OULUN AMMATTIKORKEAKOULU

Antti Konsén

3D model of a real world model

Creating import-ready asset.

3D model of a real world model

Creating a import-ready asset

Antti Konsén
Thesis
Fall 2015
Tietojenkäsittely
Oulu university of applied sciences

ABSTRACT

Oulun university of applied sciences
Tietojenkäsittely

Author(s): Antti Konsén
Title of thesis: 3D model of a real world model
Supervisor(s): Matti Viitala
Term and year when the thesis was submitted: Fall 2015 Pages: 32

The goal of this thesis was to create a 3D model of a real world model and make it into a import-ready asset to be used for games and other modelling software. The topic comes from my own experiences working in game projects and from the skills and training I acquired in Oulu Game Lab-program.

The created thesis works target was to work as an import-ready asset for different game engines and modelling software to be further used and polished. It's a platform for further development of the model or a way for someone else to learn and study how a real world model can be transferred to a game environment. The work will be defined in questions and areas of 3D model development which is not yet fully covered in the school's educational program of 3D modelling. Such as trying to replicate a real world model. At the end there should be a working 3D model asset to be used in future projects for game development or further polish the model in a 3D modelling program.

Source material used for this thesis combines of several guides and documentations created by the software developers and 3D modellers of Blender and SimBin communities. These communities and developers user manuals provide an ample source for information and guidance for the thesis.

The subject is worth studying as creating models for already made games is often the first touch many people have to creating content to games before making their own games from scratch. The most important questions the work concentrates on is acquiring correct reference material for the model and using it as a guideline for creating the model and making it to a readily usable asset for game engines or further polish for other users.

The tools used in this thesis project include but are not limited to Blender 3D modelling software, SimBin importing software and limited use of Adobe Photoshop for creating reference material.

Keywords: 3D modelling, games, Blender, asset, model

CONTENTS

TERMS AND ABBREVIATIONS

Term or abbreviation	Explanation
Active	One of the three selection states in Blender. Active objects appear in yellow.
Aliasing	Rendering artifacts in the form of jagged lines.
Alpha Channel	Additional channel in 2D image for transparency
Axis	A reference line which defines coordinates along one cardinal direction in n-D space.
Bevel	Operation to chamfer or bevel edges of an object
Bezier	A computer graphics technique for generating and representing curves
Curve	A type of object defined in terms of a line interpolated between control vertices.
Edge	Straight segment that connects two vertices, and can be part of a face.
Face	Mesh element that defines a piece of surface. It consists of 3 or more edges.
Layer	A device for organizing objects.
Mesh	Type of object consisting of vertices, edges and faces
Normal	The normalized vector perpendicular to a surface.
NURBS	A computer graphics technique for generating and representing curves and surfaces. (Non-uniform rational basis spline)
Pixel	The smallest unit of information in a 2D raster image, representing a single colour made up of red, green and blue channels.
Polygon	Any two-dimensional shape with multiple sides connected at vertices to enclose the shape. Used as a measure in 3D modelling to measure the level of detail.
Primitive	A basic object that can be used as a basis for modelling more complicated objects.
Projection	In computer graphics there are two common camera projections used.
Perspective	A perspective view is geometrically constructed by taking a scene in 3D and placing an observer at a point.
Orthographic	In an orthographic projection, you have a viewing

	direction but not a viewing point.
Render	The process of computationally generating a 2D image from 3D geometry.
Subdividing	Technique for adding more geometry to a mesh. It creates new vertices on subdivided edges, new edges between subdivisions and new faces based on new edges.
Texture	Specifies visual patterns on surfaces and simulates physical surface structure.
Topology	Arrangement of Vertices, Edges, and Faces which define the shape of a mesh. See vertex, edge, and face.
Vertex	A point in 3D space containing a location. It may also have a defined colour. Vertices are the terminating points of edges.
Vertex group	A collection of vertices.

1 INTRODUCTION

1.1 3D in video games

3D games can be traced back as far as 1970's when games such as Maze War. Maze War was originally written by a NASA employee Steve Colley in 1973 who had written the program for portraying and navigating mazes from first-person perspective in a 3D environment. Maze Wars was later developed further to include multiplayer and ability to shoot other players. The game mechanics were simple as you could only move front, back, left and right one tile at a time and other players appeared as eyeballs. Maze Wars could also be considered the first multiplayer/first person shooter game as it allowed 4 teams of 8 people for a total of 32 players to battle. (Stories from the Maze War 30 Year Retrospective, 2015.)

