International Journal of Computer Applications (0975 – 8887)
Volume 156 – No 3, December 2016

Virtual Reality Content Creation using Unity 3D and Blender

Rajat Gupta
V.E.S. Institute of Technology
University of Mumbai

Rohan Nawani
V.E.S. Institute of Technology
University of Mumbai

Vishal P. Talreja
V.E.S. Institute of Technology
University of Mumbai

ABSTRACT

‘Room Escape VR’ is an interactive virtual reality game developed using Unity3D game engine, Blender, and Photoshop. The game is being built for Android and iOS smartphones that support Google’s Cardboard and Daydream SDK. A game development lifecycle explaining the production of the game from concept phase to its release is presented in this paper. Looking at the current trends of technology, the ever adaptive and ever increasing gaming industry is becoming more demanding when it comes to terms of realism. The idea behind ‘Room Escape VR’ was to make an immersive VR experience for the smartphone like no other. Virtual Reality devices like the Oculus Rift, HTC Vive, and the PlayStation VR have started offering such experiences, but these kinds of devices are out of the reach of the common man who cannot afford the initial price tag.They then resort to the more affordable experiences offered on the mobile environment by companies like Google.

Keywords

Virtual Reality, Google Cardboard, Content Creation, Game development lifecycle, Unity 3D game engine, Blender, Adobe Photoshop

1. INTRODUCTION

There is solid evidence that virtual reality's benefits have graduated from impressive visual demonstrations to producing results in practical applications. [1]The idea behind ‘Room Escape VR’ was to make an immersive VR experience for the smartphone like no other. Virtual Reality devices like the Oculus Rift[2], HTC Vive[3] and the PlayStation VR[4] have started offering such experiences, but these kinds of devices are out of the reach of the common man who cannot afford the initial price tag. They then resort to the more affordable experiences offered in the mobile environment by companies like Google[5]. There are many VR games for smartphones in the market, but none of them are interactive enough or even aesthetically pleasing. A major drawback of making games in VR on smartphones is that the smartphones have limited processing power and the VR games require double the processing power of a regular game as two cameras are used to render the virtual world. Thus we need to optimize the graphics to the extent that the game can run smoothly on a phone with the most basic processing power.

A game is a kind of software with thegoal to provide entertainment. However, during the real game development practice, simply adopting the software development life cycle (SDLC) is not enough, as the developers face several challenges during its life cycle. To address the problem, game development uses a kind of specific approach called game development life cycle (GDLC) to direct the game development[6].

Unity 3D 5.4 with native Daydream support was used as the game engine, but theprimary focus of the project was placed on 3D content creation for real-time application, using, besides other tools, Blender. Also, Google’s Cardboard and Daydream SDK[7] was used to assist while building the game. Every member of the team was given at least one role. The resulting game ‘Room Escape VR’ is a hacker-themed room escape game, where the player has to solve multiple puzzles to escape the room.

Figure 1 shows an example screenshot of the game.

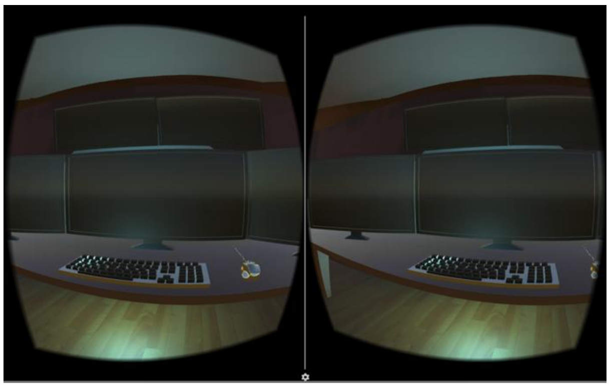


Figure 1: Example screenshot of the game.

2. TOOLS

The maintools used throughout this project were mainly, Blender[8] for modeling, animation and rigging and Unity 3D game engine[9] for implementation. Also, imageediting tool, Adobe Photoshop was used to create textures.

