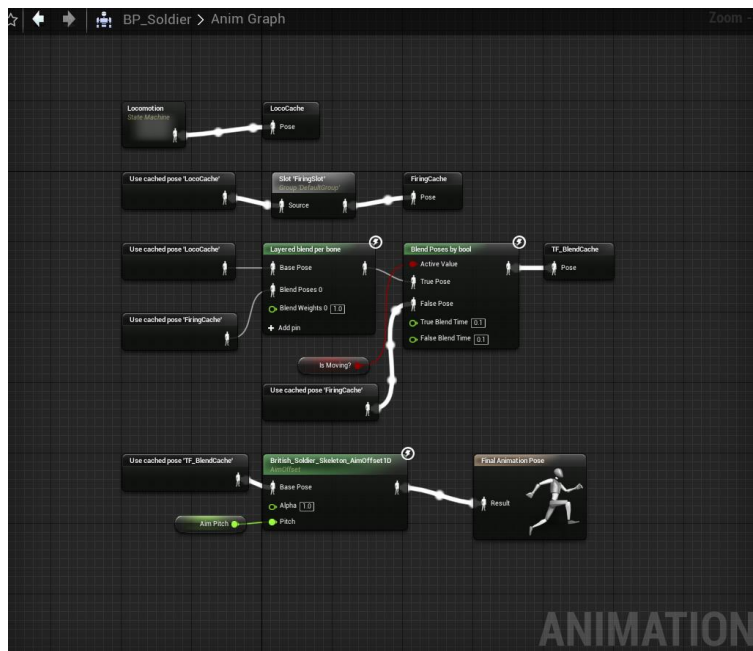


Figure 78: Animation Graph of the Brit Soldier Blueprint

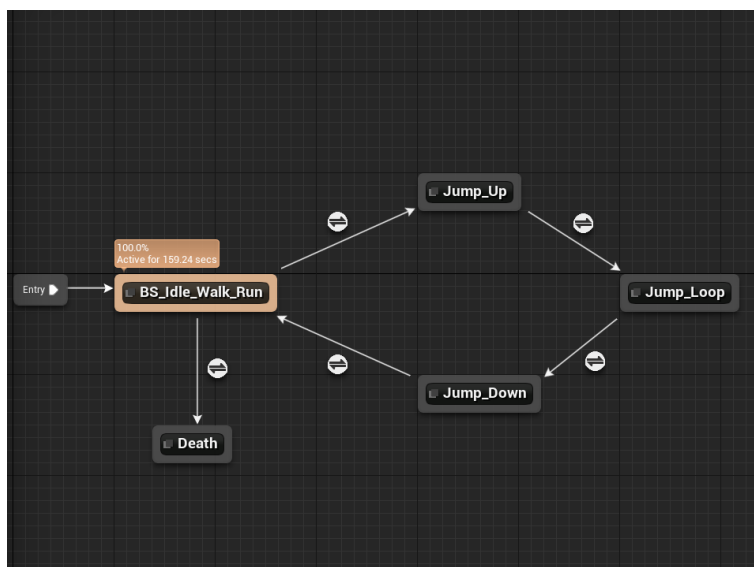


Figure 79: British Soldier Locomotion

In the animation graph the caches are created to reference an animation so it could be used elsewhere. FiringSlot has the firing animation montage I set up. It is also cached so I can use both the locomotion and firing animations from another node based on true/false conditions.

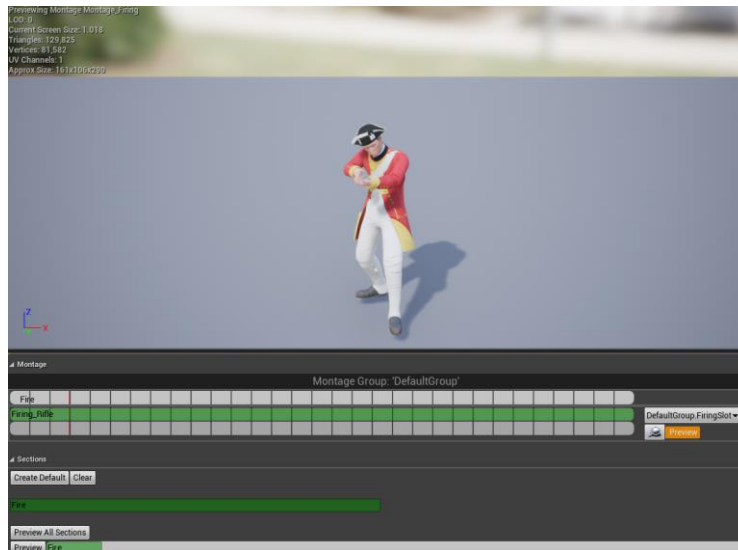


Figure 80: Firing Animation montage

Based on the Is Moving active value it is decided whether the character moves and fires or it keeps firing in a standing pose. Once again we have a cache output from this and we use it to have our final animation pose that is driven by the aim offset. We also have an aim pitch that decides in which direction the character is aiming at.



Figure 81: Aim Poses

To make the character now playable with all the animations I started to work on the Event Graph of my character's blueprint. The event graph is comprised of nodes that are driving my British Soldier with the animations I set up in anim graph. The event graph contains actor and pawn. Based on the value of different variables and functions animations are played in the game.







Figure 84: Timari Fort Placement

Since I was already happy with the colors of terrain I didn't feel the need to do all the work of putting texture maps for each color and create a complex material node with macro variation and other things necessary to make the different textures look naturally tiled and blended. It would also require some landscape painting work to make the tiling less obvious. Besides I would put grass and foliage so the plain colors wouldn't harm me.

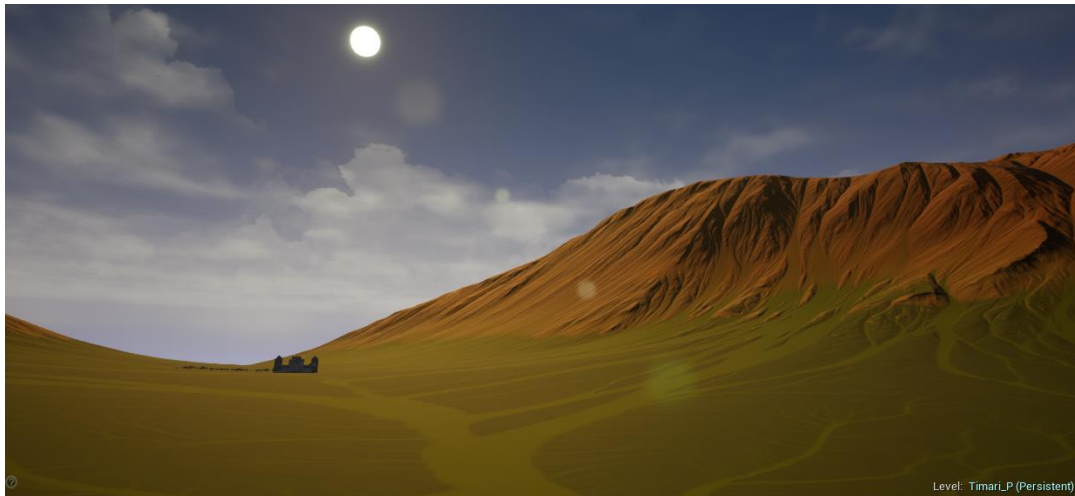


Figure 85: Checking out the Terrain with colors

To populate the terrain with grass I set up a grass node in my landscape material that enables all areas covered by Grass layer to be populated with the grass tool procedurally. The grass tool contains grass meshes with their dedicated grass material. The material not only controls the color but also a lot of other things as bright and dead grass, wind normals, fade with cull distance, cheap distance camera fade, height offset etc.