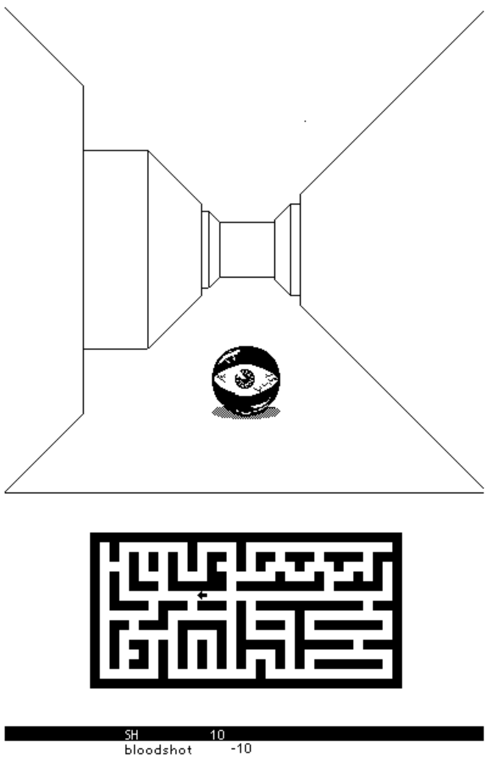
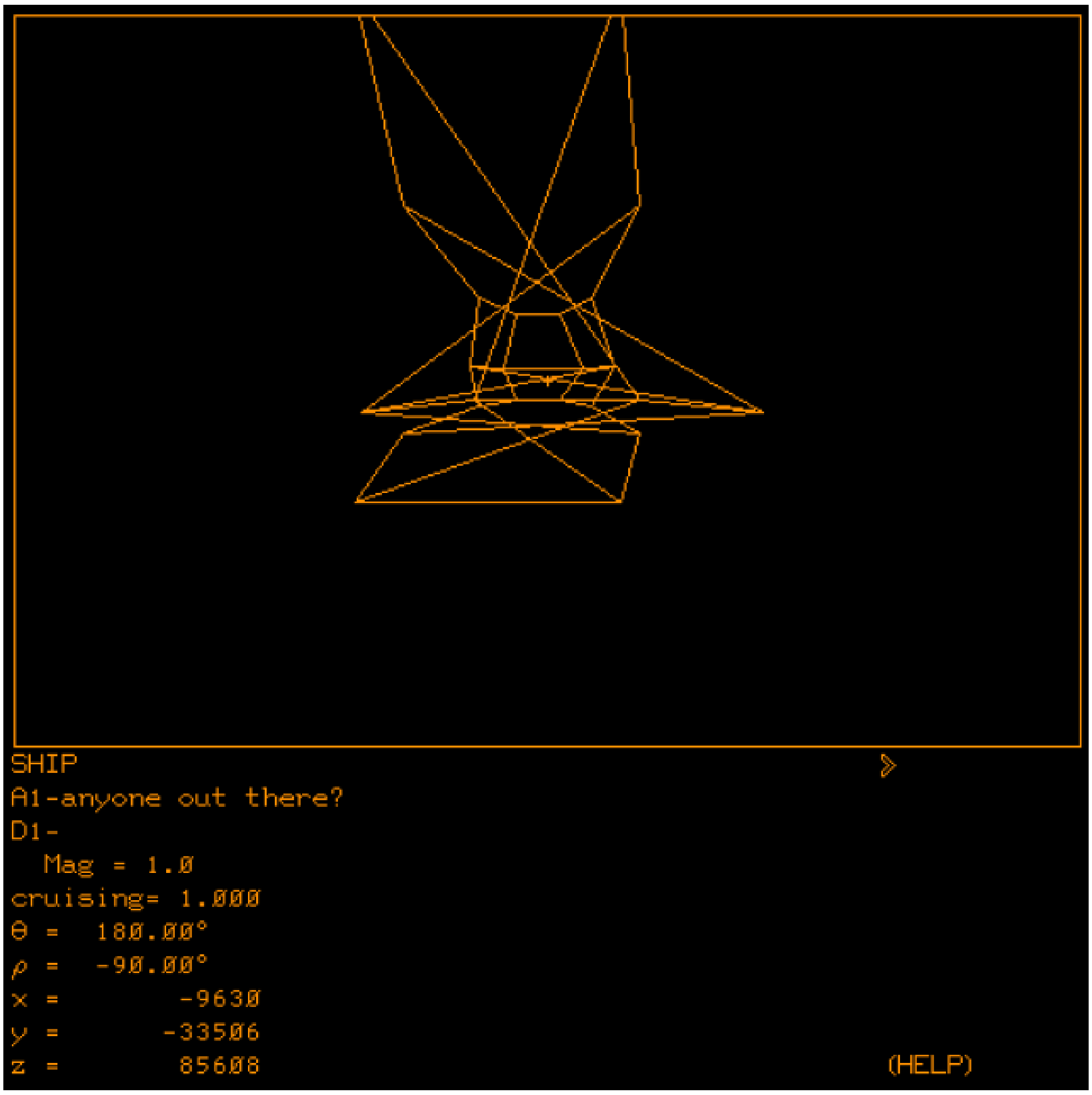