2.1 3D Modeling and Animation Software (Blender)

Blender was chosen as the primary tool for all 3D. It is a free and open source software used for modeling 3D objects, UV mapping, sculpting, rendering, animation, rigging, etc. Blender provides an end-to-end creative workflow that makes it easier for the artists to create complete working models for the programmers.

2.2 Game Engine (Unity 3D 5.4)

In this project, the free version of the game engine Unity3D is being used for the production of ‘Room Escape VR’. Unity is best suited for small and middle-sized development studios who cannot afford to invest in expensive high-end rendering engines.

The primary reason for choosing Unity was the fact that it is very easy to use and to learn. Developinggames with Unity is mainly based on a drag and drop functionality with the occasional adapting of scripts rather than writing code.Also, Unity provides multiple built-in shaders and effects as well as a physics engine and collision detection. Apart from shaders and effects, Unity provides various scripts mostly written in C# or Unity’s JavaScript like script called UnityScript which can be added onto the 3D models. These scripts provide the models with the basic functionality andcan be used for example for purposes like character controlling, defining actions of a rigid body object, etc.One of the drawbacks of using Unity is that it lacks forward compatibility, i.e., it cannot accept inputs from a version other than which project is being developed on. Unity also lacks integrated modeling abilities, which is the reason for using Blender as anexternalmodeling tool.

3. GENERAL GAME PRODUCTION PIPELINE

An idea of the workflow management that isused in the game development process is known as the game production pipeline. The stages of this pipeline are certain tasks or duties that need to be fulfilled. Some of these tasks are required to be carried out sequentially. However, to increase productivity, one should break up this sequential approach, and allow people to work in parallel on different tasks simultaneously.

Figure 2 gives a detailed overview of the game development lifecycle of this project.

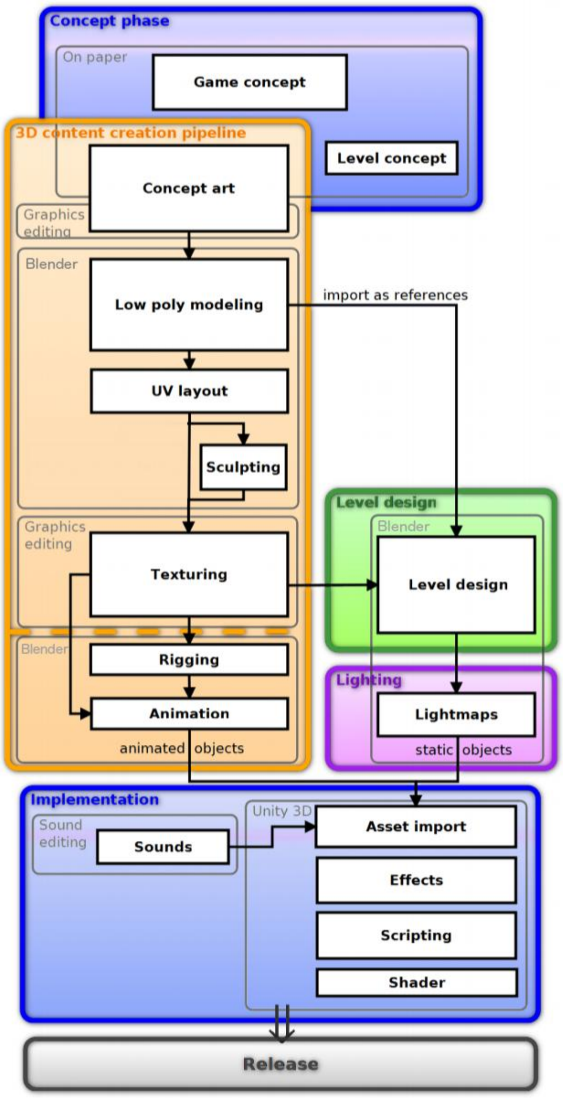


Figure 2: Game Production Pipeline Overview.