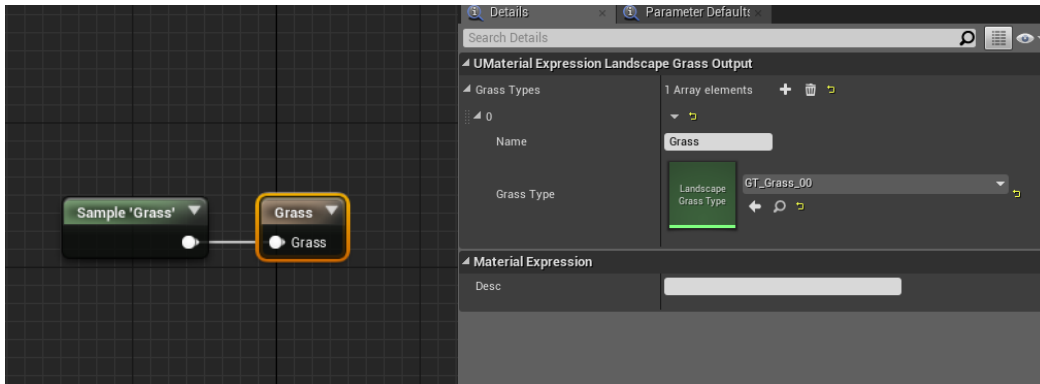


Figure 86: Populating terrain with grass from the landscape material

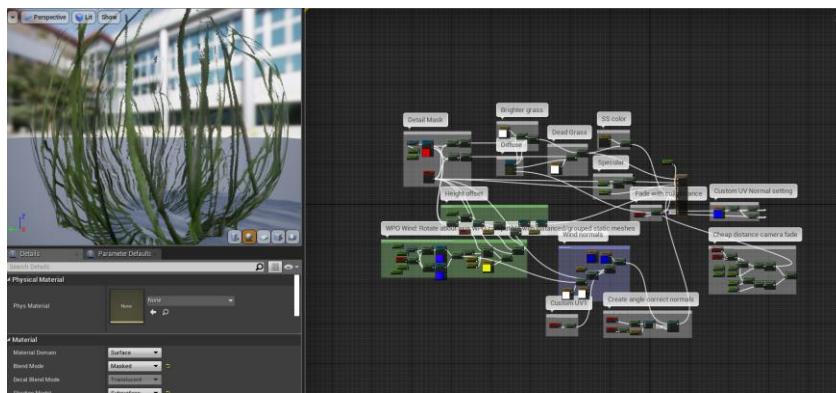


Figure 87: Field Grass Material

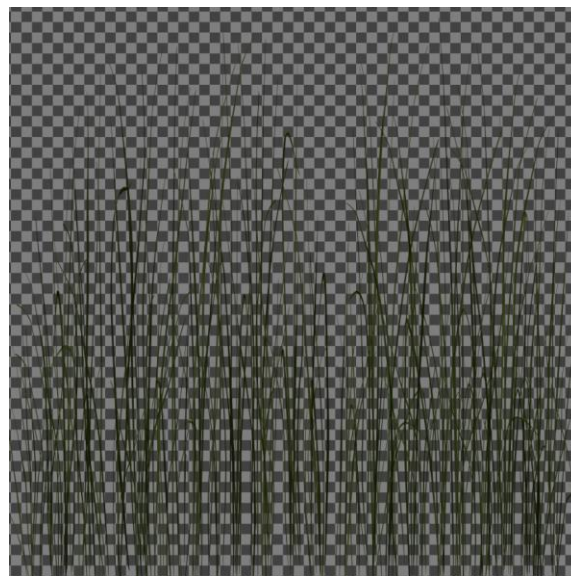


Figure 88: Field Grass Diffuse

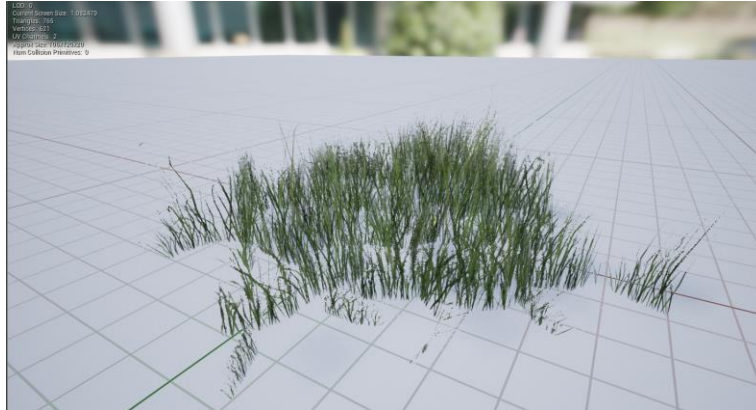


Figure 89: Field Grass Output

To populate the terrain with foliage I put a variety of foliage from the foliage tool. I would have a procedural foliage spawner that generates foliage included in it's foliage types procedurally in the selected area of a terrain.

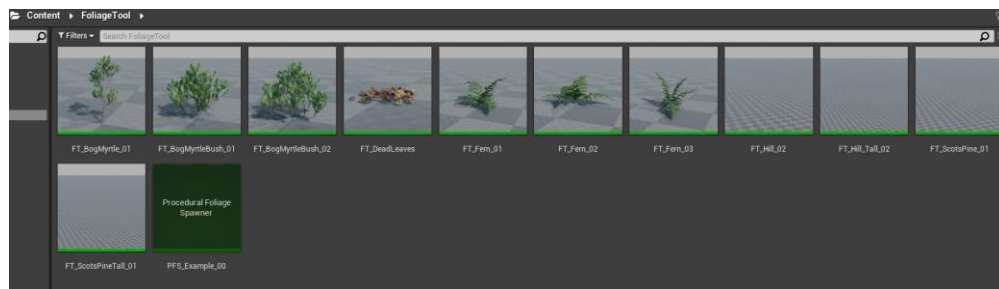


Figure 90: Foliage tool contents

Each of the foliage mesh is powered by diffuse and normal atlases in it's material node.

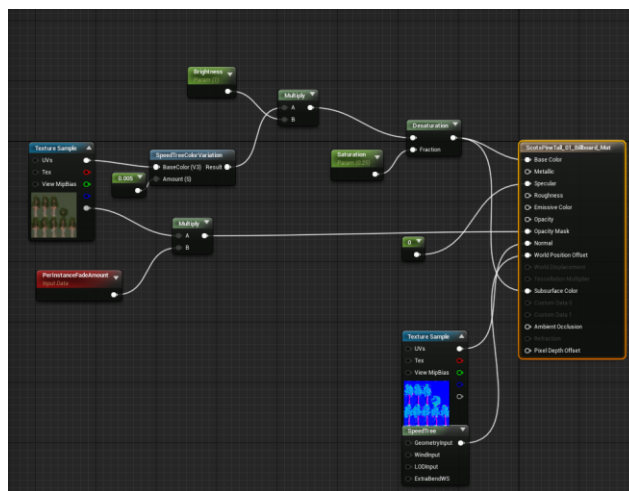


Figure 91: Billboard material of ScotsPine

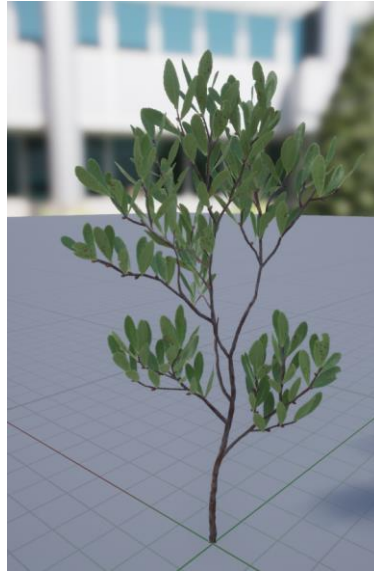


Figure 92: BogMyrtle Output

When I was satisfied with the variety of foliage set up inside my foliage tool I then put them in Procedural Foliage Spawner. I could generate the foliage in any part of my terrain any time and optimize them by tweaking the density, lod etc while checking the game's performance in real time.