FIGURE 1: Player in Maze Wars

Around the same time Spasim (abbreviation of ”Space Simulation”) came out which also featured 3D graphics and multiplayer. The game involved 4 planetary system with up to 8 players per planetary system, flying around in space which the players appeared to each other in wire-frame space ships. In it's initial release in March of 1974 the game was a simple team-based Star Trek-type game with multiplayer and first-person-shooter dynamics. The second version released later in July the same year included more strategy elements, including space stations and resource management. The objective in the second version was to try to avoid going to war with other players and possibly cooperate to get to a far off planet where players could obtain enormous amounts of resources. If the player went to war or just flew around admiring the constellations they could experience revolts on their home system and watch helplessly as their planets population and resources diminished. (Educational Feature: A History and Analysis of Level Design in 3D Computer Games, 2015.)



```
SHIP                                             ≯
A1-anyone out there?
D1-
  Mag = 1.0
cruising= 1.000
θ =   180.00°
ρ =   -90.00°
x =        -9630
y =       -33506
z =        85608                                (HELP)
```

FIGURE 2: Vector graphics in Spasim

3D graphics were however difficult to produce on early computers and they didn't come to mainstream use. The first big 3D success was Battlezone, a tank warfare game released in 1980 that used vector graphics, just like Asteroids. While the game seems simple by modern standards it was incredibly complex for such an early adaptation of 3D graphics, offering the players the ability to go anywhere in the game world, hide from attacks and fight enemies. It was also one of the first times a 3D game was applied to real life training, a version called ”The Bradley Trainer” was designed for use by the U.S. Army. The army version used viewing goggles that the player puts his face into, making Battlezone as the first virtual reality game and the first virtual reality training device used by the U.S. Army. (Bradley Trainer, 2015; Electric Escape, The Army Battlezone Q & A, 2015.)