3.1 Concept Phase

During the concept phase, every aspect of the game is to be thought of meticulously like, the theme of game, whether or not the game will be 3D, the target audience of game, whether the game engine used would be Unity or Unreal [10] or anything else, the target platform, etc. All the decisions made were agreed upon after a brainstorming session, and a final plot of the game was made up. The theme chosen for 'Room Escape VR' is hacking and the game plot involves a protagonist who finds his way out of an abandoned building, if he can solve all the clues hidden in the high-tech apartments, by the antagonist who has abducted him.To make players avoid acollisionwith the obstacles in their way while playing 'Room Escape VR', the player need not necessarily be walking or moving from his/her position. They use a controller to move left, right, forward, backward or to jump.

After this phase, the look and feel of the game content need to be worked out which requires multiple revisions of concept arts. Inspiration is taken from all different sorts of media, starting with similar computer games and movies to comics and music. A dirty textured comic look was chosen, inspired by the game 'No Man's Sky' [11].

3.2 3D Content Creation Pipeline

Part of Figure 2 illustrates the 3D content creation process which contains the stages that every 3D model must pass through to transform from concept art to the final model. The following paragraphs describe these steps in detail.

3.2.1 Concept Art

After the initial sketches on the paper, once we were satisfied with the look of the main characters, we straight away moved on to create the 3D model in Blender. By using the initial sketches with the front and side view of the character, the character model was created.

3.2.2 Low Polygon Modeling

Low polygonmodel is a polygon mesh with arelatively small number of polygons. These polygons are the basic building unit of models. Low poly meshes are used in games and high poly ones, on the other hand, are made use of in animated movies.

Another reason for using low polygon modeling is the limited processing power of smartphones currently being used by people worldwide. Therefore, it should be made sure that the game is playable and doesn't useexcess RAM and does not cause the phone to hang while playing. For that to happen, models are made cartoony style at a minimal level by keeping the polygon count and detailing as low as possible.

Polygon modeling involves primitive geometric shapes such as quadrilaterals, triangles, etc. which make up a whole object.Since Unity3D can handle both triangles and quadrilaterals with relative ease, there was no restriction on the artist to use a certain modeling technique or primitive only. Nevertheless, most modeling artists preferred usingquadrilaterals over triangles.

Low polygon modeling also includes processes such as smoothing or hardening normals. To avoid models with very varying levels of detail, it is highly recommended that the artist defines a maximum polygon count for the model in advance depending on the model's size and importance

Figure 3 shows a simple Low Polygon model being built in Blender

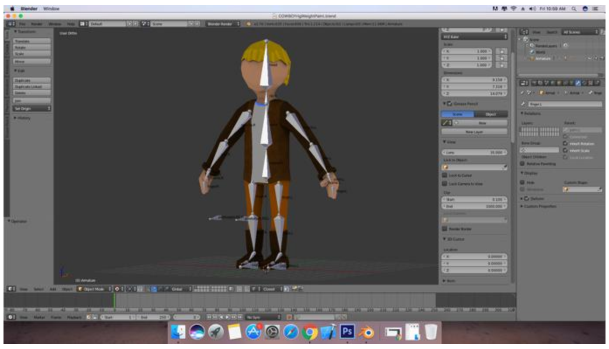


Figure 3: Low Polygon Model in Blender.

3.2.3 UV Layout and Texturing

The letters 'U' and 'V' stand for the axes of the 2D texture because 'X', 'Y' and 'Z' are already used to denote the axes of the 3D object in model space. UV texturing permits the polygons that make up a 3D object to be painted with color from an image. [12] All models built in Blender were mapped in UV so that they could be painted using textures in Adobe Photoshop CS6.