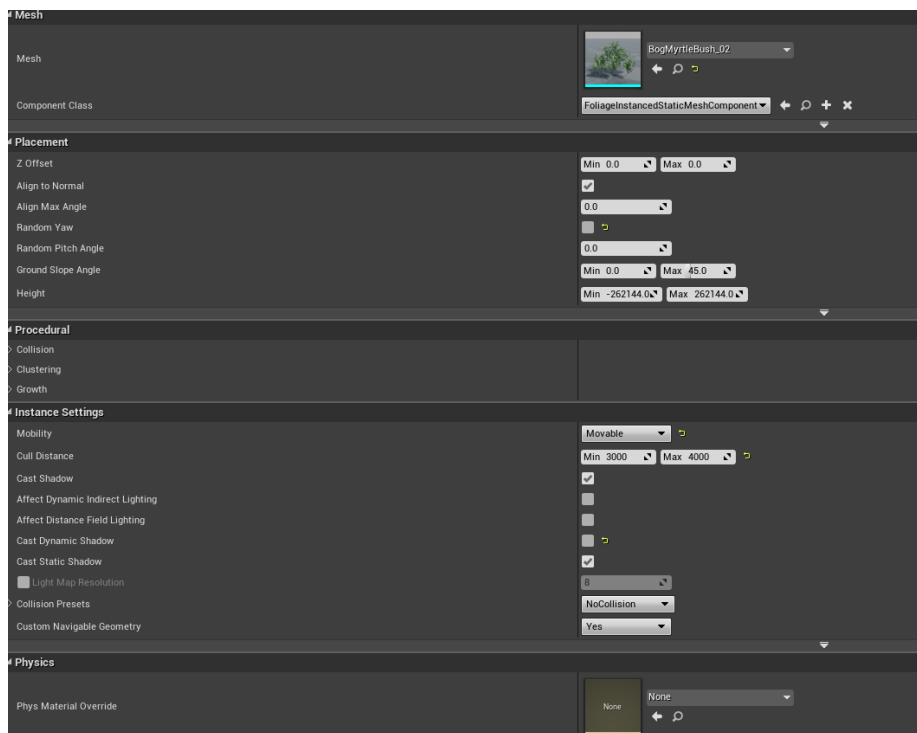


Figure 93: Options that can be tweaked for each foliage

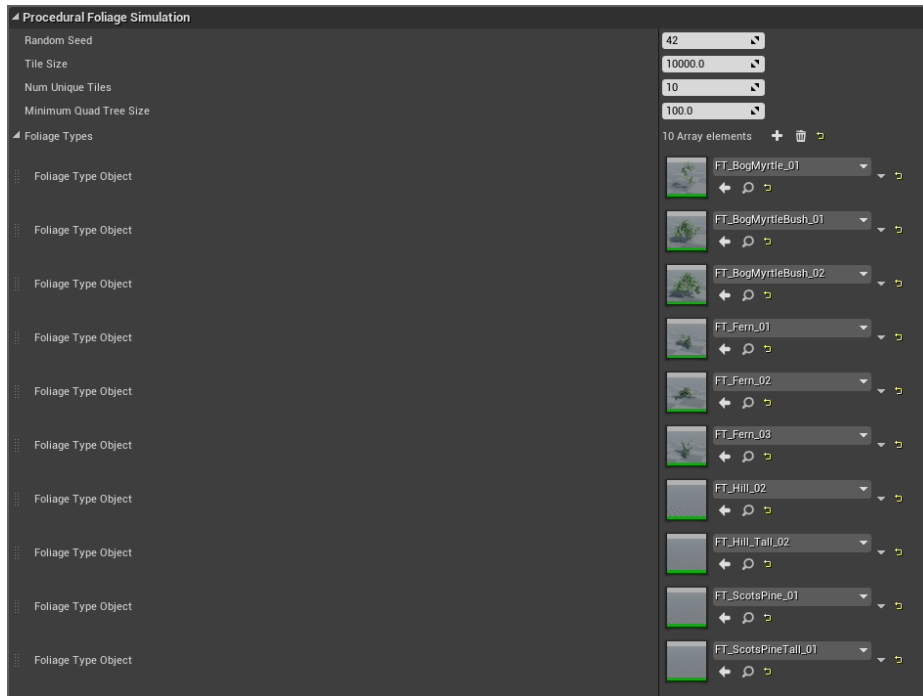


Figure 94: Procedural Foliage Simulation tool

After generating some foliage and already having the grass on my terrain it was time to setup the lighting and post process first before diving deep into optimization with not only foliage but also some other things.



Figure 95: Arcot Terrain populated with grass & foliage

## 4.5 Lighting of the world:

4.5.1 Sunlight: I wanted a shiny but foggy morning time look for my overall environment. That would make my green dominated terrain come alive really well. So I started tweaking my directional light source settings to find out the sweet spot.

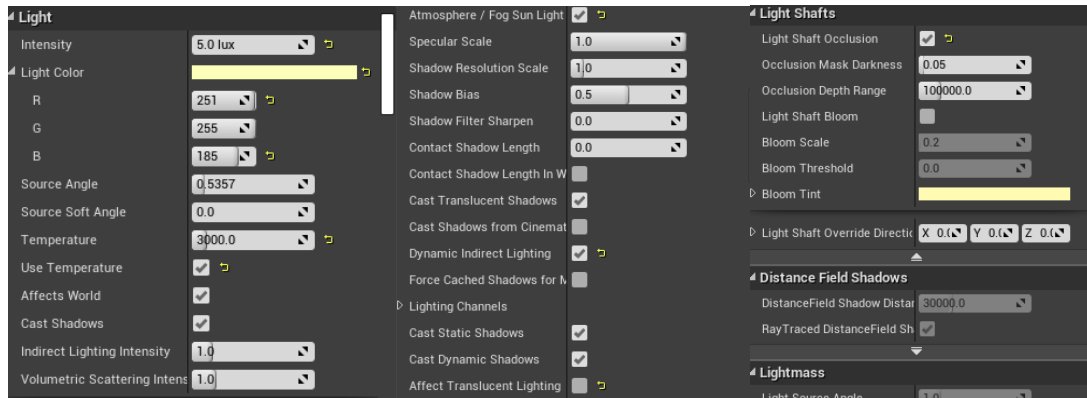


Figure 96: Directional Light settings

After playing with the settings of directional light I found the settings shown above as optimum for my light to be convinced as a sun.

4.5.2 Exponential Height Fog: To enjoy the post process effects and the overall effects of the light to their fullest I needed the exponential height fog in my scene. Not only for this reason but also to match my pre plan and world composition I was looking to put some fog. The exponential height fog found in the visual effects of Unreal Engine is a very powerful method to achieve some extra depth in the visuals of any video game.

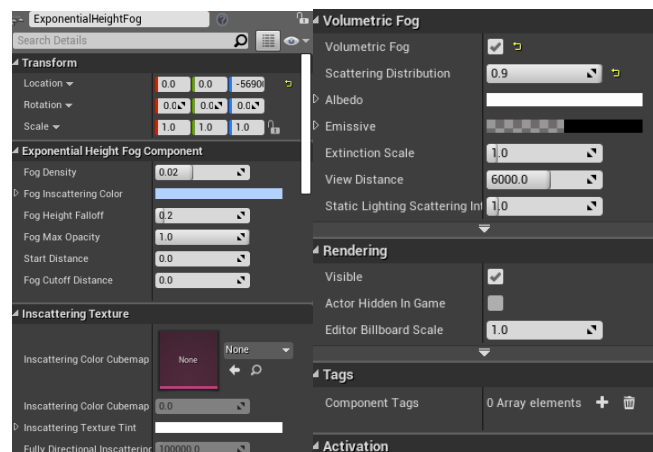


Figure 97: Exponential Height Fog parameters

Fog inscattering color and fog density control overall tint of the fog and thickness respectively. It is necessary to check the volumetric fog to achieve extra details like



volumetric God rays and bloom properly. The scattering distribution controls how much incoming light are scattering in various directions.

The sun and fog were pretty much the things I needed to setup and tweak to achieve my desired lighting. Since it's a daytime scene and I have no interior gameplay I didn't need to use any other kind of lights. Due to having so many moveable foliage as well as a completely moveable character it would've been stressful for the memory if I used a mixture of dynamic and baked lighting that is stationary lighting. Since the forts and some environment assets are the only static objects in my scenes I decided to have all my lights dynamic so the entire lighting and shadows of my game are being rendered real time putting less stress on memory and cpu while utilizing the gpu more.



Figure 98: Arcot Terrain with Sun and Exponential Height Fog enabled

#### 4.6 Post Process:

Unreal's post process volume provides all the post process one could ever want from their cameras. We can either control them directly from the cameras of our scene or we can use post process volume. The post process volume allows to either have post process effects for a certain portion of the level or for the entire scene globally. I used infinite extent to have the post process volume controlling the effects for my entire level.



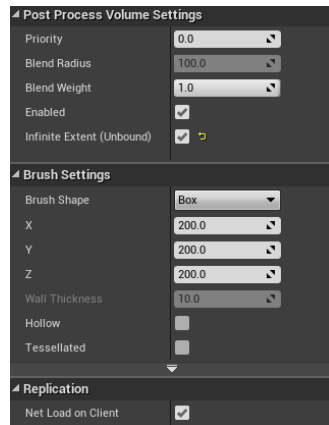


Figure 99: Post Process Volume enabled with Infinite Extent

I tried to utilize most lens effects to make my scene visually appealing and presentable with a desired level of quality.

4.6.1 Depth of Field: Just like regular DOF of cameras Unreal's depth of field comes with similar features to separate the foreground, middle ground and background of an image to provide a cinematic look in the game. It is kind of tricky to tweak and setup an optimal DOF for the player to enjoy the game with because sometimes the use of DOF can create disturbance for objects or enemies in distance to be noticed resulting in not so pleasing gameplay experience.

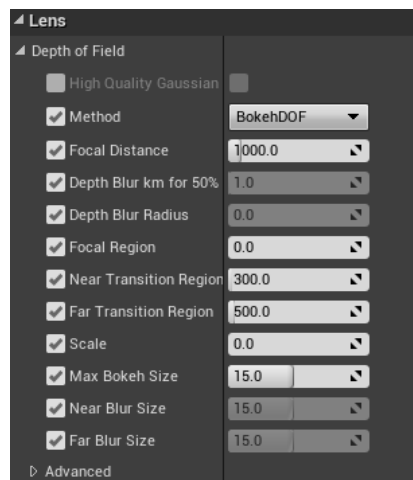


Figure 100: Depth of Field parameters

4.6.2 Bloom: Bloom is seen in real world as a light phenomenon. It happens with very bright objects in darker background. The bloom effect simulates what we can see through real eye when light hits the the film. The bloom effect gives the overall scene a different look than of that without bloom. In Unreal multiple wide gaussian blurs are combined



with different radius to achieve bloom effect. I have used the standard method and did not tweak much.

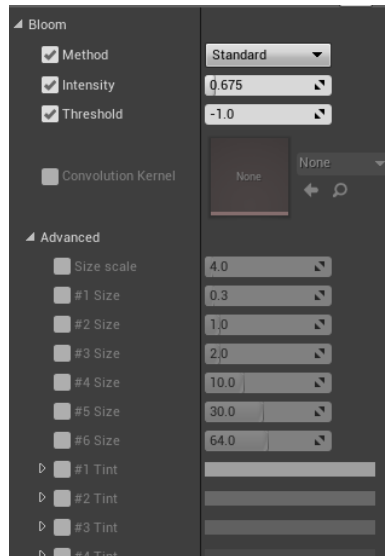


Figure 101: Bloom Parameters

4.6.3 Exposure: I have used the auto exposure or eye adaptation. It adjusts the exposure of the scene automatically depending on the brightness of the light. It simulates how human eyes adjust when switching from bright environment to darker environment and vice versa. I have used the auto exposure histogram with following settings.