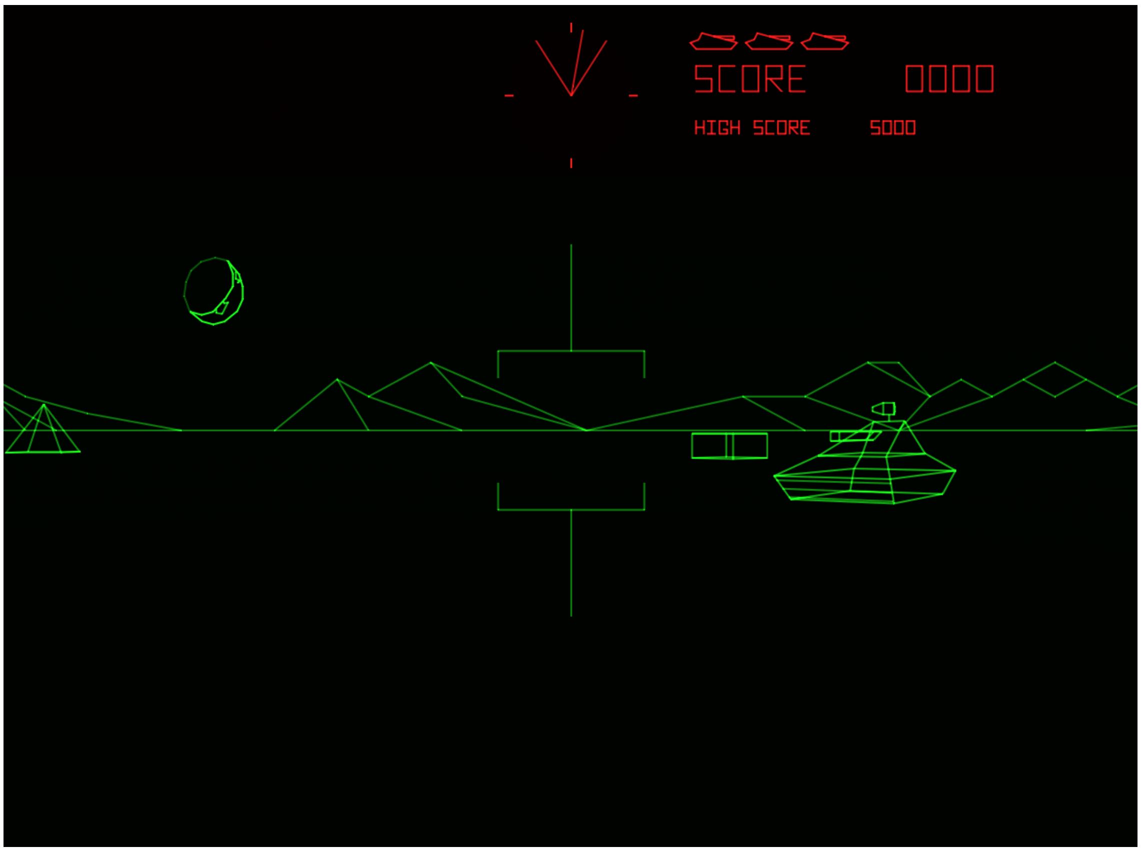


figure 3: Battlezone view through the periscope

Majority of the early 3D games stuck with simpler technologies, simply getting a game to look like 3D was impressive, throughout the 1980's games used simple vector graphics which worked great as vector graphics is the use of geometrical primitives such as points, lines, curves and shapes, all of which are based on mathematical expressions which require by modern standards very low system requirements to run. In most cases throughout 1980's the trick to having the effect 3D effect was to fake the third dimension, not to do it at all. Games such as the early Wing Commanders or Pole Position gave the illusion of moving through 3D space when in reality the feel of depth was created by using scaling sprites up and down. This technique is referred to as pseudo-3D. (The International Arcade Museum, Pole Position, 2015.)

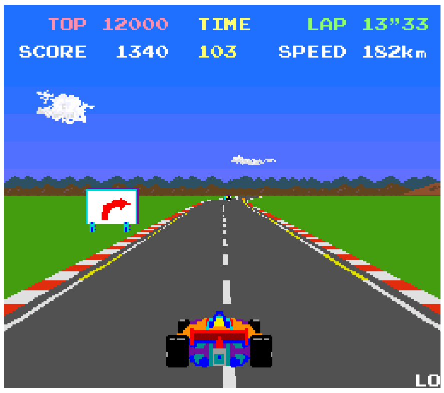


FIGURE 4: Gameplay view of Pole Position

The first full 3D world came with Quake in 1996. It featured fully modelled and animated enemies and had the first 3D boss fight in a first-person-shooter. But because of these modelled and animated enemies and limited system resources the whole world looked a bit drab and simplistic. Textures were with minimal colours and repeated their pattern often to make up more processing power and storage for the 3D models. Quake was very much the pinnacle of what could be done with only a CPU to portray game environments, as dedicated 3D cards came into their own around 1997.(American McGee on Quake, 2015)