Texturing is the essence of realism. A realistic material needs Bump maps, Reflection maps and Displacement maps available in Blender.The downside of adding too much realism is that it increases the time required to render the scene.[13]

Figure 4 shows the UV layout and Texture Map for the 3D model in Blender.

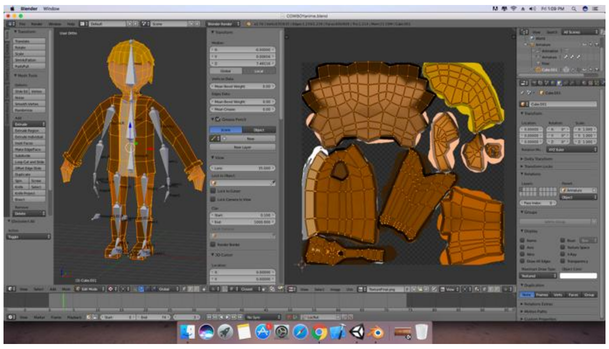


Figure 4: UV Layout and Texture Map for 3D model.

3.2.4 Rigging and Animation

Rigging is setting up a virtual skeleton that moves the mesh in Blender to create natural looking animations. The skeleton is called an armature. Each bone can have some vertices assigned to it, and when you move a bone, just the vertices assigned to it will move.[14]

Figure 5 shows the Rigging of the Low Polygon Model.

To target the particular vertices to be moved by a specific bone we use a technique called as weight painting. The process of weight paintingis explained using the following example.Imagine two bones, Bone 1 and Bone 2, with the same vertices assigned to each of them. Now assume you tell Bone 1 to go one way, and Bone 2 to go the other way. Which bone do the vertices follow?

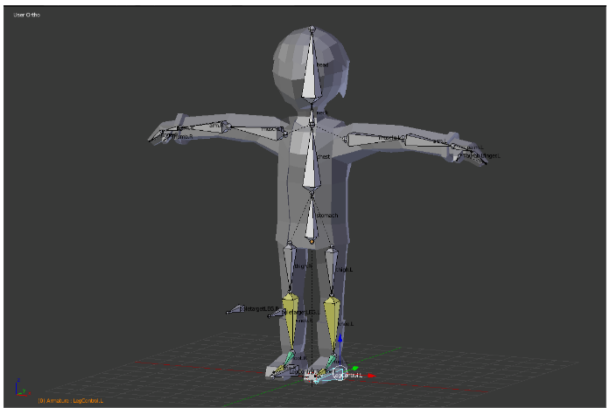


Figure 5: Rigging of the Low Polygon Model in Blender.

The answer depends on the weight of the vertices for each bone. By default, vertices assigned to two different bones have a weight of 50% each bone. The weight on the vertices will be 50% Bone 1 and 50% Bone 2. This means that when you move Bone 1 one way and Bone 2 another way, the vertices will try to split the difference. That is, follow Bone 1 50% and Bone 2 50%.

If you wanted those vertices to follow Bone 1 more closely than they follow Bone 2 then you would weigh the vertices to be, say, 75% Bone 1 and 25% Bone 2. The easiest way to assign weights to vertices is in Weight Paint Mode.

To get into Weight Paint Mode, the shortcut is Ctrl⇆ Tab, analogous to how you got into Pose Mode for an armature. The first thing you notice is that the mesh turns dark blue. The color is important in weight painting! It tells you what weight the vertices have for a given bone. Dark Blue = 0; Light Blue = 25% (or 1/4); Green = 50% (or 1/2); Yellow = 75% (or 3/4); Red = 100% (or 1).
Unity has a rich and sophisticated animation system sometimes referred to as 'Mecanim' which provides easy workflow and setup of animations for all elements of Unity including objects, characters, and properties with layering and masking features.[15]