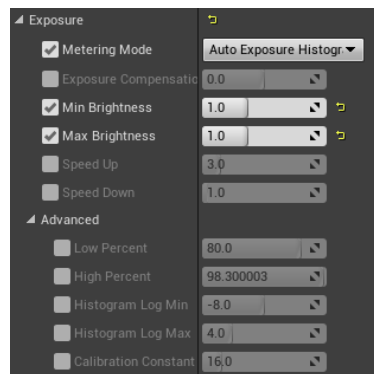


Figure 102: Exposure Parameters

4.6.4 Lens Flare: Lens Flare effect simulates the camera lens effect when looking at bright objects or lights. The light scatters in different directions which happens because of the imperfection in lenses resulting in some beautiful effect. In Unreal we can achieve the same effect through post process. It is important for the player to experience lens flare when looking at the sun. As well as the intensity I could control the color of tints visible in the flare.

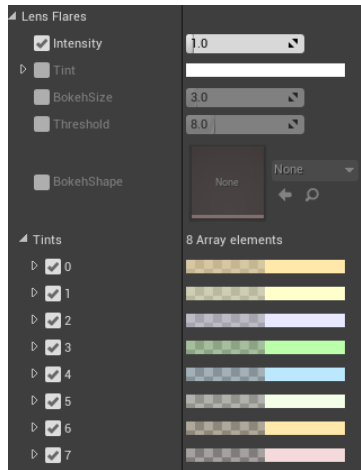


Figure 103: Lens Flare parameters

4.6.5 Light Propagation Volume: Light Propagation Volume is a feature that is still not ready for production from Unreal Engine. It aims to achieve Global Illumination approximately in real time. To make my sunlight and ambient skylight give the desired global illumination I turned this feature on. By using lattices and spherical harmonics light propagation volume represents spatial and angular distribution of light in the scene.

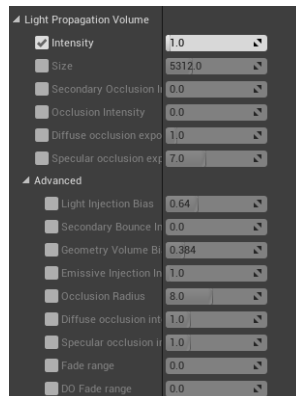


Figure 104: Light Propagation Volume parameters



Figure 105: Timari Terrain with Post Process features enabled

#### 4.7 Hardware Requirements & Optimization:

I needed to optimize the drawcalls, polycount, instances, collider meshes, lod and texture resolution within the game so that the game runs at 60 fps in 1080p and 1440p resolutions in the target hardware.

4.7.1 Hardware Requirements: I tried to make my game run at average mid range computers as well as utilizing its full potential in a widely accepted configuration by gamers in today's pc game market. The minimum requirements to run Falling at 1080p resolution are as of the following

- Nvidia Geforce GTX 560 or equivalent AMD DX11 GPU
- 1 Gigabyte video memory
- Core i5 2.5 ghz cpu
- 8 gigabytes of ram
- Windows 7/8/8.1/10 64 bit Operating System

Recommended System Requirements to run Falling at 1440p or 4k

- Nvidia Geforce GTX 1060 or AMD RX 480
- 6 Gigabyte video memory
- Core i5 3.5 ghz cpu



- 16 gigabytes of ram
- Windows 10 64 bit Operating System

4.7.2 Draw Call optimization: Each Soldier has 12 UV sets and Unreal Engine puts 5 draw calls per uv set with all the lights and shadow enabled whether baked or in realtime. So for each soldier I have 60 draw calls on my screen. I tried to maintain a moderate amount of soldiers on the screen per frame so that the game's quality isn't compromised while it runs at the desired 60 frames per second.

4.7.3 Polycount and LOD optimization: The polycount of characters and fort assets didn't cause any issue for optimization. But I had to place the LOD for every character, gun, grass, foliages and environment asset very carefully. The grass and foliages are set up in a way so that when we look at them from distance they appear as billboards. For characters, gun and environment assets LOD varies from higher to lower polycount depending on the draw distance.

4.7.4 Instancing Foliages Procedurally: Due to using procedural foliage simulation tool I could create lots of foliages with variations in any place of the terrain. The procedural foliage simulation not only allows to create dense foliages quickly and procedurally but also it is very performance friendly. The grass was generated entirely using landscape material so there was not much to worry about performances. Also I've used LOD for both grass and foliages which improved the overall performance even more. If I had to manually scatter foliages and grass it would take a lot more time and it would create a lot of problems to overcome for achieving desired level of game performance.

4.7.5 Texture Resolution: While making assets and textures I had to bare in mind that I need to manage less textures possible with all of them having 2k and some having a maximum of 4k resolution. With less textures being repeated I would require less memory and the game will be very friendly on ram. A lot of objects are being repeated in the game so they contain the same texture maps and same material. The characters, gun, Arcot fort assets are comprised of 4k texture maps. Timari fort and some other assets come with textures in 2k resolution.

4.7.6 Collider Meshes: I needed to build colliders in maya for every single environment assets and gun object. Collider is a must for making games in any game engine. Complex colliders are never the way to do it because they come with a high cost in performance

even if they give more precision. I tried to have box colliders for most assets. The only assets that don't have box colliders are equipped with ramp shaped colliders considered as complex collision. Having box colliders I could set the collision method to simple collision for most assets that does the job of making my traversable areas playable as well as gives me a lot of performance gain.

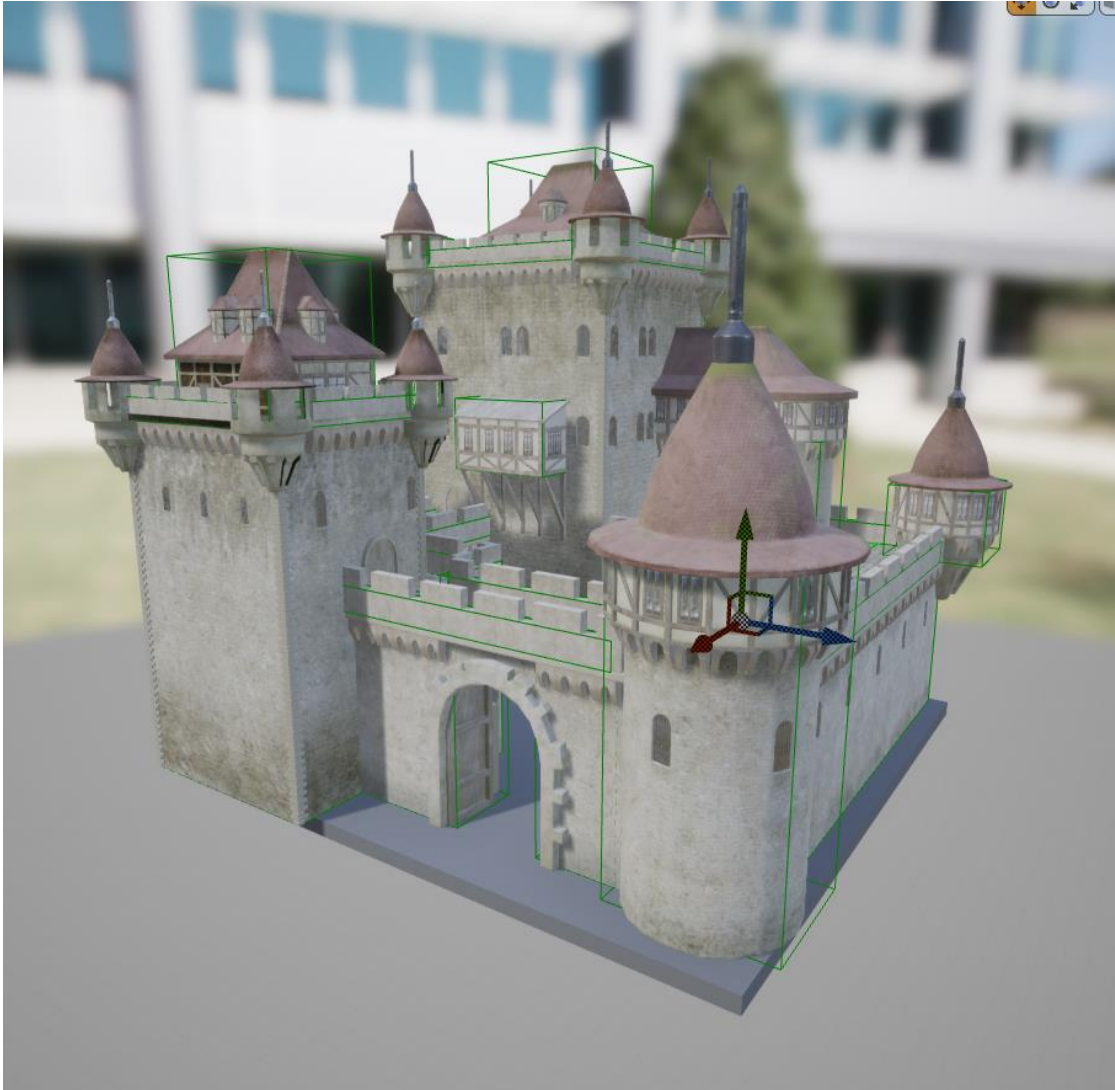


Figure 106: Box colliders for Arcot Fort

4.7.7 Performance Test: After all the aforementioned process being done I was able to get a playable build that I could test in two different computers.