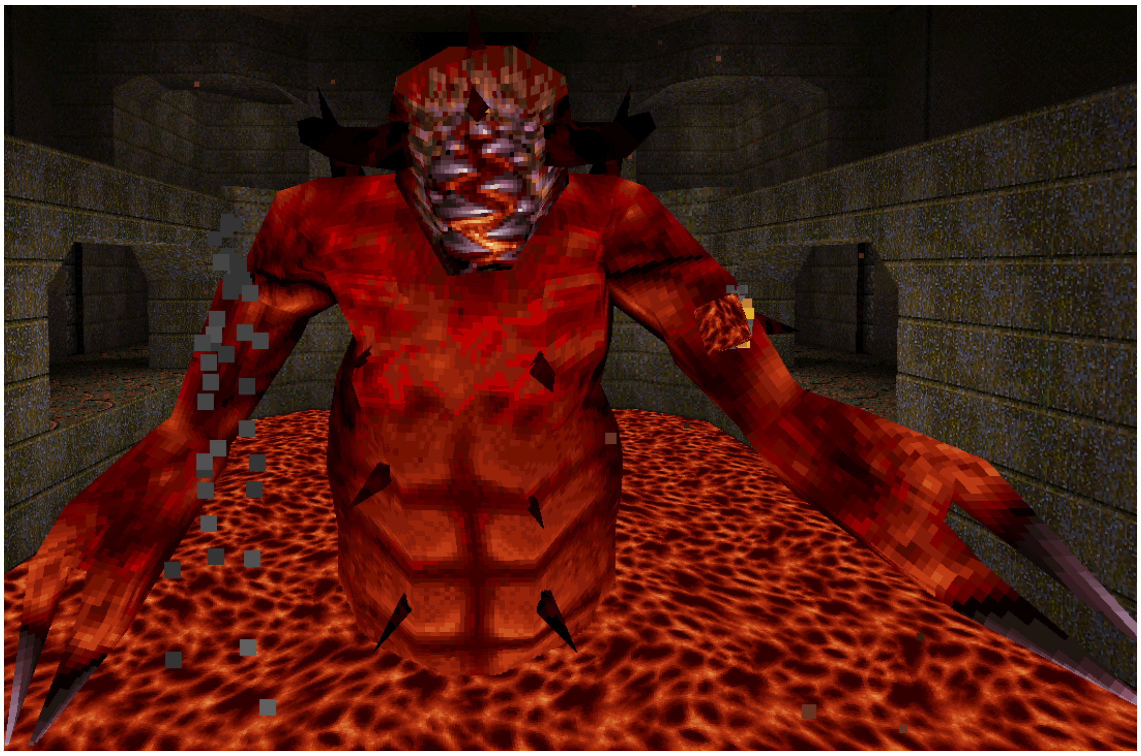


FIGURE 5: Chthon from Quake

When dedicated 3D graphics cards started arriving to consumer market the race to become to most dominant card was between several mutually incompatible brands. By far the most successful was 3DFX and it's Voodoo cards. Even today with integrated 3D graphics that come standard on most motherboards, a separate 3D card is required for most games. However in the late 90's the core functions of separate graphics cards were transformation, clipping and lighting – basically using the card to work out where objects in the world were and how they should be lit.

The next big steep was the addition of shaders. Shaders are additional calculations thrown into the rendering pipeline that work on individual pixels, vertices and pieces of geometry that add effects and change the final image. Effects such as working shadows for characters and objects, making a flag flutter or adding bump mapping that gives the illusion of a raised and lowered areas on an object without the need to add polygons. (The History of the Modern Graphics Processor, 2015)

As developers didn't have to work their games around existing hardware as much anymore, they could focus on making the most of what they could. Valve was the first company to largely use skeletal-based characters rather than keyframe-by-keyframe animated enemies in their first released game Half-Life. The use of ragdolls in characters spread to every platform and game until the ones that didn't offer them felt clunky and old in comparison. Importantly the new realism of these games finally opened new possibilities to developers for genres they weren't able to do before as corridor shooters. The first major step into full living 3D modelled worlds came with Rockstar North's (then known as DMA Design) Grand Theft Auto III in 2001. (History of Grand Theft Auto, 2015)

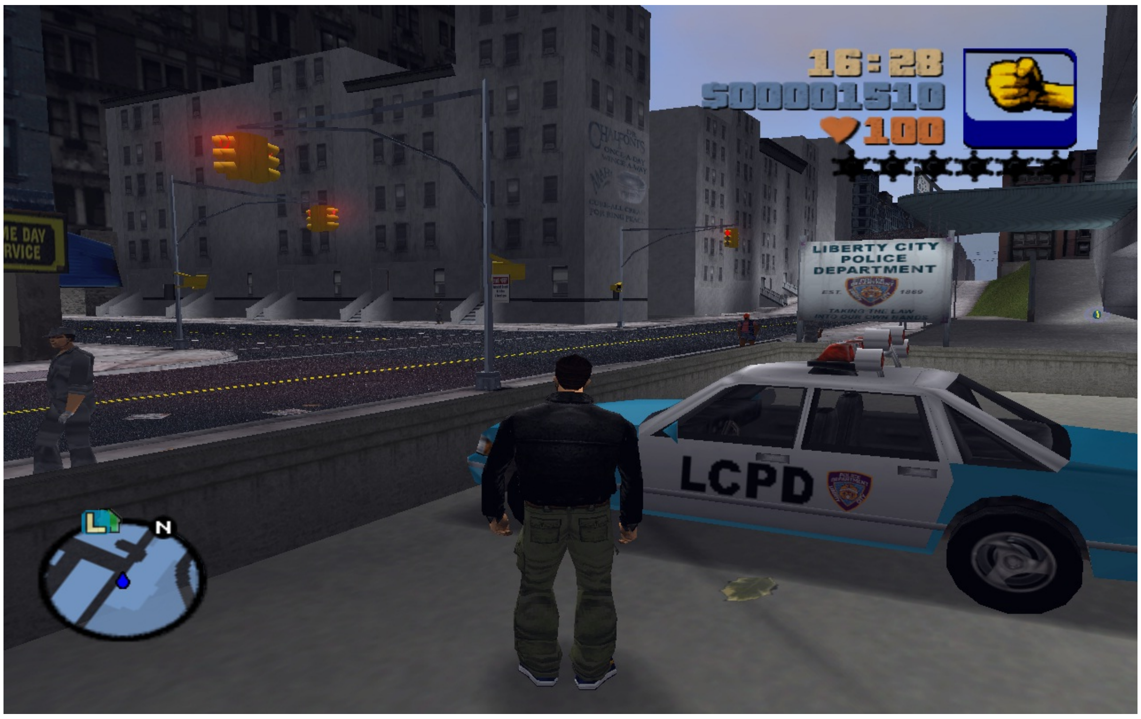


FIGURE 6: Grand Theft Auto had ground breaking graphics for its time

Now that the issue is not processing power anymore, it's creating content for these games. A simple corridor shooter or a maze for 3D shooter could be done from game concept to release in under a year by a competent team, but building coherent living worlds, universes and other real-world elements require an immense amount of work hours and assets.