Figure 6 shows the setting up of the animation of the 3D model in Blender

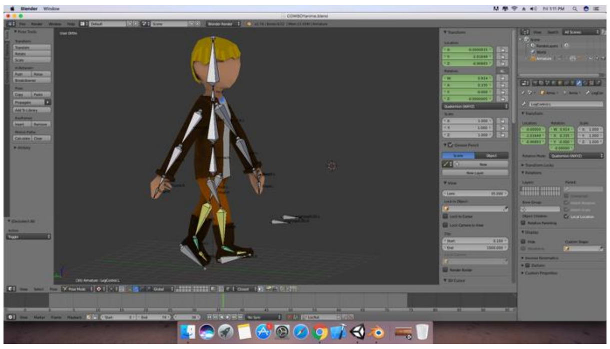
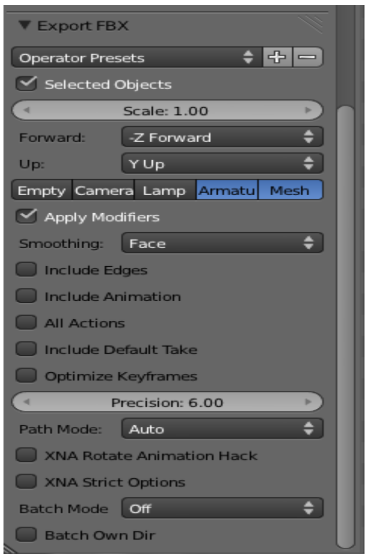


Figure 6: Animation of 3D model in Blender.

3.3 Level Design

Once, we have finished modeling we need to export the model to Unity. To simplify the export and the import into Unity 3D, we remove the lamp and the camera from the Blender scene.We export the Blender files under the FBX [16] format because it's one of the input formats of Unity 3D along with 3DS and other ones.[17]

Figure 7 shows the FBX Export setting from Blender



▼ Export FBX
Operator Presets
✓ Selected Objects
Scale: 1.00
Forward: -Z Forward
Up: Y Up
Empty Camera Lamp [✓ Armatu] [✓ Mesh]
✓ Apply Modifiers
Smoothing: Face
☐ Include Edges
☐ Include Animation
☐ All Actions
☐ Include Default Take
☐ Optimize Keyframes
Precision: 6.00
Path Mode: Auto
☐ XNA Rotate Animation Hack
☐ XNA Strict Options
Batch Mode: Off
☐ Batch Own Dir

Figure 7: FBX Export Settings in Blender.

Level design for each level in a modern game typically starts with concept art, sketches, renderings, and physical models. Once completed, these concepts transform into extensive documentation, environment modeling, and the placing of game specific entities (actors), usually with the aid of a level editor.

Figure 8 shows the concept model of the scene in Blender.

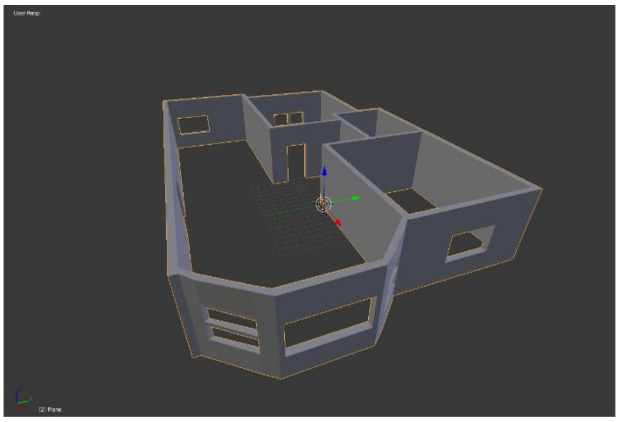


Figure 8: Concept design of the Scene in Blender

Level design is both - an artistic and technical process. Level design is necessary for two primary purposes - providing players with a goal and providing players with an enjoyable experience. Good level design in VR would strive to produce quality gameplay, and provide an immersive experience.[18]

It includes three stages: Concept, where the whole team primarily works with different ideas on paper before anything is implemented, the terrain modeling phase and the final scene. During the creation of the level, conceptual areas are planned for later object placement. (e.g. computer den, entertainment area, spots for objects in the room).