First computer is equipped with an intel core i5 4590 cpu, 8 gigs of Transcend 1600mhz ddr3 ram, Gigabyte Z97 Gaming Sniper motherboard and Asus Nvidia Geforce GTX



1060 6 gigabyte GPU. The average performance data from MSI Afterburner monitoring goes by the following:

GPU usage: 83%

GPU temperature: 65 degree celcius

GPU memory consumed: 3212 Megabyte

CPU usage: 45%

CPU temperature: 60 degree celcius

RAM consumed: 5913 Megabyte

Average frame rate per second: 120

Second computer came with the same intel core i5 4590 cpu, 8 gigs of TwinMOS 1600mhz ddr3 ram, 8 gigs of G Skill Trident X 2400mhz ddr3 ram(Total 16 gigs of 1600mhz ddr3 memory), Gigabyte Z97 Gaming 3 motherboard and Gigabyte Nvidia Geforce GTX 970 4 gigabyte GPU. The average performance data are given below.

GPU usage: 85%

GPU temperature: 76 degree celcius

GPU memory consumed: 3142 Megabyte

CPU usage: 54%

CPU temperature: 79 degree celcius

RAM consumed: 5817 Megabyte

Average frame rate per second: 62

It is clearly noticeable that the use of dynamic lighting has been really efficient for memory and cpu. Even though the game is utilizing the power of GPU it is still very performance friendly for available mid range graphics cards in the market.

## 4.8 Soldier and Weapon Functionalities:

The overall event graph of British Soldier's mechanics look like as of the following

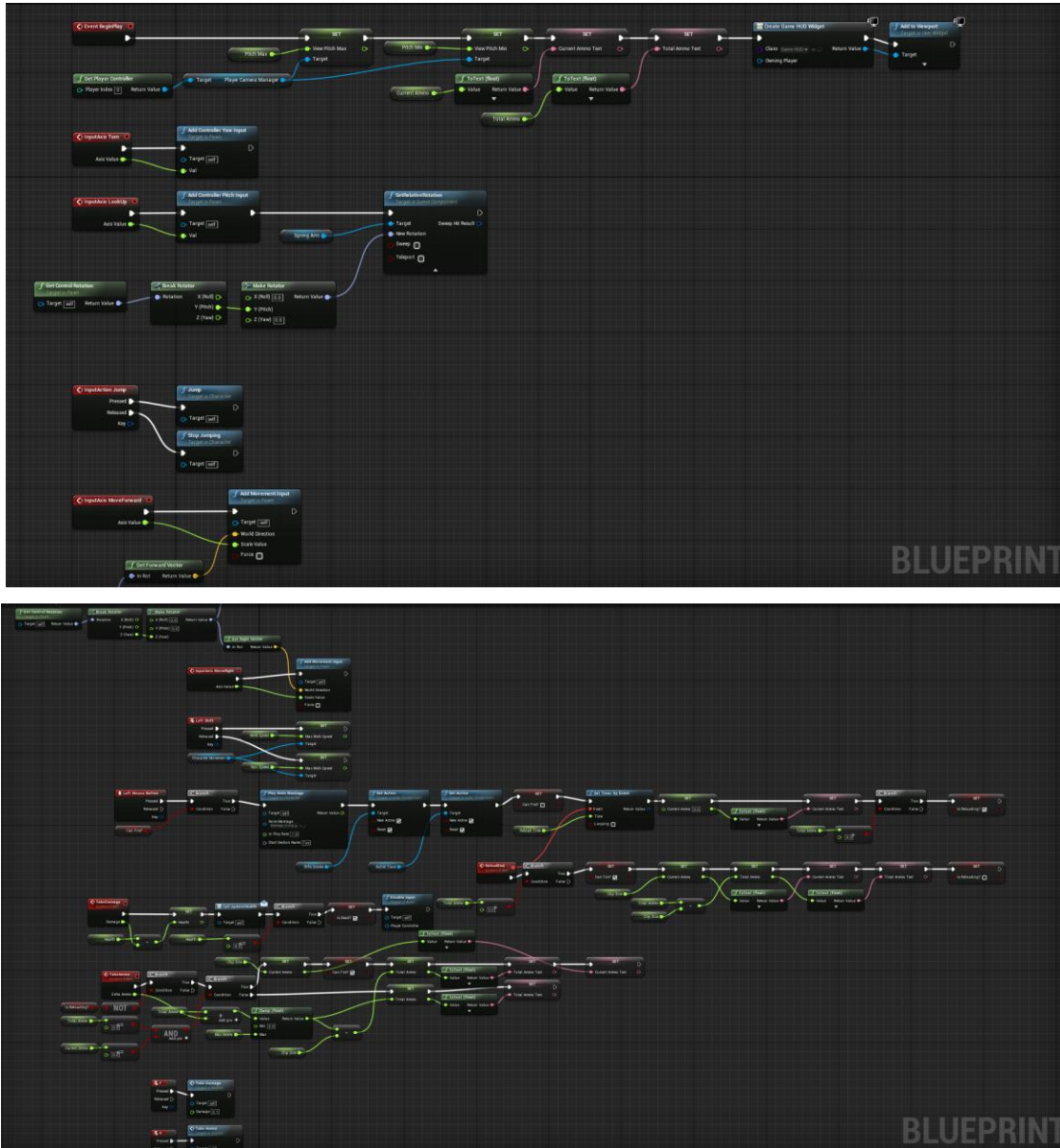


Figure 107: British Soldier Blueprint

The animation blueprint is for controlling all the animations but this blueprint is responsible for gameplay mechanics of my playable character as well as the weapon. This is where the fire, reload and ammo count are being controlled from. We can also call this





the backend of what's happening inside my game. I set up two different particles for the firing of my rifle. One is the bullet spark and another one is a smoke trail.

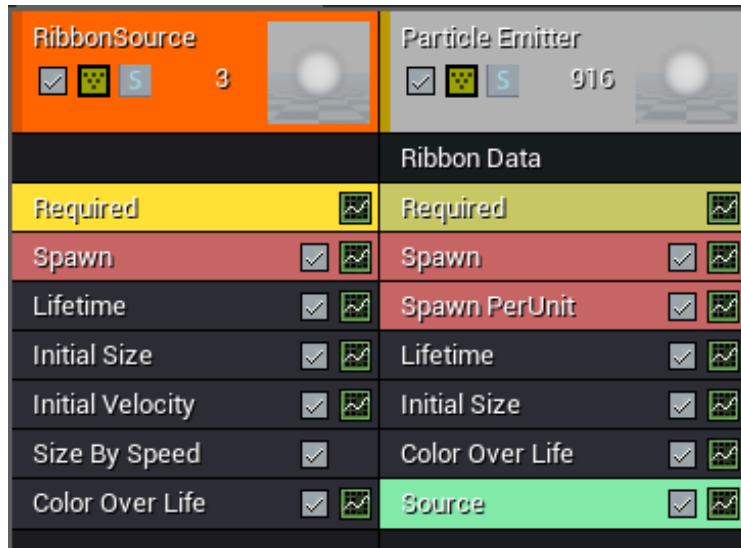


Figure 108: Sparks Particle controls

The Ribbon data is responsible for creating the spark trail. Lifetime determines how long it's going to stay on the screen after emission. Initial size and Velocity determine the size and velocity right after emission. Size by speed is for controlling the max scale and speed scale parameters. Color over life allows to change the color of trail. Here I have been creating only one emission at a time. The spark emits and disappears to be reset and emit the next spark when fire button is pressed.

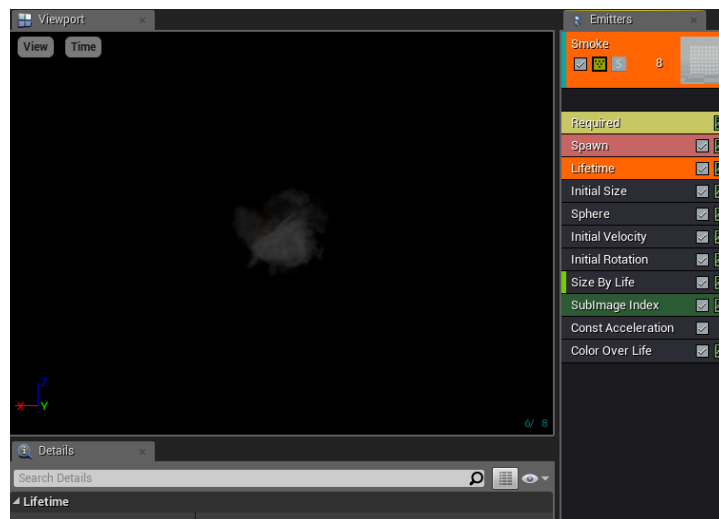


Figure 109: Smoke Particle controls

The smoke particle has similar settings as of the sparks. It emits right after the spark emission every time fire button is pressed. The emissions are also controlled from the soldier's functionality blueprint.