1.2 3D modelling basics

As a general rule, a 3D model is built from small virtual elements, surfaces (face) that form a mesh network. They represent a physical body using a collection of points in 3D space. These models are connected by various geometric entities such as triangles, curved surfaces, lines, etc. The surface itself is formed by at least three vertices (vertex) and the top (edge). A polygon mesh is formed by surfaces that share one or more neighbouring vertex. Meshes can be edited by moving vertices. Generally meshes density affects how detailed the 3D models shape become as the smoothness and accuracy of the model are defined by these factors. 3D modelling tools such as Blender, 3dsmax or Maya provide the most versatile mesh editing environments, such as the simultaneous transfer and modification of multiple vertices to define their distances to scale.

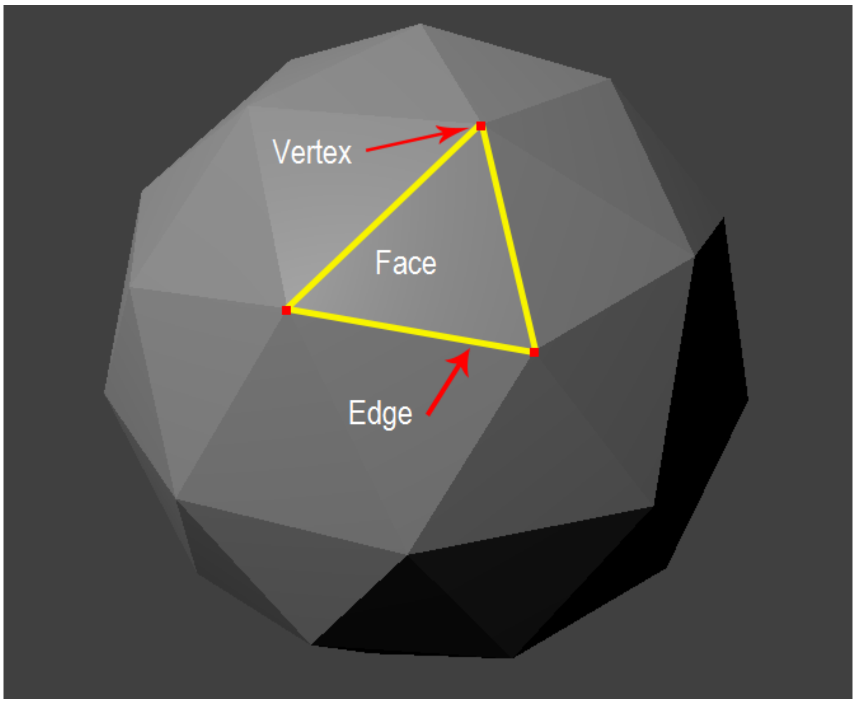


FIGURE 7: Elements of mesh

Almost all 3D models can be divided into two categories. Solid and Shell. Solid models define the volume of the object they represent, these are more realistic but also harder to model. Solid models are mostly used for non visual simulations such as engineering simulations, for CAD and specialized visual applications. Shell models represent the surface of the object and not it's volume. These are easier to work with than solid models and almost all models used in games and film are shell models.

1.3 3D modelling process

Modelling is the process of taking a shape of an object and molding it into a completed 3D mesh in a 3D environment. The typical means of creating a 3D model is to take a simple object, called a primitive, and extend or "grow" it into a shape that can be refined and detailed. Primitives can be anything from a single point (called a vertex), a two-dimensional line (an edge), a curve (a spline), to three dimensional objects (faces or polygons). Using the specific features of your chosen 3D software, each one of these primitives can be manipulated to produce an object. When you create a model in 3D, you'll usually learn one method to create your model, and go back to it time and again when you need to create new models.

There are three basic methods you can use to create a 3D model, and 3D artists should understand how to create a model using each technique.

Spline or patch modelling, in which the spline is defined by at least two control points in the 3D space, the most common splines used in 3D are bezier curves and NURBS (modelling software such as Maya are heavily based on using this method of modelling). This method is perhaps the oldest and most traditional form of 3 modelling available. The 3D modelling software can create a patch of polygons to extend between two splines, forming a 3D skin around the skeleton shape. Splines are not used very often these days for character creation, as creating accurate models using this method can be very time consuming. Spline modelling is used primarily for the creation of hard objects, like cars, buildings and furnitures. This is one of the methods that are be utilized in this thesis project.