Figure 9 shows the initial build of the scene in Unity with some objects placed in and around the scene.

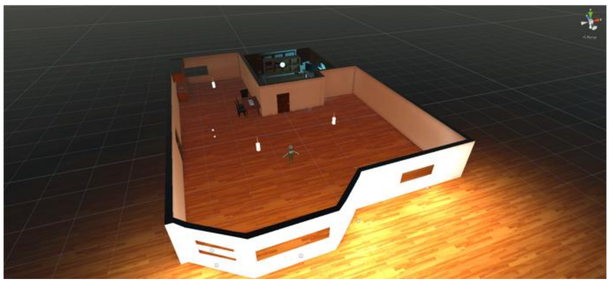


Figure 9: Initial build of the scene in Unity 3D

3.4 Lighting

After modeling and texturing, lighting is theessential building block of 3D because good lighting can improve your work, but bad lighting can ruin it. Since we need to build a game for smartphones with limited processing power, we have used baked global illumination lighting, which keeps the game optimized for running smoothly at over 60 frames per second. The whole scene is pre-lit during the process of light baking, and the final color distribution is stored in textures. Enclosed areas used different sources of light while some scenes in the game use natural daylight. The lighting and illumination are often used to steer the player towards the correct path unambiguously.

Lighting is the most important process in game development, only because it sets the theme and the mood for gaming. And especially in Virtual Reality where full immersion in the game is the primary objective, Lighting is the backbone for creating realistic visuals. Global Illumination[19] is achieved by ‘light baking’ the whole scene.

Baked Lighting is not calculated in real-time, it is pre-baked on the textures of different models. It takes a lot of time to bake the light on every texture present in the scene and each and every model. But once it’s done, during the gameplay there is nothing tobe processed on the lighting part.

Figure 10 shows the lighting being implemented and baked in a scene and the scene after the lighting has been applied successfully in that scene.

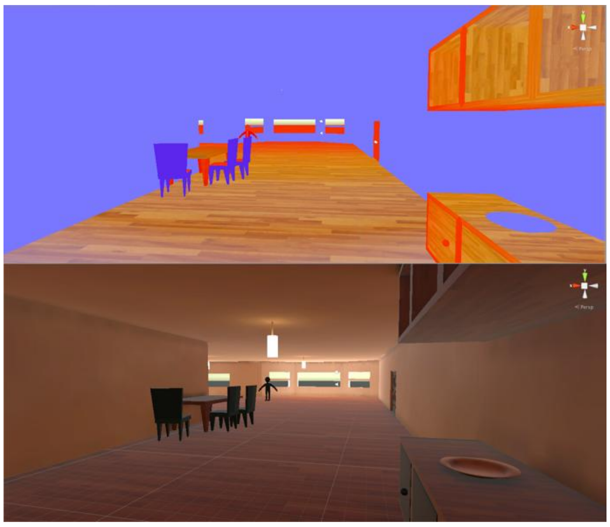


Figure 10: Lighting being implemented and it’s finished effect on the scene

3.5 Implementation

Scripting in game was done using C# on MonoDevelop. [20] Scripting is used to control all aspects of the game from player movement and animation to camera rotation.

Player movement in thegame was to be controlled by the head tracking on the phone and with the use of a Bluetooth Controller to move in all directions. The players' interactions with objects in his environment were also controlled through the utilization of the controller. To make use of the Bluetooth controller in thegame the 'Mobile VR Movement Pack' [21] by 'In Your Face Games' was downloaded from the Unity Asset Store[22]as it contained prefabs that could be configured in Unity to suit our needs. The interactions with objects were triggered using trigger colliders. [23]Unity's Audio features include full 3D spatial sound, real-time mixing, and mastering, hierarchies of mixers, snapshots, predefined effects and much more. [24]

Figure 11 shows the implementation of the character model in the game.