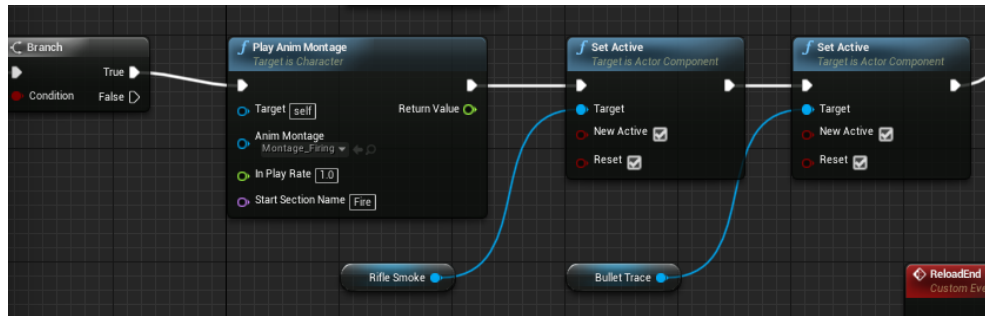


Figure 110: Rifle Smoke and Bullet Trace in Blueprint

#### 4.9 Game Hud:

To create the game hud first I created a widget blueprint from user interface of create advanced asset options. This gave me a widget blueprint along with a designer window. The designer window is where I can create the outlook of my hud. I put a progress bar for health, health icon, gun icon, crosshair, objective icon and texts necessary to update objectives and ammo count. Texts for current ammo count and total ammo count when using the weapon and reloading are controlled from my soldier's functionality blueprint. When total ammo count is 0.0 my character will start to reload the weapon and after reloading ammo count will be filled again so he can fire.



Figure 111: GameHUD widget designer window



The progress bar that shows the character's health is controlled from the widget blueprint. Here I have variables for Crosshair, GunIcon, Healthbar, HealthIcon, ObjectiveIcon and Character Reference.

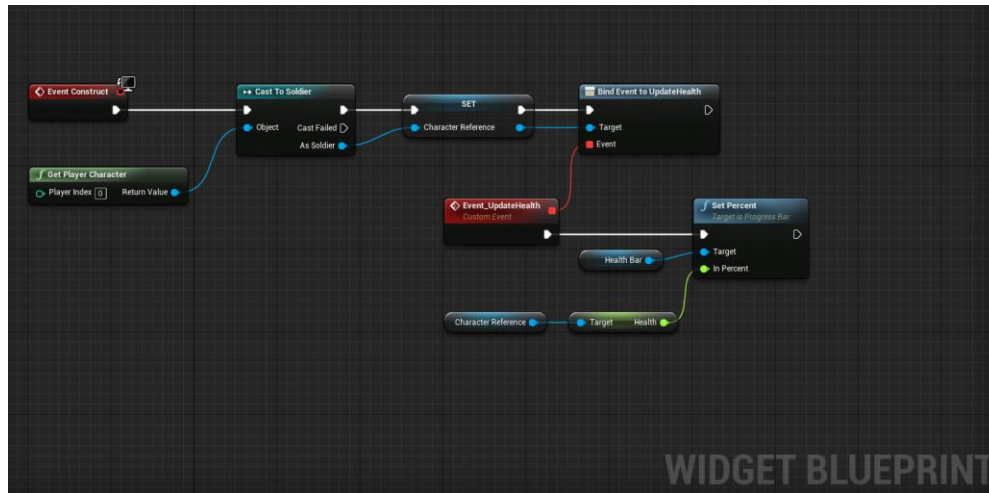


Figure 112: GameHUD widget blueprint

#### 4.10 Challenges:

I have faced challenges in every single step and that is the reason I took on this project. To face challenges, to overcome them and eventually end up in something I can be proud of and most importantly I wanted to make a huge leap in terms of my growth as a 3d artist for the video game industry. The biggest challenges came in Weight Painting process and making the animations of mixamo work properly with my characters. Replicating the landscapes of Timiri and Arcot was another challenge. Though World Machine is a really powerful tool it was challenging to create the erosions and heights properly to make the terrain look visually appealing and not so far from the actual location of India. Time management was another challenge as almost every task in the pipeline took longer than I thought and planned my workflow upon. Due to problems arising in both technical and creative ends I needed extra time for brute forcing and finding out workarounds or solutions. So in order to get a functional build of my game ready within the deadline I had to compromise and manage a lot of things. Even though Marvelous Designer is a very efficient cloth simulation tool and it gives us a faster workflow to create realistic costumes for characters than with regular way of digital sculpting it requires some research and study to be familiar with the software itself. The software is different from any other 3d software we use. It is also required to sculpt



further details in zbrush after the cloth is created in marvelous. The complete Maya-Marvelous-Zbrush workflow requires some other housekeeping and technical works too in order to get a visually appealing costume ready for any game character. Creating Clive's costume with Marvelous I faced challenges and even though I overcame them I still need more study with this workflow before I can start using it frequently. So eventually it took me longer than what it would have taken if I sculpted the entire costume in zbrush whereas it was supposed to take less time. Picking up Unreal's material editor and it's node based workflow had proved to be challenging for a creative artist like me. It is technical yet efficient to really enhance textures and it has controls to do a lot of things with texture maps. To learn working with this vast editor was a bit difficult at first. With more knowledge on it I could make my textures look better resulting in overall visual of the game more appealing.

#### 4.11 Limitation:

The biggest limitation I had was in terms of animation. As a 3d artist specializing in sculpting and modelling, I don't have the skills of a keyframe animator to make realistic animations of the character within a tight time budget. With manual keyframe animations, the movement of the character and overall gameplay would've felt better and more responsive. Another limitation was my lack of time estimation skill. It is evident from the output of my build that with more time this project could bring out a lot more from it's potential. But it's also true with better time estimation I could plan out the steps of my workflow and overall pipeline a lot better during the concepting phase. That would ultimately result in a more efficient pipeline. The workflow included a lot more study and research to find out improvements rather than just implementation of existing skills. So it was evident that I would find the time constraint difficult if my project scope is too big. Ultimately all that matters is the output because we know "The end crowns the work" so it is a must to end up in something which gives value to the labour that produced it.