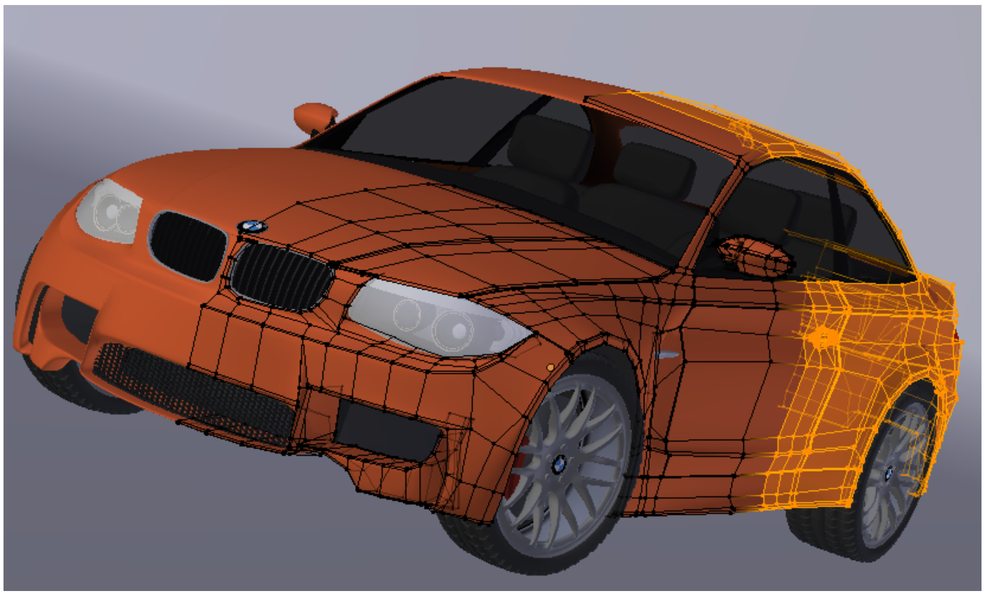


FIGURE 8: 3D-graphics

Box modelling is probably the most popular technique and bears a lot of similarities to traditional sculpting or molding. In this technique one starts with a primitive, usually a cube and begins adding detail by slicing and adding to the cube by extending faces of the cube to gradually create the form you're after. Box modelling is often used to achieve the basic shape of the model. Once practised this method is a very quick way to achieve acceptable results.

Poly modelling & edge extrusion which is probably not the easiest to get started with but it is perhaps the most effective and precise technique. In poly modelling, one creates a 3D mesh point-by-point, face-by-face. One will often start with a single quad (3D object consisting of 4 points) and extrude an edge of the quad, creating a second quad attached to the first. The model is then created gradually in this way. While this technique is not as fast as box modelling, it demands less tweaking of the mesh to get it to the desired level of detail.

2 SOFTWARE USED IN THESIS

2.1 Blender

Blender is the main 3D modelling software that is used in this thesis work, it is a free and open source 3D creation suite which supports the entirety of the 3D pipeline – modelling, rigging, animation, simulation, rendering, compositing and motion tracking. While in this thesis the focus is on the creation of the model from reference material and bringing that model to a import-ready state to be used in games.

Blender is cross-platform and runs on Linux, Windows and Macintosh computers. It was first created in 1995 in Holland by a advertising agency called NEO-GEO for their internal use and it ran on a UNIX-based operating system called IRIX. When the agency closed their doors a new company was formed called Not a Number (NaN) who kept developing the software further and published versions for Windows and OS X. Throughout it's development it was distributed as freeware without source code. NaN declared bankruptcy in 2002 after which Blenders main developer Ton Roosendaal started a campaign to gather 100 000 euros to buy the rights for the source code, the campaign was successful and the software was re-licensed under the GNU General Public License which gives the right for anyone to use, copy, modify and redistribute the software and it's source code. (History of Blender, 2015.)

Blender has a reputation of being a hard to use software as it relies heavily on the users knowledge of using shortcuts rather than browse through menus. Blender is mainly used in different modes which are object-oriented which means that the whole application is always in one and only mode, and that the available modes vary depending on the selected active object's type, most of them only enable the default object mode. Each mode is designed to edit an aspect of the selected object.