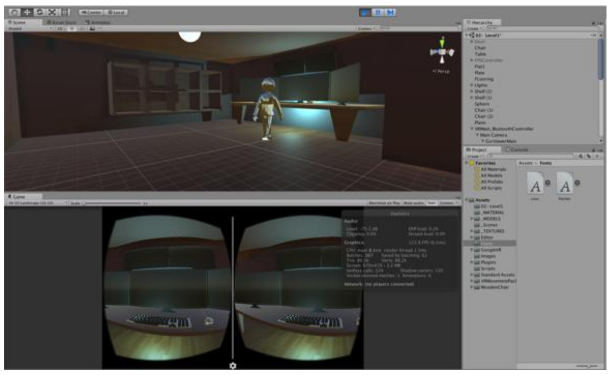


Figure 11: Implementation of the character model in-game.

3.6 Release

The game is still in the production phase, and the release date has not been decided. The final release of the game is slated to contain at least one complete single player level witha complete user interface and menu systems. We are also prototyping a local multiplayer system that will allow you to solve the puzzles along with your friends simultaneously.

4. SUMMARY

Together, as a team, we have deployed a professionalgame development studio workflow in the context of a virtual reality game development project. The utilization of 3D content in a virtual reality game put our theoretical knowledge into practical use. We improved not only our technical skillsbut also learned valuable lessons in teamwork. The biggest challenges we faced during the project were the different theoretical and practical understanding of the subject that each member had, theunalike working methods andvariable personal availability of the members.

5. REFERENCES

[1] Doug A. Bowman and Ryan P. McMahan, 2007, Virtual Reality – How Much Immersion is enough?

[2] Oculus Rift: https://www3.oculus.com/en-us/rift/

[3] HTC Vive: https://www.vive.com/us/

[4] PlayStation VR: https://www.playstation.com/en-in/explore/playstation-vr/

[5] Google VR: https://vr.google.com/

[6] Rido Ramadan andYaniWidyani, 2013, Game Development Lifecycle Guidelines

[7] Google VR SDK for Unity: https://github.com/googlevr/gvr-unity-sdk

[8] Blender: https://www.blender.org/

[9] Unity 3D Game Engine: https://unity3d.com/

[10] Unreal Engine 4: https://www.unrealengine.com/what-is-unreal-engine-4

[11] No Man's Sky (Hello Games): http://www.no-mans-sky.com/

[12] UV Mapping – Wikipedia: https://en.wikipedia.org/wiki/UV_mapping

[13] Basics of Realistic Texturing in Blender: https://www.blenderguru.com/tutorials/basics-realistic-texturing/

[14] Basics of Rigging in Blender: https://wiki.blender.org/index.php/Doc:2.4/Tutorials/Animation/BSoD/Character_Animation/Rigging

[15] Unity's Manual Pages for Animation Systems Overview: https://docs.unity3d.com/Manual/AnimationOverview.html

[16] FBX File Format - Wikipedia: https://en.wikipedia.org/wiki/FBX

[17] How to export from Blender to Unity 3D with textures tutorial: http://www.mat-d.com/site/how-to-export-from-blender-to-unity-3d-with-textures-tutorial/

[18] Level Design - Wikipedia: https://en.wikipedia.org/wiki/Level_design

[19] Unity's Manual Pages for Global Illumination: https://docs.unity3d.com/Manual/GIIntro.html

[20] MonoDevelop: http://www.monodevelop.com/

[21] Mobile VR Movement Pack by In Your Face Games: https://www.assetstore.unity3d.com/en/#!/content/69041

[22] Unity's Asset Store: https://www.assetstore.unity3d.com/en/

[23] Unity's Manual Pages on Colliders: https://docs.unity3d.com/Manual/CollidersOverview.html

[24] Unity's Manual Pages on Audio Integration: https://docs.unity3d.com/Manual/Audio.html