## CHAPTER 5

### Preview of The Game & Assets

#### 5.1 In Game Screenshots

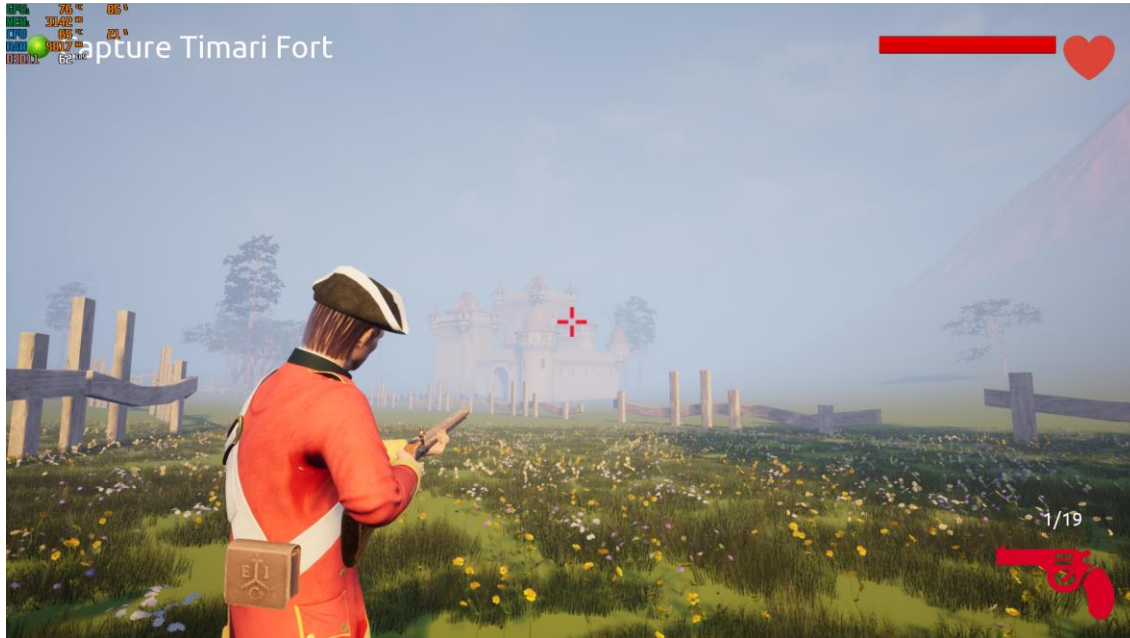


Figure 113: In Game Screenshot of Arcot Fort Front

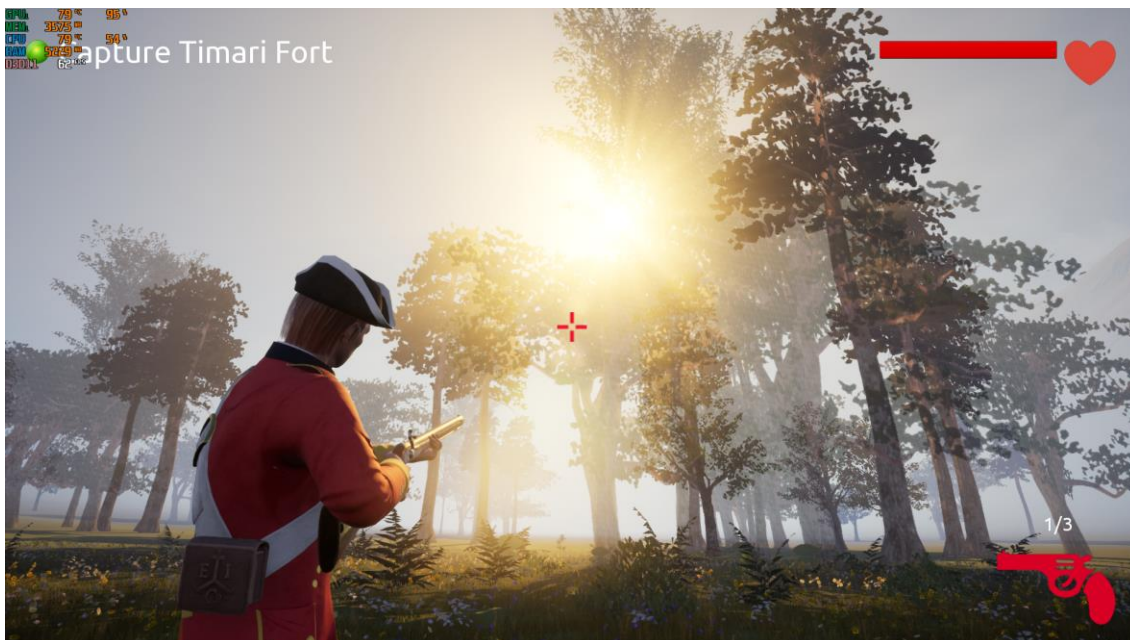


Figure 114: In Game Screenshot of Arcot Terrain Trees

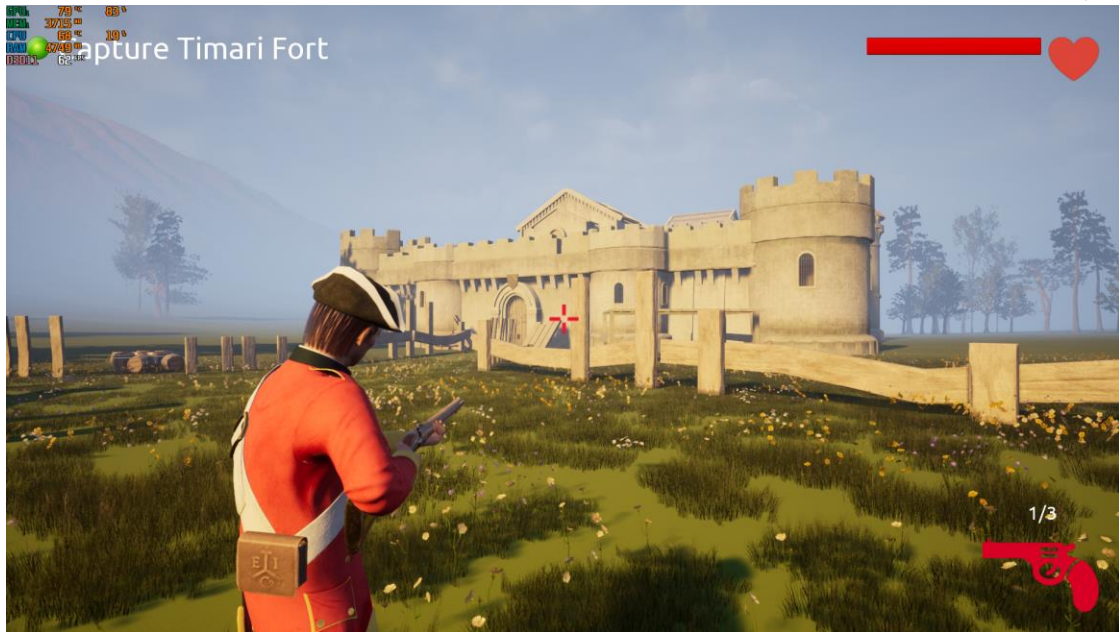


Figure 115: In Game Screenshot of Timari Fort Front



Figure 116: In Game Screenshot of Timari Terrain grass & foliage

## 5.2 Characters Presentation

### 5.2.1 Robert Clive



Figure 117: Robert Clive presentation Marmoset



Figure 118: Robert Clive front



Figure 119: Robert Clive left



Figure 120: Robert Clive right





Figure 121: Robert Clive back



Figure 122: Robert Clive Posed

## 5.2.2 British Soldier



Figure 123: British Soldier Posed



Figure 124: British Soldier Front



Figure 125: British Soldier right



Figure 126: British Soldier left

### 5.2.3 French Soldier



Figure 127: French Soldier Presentation



Figure 128: French Soldier Left



Figure 129: French Soldier right



Figure 130: French Soldier front

### 5.3 Weapon Presentation



Figure 131: Enfield Snider front



Figure 132: Enfield Snider back



Figure 133: Enfield Snider butt



Figure 134: Enfield Snider muzzle

## 5.4 Forts Presentation

### 5.4.1 Arcot Fort



Figure 135: Arcot Fort wide shot



Figure 136: Arcot Fort mid shot



Figure 137: Arcot Fort rear shot



## 5.4.2 Timari Fort



Figure 138: Timari Fort wide shot



Figure 139: Timari Fort side shot



Figure 140: Timari Fort rear long shot



Figure 141: Timari Fort overview

## 5.5 Terrain Presentation

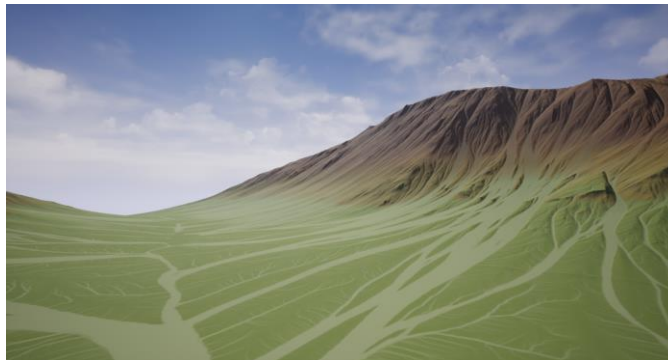


Figure 142: Timari Terrain view from the ground



Figure 143: Arcot Terrain view from the ground



## CHAPTER 6

### Discussion

The video game industry has been expanding since 2000s. The Third Person Action genre is getting widely popular for not only shooters but also games that include sword fights, hack and slash, beat em' up and puzzle solving. RPG games take this genre to a whole new level with any kind of gameplay. Even the turn based RPG games nowadays combine the third person gameplay with the regular combat system using turns(e.g: Dragon Age Inquisition, Final Fantasy XV) After 2010 the graphics of video games have been making a leap every couple of years with the use of DX11 and now DX12 features, increasingly popular PBR workflow, advancement of game engines, advancement of shaders within the engines, the affordable yet very powerful consumer level hardware and 3d software packages allowing artists more freedom and ease to translate their imagination to reality. I tried to strictly follow the industry standard workflows in the design, art and development processes while maintaining authenticity of my game that offers something new to the gamers be it in terms of story or visuals. Making an open world featuring the landscapes of Arcot and Timiri through two levels of gameplay happened by taking inspirations from many open world games but mostly Horizon Zero Dawn, Dragon Age Inquisition and The Witcher 3. Where does it stand compared to them? The composition of the world, the color palette, characters and weapons are based on the novel and concept art found on the web on the timeline of the actual event but in terms of overall style and many technical aspects we can find out a lot of similarities to the aforementioned titles. Some exclusive modern features used by current generation AAA games have been used in my game and they have made life a lot easier for me.

6.1 Frustum Culling: In computer graphics the viewing frustum refers to the field of view of the camera which is the region of space in the world that appears in the camera to the player. Frustum Culling is a technique to cull everything that is not inside of the frustum so only the things that are seen in the frustum are rendered at a time. This way the performance of the game increases a lot. This technique is used in Horizon Zero Dawn and taking advantage of this I've managed to optimize the overall performance of my

game in runtime. Because Unreal Engine has this feature by default I didn't have to do anything to build it up.

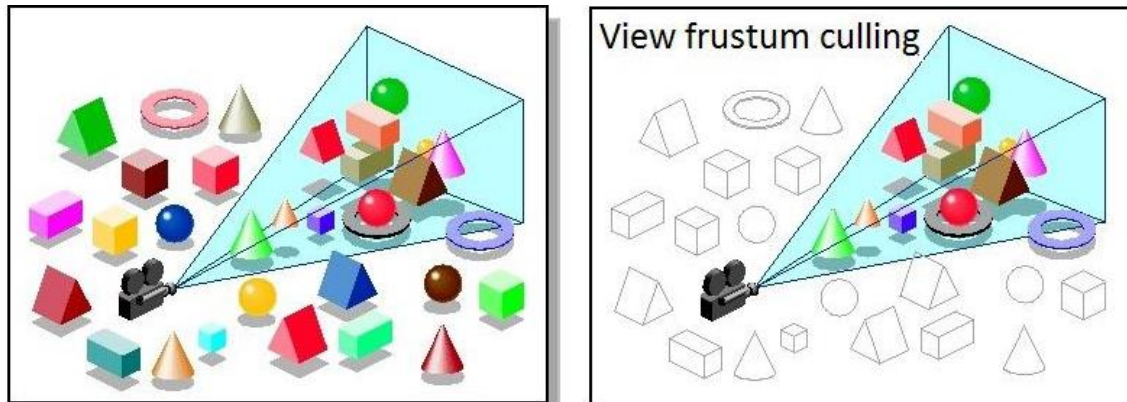


Figure 144: Without frustum culling (left) and with frustum culling(right)



Figure 145: Frustum culling of Horizon Zero Dawn

6.2 PBR Shaders: The games I have taken inspirations from have used PBR Shaders to achieve realism in the models. My game is no different. Every single 3d model is equipped with PBR materials.

6.3 Mocap data for animation: The increasing competition has seen game developers go for motion capture in order to create most animation not only to save time but also to bring more life to movements so the players find the games more responsive and they feel like they are living inside the game. The animations for characters used in my game



have been taken from Mixamo's motion capture library and heavily edited to match my characters and weapon to achieve the desired level of realism I aimed for.

6.4 Dynamic Lighting: Even though completely baked lighting or a mixture of moveable light and baked lights see games run better as most shadows and lights are not baked in realtime the advancement dynamic light inside Unreal has made the lighting process easier and more representative of the real world light and shadows. Fortnite is completely made with Unreal's dynamic lighting. I lit the world of my game with the same procedure. It utilizes the GPU a bit more but gives very impressive performance while not being much ram intensive as with the baked light since baked data takes a lot of memory.

The game as a whole stands on its defining art style. But the collective inspirations for both art and technical aspects from top triple A titles of the same genre make it a complete package.



## CHAPTER 7

### Conclusion

This project in overall has achieved a lot of things and it's not just a regular history lesson for gamers. Since 2010 when I was in the last year of high school I have always dreamt of working in the video game industry. Since 2014 the dream became more specific to the 3D department of the modern games. Not only I wanted to create a beautiful game with technically advanced 3d models but also I wanted to make a complete authentic game that offers a uniqueness in terms of the overall story, theme and art style. Writing this paper is a dream come true because this paper breaks down all the process behind the game, the motivation, the thoughts and all the other things that went on to complete the project. The paper is a sum of all the research, technical and creative work as well as overall improvement needed for me as an artist in order to create a game. The anticipated application of this game could come as a complete game in Steam store and other marketplaces for PC and consoles. It can be shipped either as a complete open world game or an episode based game through a season like Hitman 2016 or The Walking Dead series by Telltale Games. The video game industry will keep growing and I have to keep growing as an artist as well as I have to keep myself updated with all the new software and tools. The competition is fierce and every year the games are looking better, they are offering new contents, innovative features and so on. My initiative is a small drop in the big ocean but I want it to contribute in the world of video games as an authentic product and act as an inspiration for other indie game developers who are aspiring to grow and create. With all the technical tools available anyone with artistic and technical prowess can create lots of new things from their imagination. Video game is an impressive medium for that.



## References

- [1] Rudyard Kipling, G. A. Henty, H. Rider Haggard, British Empire Adventure Stories, Prion, With Clive in India; Chapter 7 The Siege of Arcot, pp. 526-536, 2008
- [2] Likeness Sculpting with Frank Tzeng  
[https://gumroad.com/frank\\_tzeng\\_art](https://gumroad.com/frank_tzeng_art)(Accessed on 20th December 2017)
- [3] Creating Hyper Realistic Characters in Zbrush with Hossein Diba  
<https://www.thegnomonworkshop.com/tutorials/creating-hyper-realistic-characters-in-zbrush>(Accessed on 15th January 2018)
- [4] Uldis Zarins, Sandis Kondrats, Anatomy For Sculptors: Understanding the Human Form, Exonix LLC, 2014
- [5] Character Modeling for Production: Maximize your workflow for production with Krystal Sae Eua  
<https://www.thegnomonworkshop.com/tutorials/character-modeling-for-production-maximizing-your-workflow-for-deadlines-with-krystal-sae-eua>(Accessed on 10th March 2018)
- [6] PBR Character for Games by Marc Brunet  
<https://cubebush.co/mb/products/1ztdpw/pbr-character-for-games>(Accessed on 17th January 2018)
- [7] Marvelous Designer Fundamentals by Natalija Jagarinec  
<https://www.pluralsight.com/courses/marvelous-designer-fundamentals>(Accessed on 15th March 2018)
- [8] Creating Men's Formal Wear with Marvelous Designer by Natalija Jagarinec  
<https://www.pluralsight.com/courses/marvelous-designer-creating-mens-formal-wear>(Accessed on 18th March 2018)
- [9] Substance in UE4: Sci-fi Crate Project  
[https://tutorials.allegorithmic.com/courses/substance-in-ue4-sci-fi-crate-project/youtube-V24Fe6F\\_jFA](https://tutorials.allegorithmic.com/courses/substance-in-ue4-sci-fi-crate-project/youtube-V24Fe6F_jFA)(Accessed on 5th April 2018)
- [10] Understanding PBR Workflows  
<https://tutorials.allegorithmic.com/courses/understanding-pbr-workflows/youtube-ueC2qGzWrgQ>(Accessed on 1st April 2018)
- [11] Fundamentals of Baking Textures  
<https://tutorials.allegorithmic.com/courses/fundamentals-of-baking-textures/youtube-ePnLTuzRABg>(Accessed on 1st April 2018)
- [12] Substance Painter Model Preparation  
<https://tutorials.allegorithmic.com/courses/substance-painter-model-preparation/youtube-j5isGX8zwSI>(Accessed on 2nd April 2018)
- [13] Understanding Normal Map tangents and Shading  
<https://tutorials.allegorithmic.com/courses/understanding-normal-map-tangents-and-shading/youtube-gR3r7Xmhmlk>(Accessed on 2nd April 2018)
- [14] Understanding UVs and texel density



- <https://tutorials.allegorithmic.com/courses/understanding-uvs-and-textel-density/youtube-bwusznnkRxo>(Accessed on 2nd April 2018)
- [15] Getting Started with Substance Painter 2017  
[https://tutorials.allegorithmic.com/courses/getting-started-with-substance-painter-2017/youtube-Qk\\_YRkDsJrQ](https://tutorials.allegorithmic.com/courses/getting-started-with-substance-painter-2017/youtube-Qk_YRkDsJrQ)(Accessed on 3rd April 2018)
- [16] Ultimate Weapon Tutorial Complete Edition by Tim Bergholz- ChamferZone  
<https://gumroad.com/timb>(Accessed on 5th May 2018)
- [17] Environment Modeling and Sculpting for Game Production with Nate Stephens  
<https://www.thegnomonworkshop.com/tutorials/environment-modeling-and-sculpting-for-game-production>(Accessed on 3rd June 2018)
- [18] GDC 2017: Face Texturing in Substance Painter w/Magdalena Dadela  
<https://tutorials.allegorithmic.com/courses/gdc-2017-presentations/youtube-9ywhVkZBtq8>(Accessed on 26th April 2018)
- [19] Getting Started with Substance Painter 2018  
<https://tutorials.allegorithmic.com/courses/getting-started-with-substance-painter-2018/youtube-IGGQI9kVB1M>(Accessed on 15th August 2018)
- [20] Digital Tutors: Building a Landscape from the Ground up in World Machine by Dylan Yarbrough  
<https://www.pluralsight.com/courses/building-landscape-from-ground-up-world-machine-2313>(Accessed on 2nd September 2018)
- [21] Unreal Engine Documentation  
<https://docs.unrealengine.com/en-us/>(Accessed on 25th September 2018)
- [22] Getting Started with UE4  
[https://www.youtube.com/user/UnrealDevelopmentKit/playlists?shelf\\_id=17&view=50&sort=dd](https://www.youtube.com/user/UnrealDevelopmentKit/playlists?shelf_id=17&view=50&sort=dd)(Accessed on 7th July 2018)
- [23] Unreal Engine Live Training: Getting Started with Landscape  
<https://www.youtube.com/watch?v=gMKjIZMPJ0Q&t=4333s>(Accessed on 15th July 2018)
- [24] Unreal Engine Live Training: Getting Started with Landscapes Material and Foliage  
<https://www.youtube.com/watch?v=PDIGKZ1c3Zc>(Accessed on 20th July 2018)





## Bibliography

Quixel Megascans: <https://quixel.com/megascans>

Textures.com: <https://www.textures.com/>

Poliigon: <https://www.poliigon.com/>

FlippedNormals: <https://flippednormals.com/>

The Gnomon Workshop: <https://www.thegnomonworkshop.com/>

Pinterest: <https://www.pinterest.com/>

OpenTopography: <https://opentopography.org/>