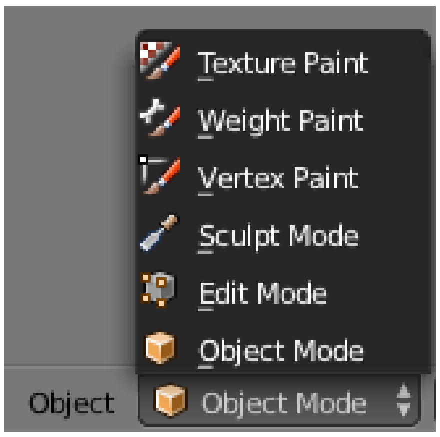


FIGURE 9: Blender modes

In this thesis project there are three main modes that were used.

Name	Description
Object mode	The default mode that is available for all object types, it is dedicated to object editing.
Edit mode	A mode available for all renderable object types, it is dedicated to their shape editing.
Sculpt mode	A mesh-only mode that enables Blender's 3D-sculpting tool.

Blender also has it's two own integrated 3D-renderers, Blenders internal render engine and Cycles-renderer.(Blender Manual, 2015.)

2.2 Adobe Photoshop

Adobe Photoshop is the predominant photo editing and manipulation software on the market. It is an image editing software that can edit and compose raster images in multiple layers. In addition to raster graphics Photoshop has it's limited abilities to edit or render text, vector graphics, 3D graphics and video. These features can be extended by adding plug-ins, programs developed and distributed independently of Photoshop that can run inside it and offer new or enhanced features.

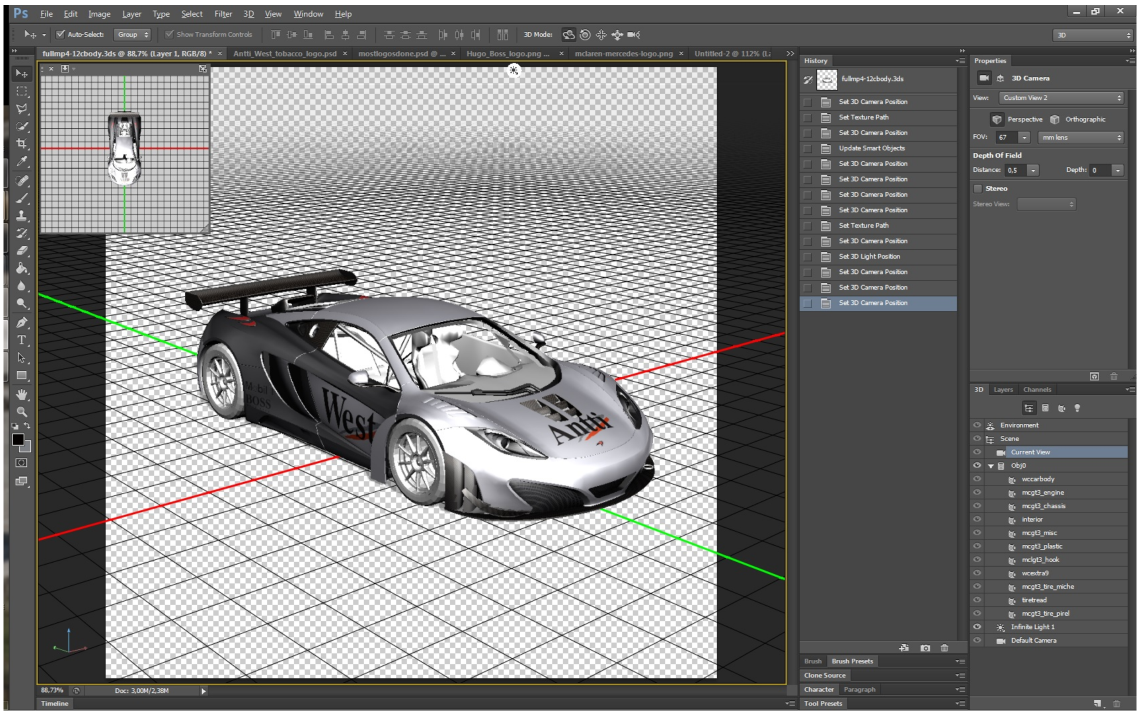


FIGURE 10: Textured 3D model in Photoshop

Photoshop has a vast support for graphic file formats but uses its own .PSD and .PSB formats that Blender also supports.(Adobe Photoshop Softpedia, 2015)

2.3 Unity

Unity is a cross-platform game engine that is used to develop video games. It's free for non commercial use and is widely used by game developers and it's being used as the primary tool in OAMK game development courses and programs. On top of the development platform Unity offers Asset Store where developers can share and sell their import-ready assets of models, textures, scripts, animations and sounds which makes it a good platform to export the thesis work for further use. (Unity, 2015.)

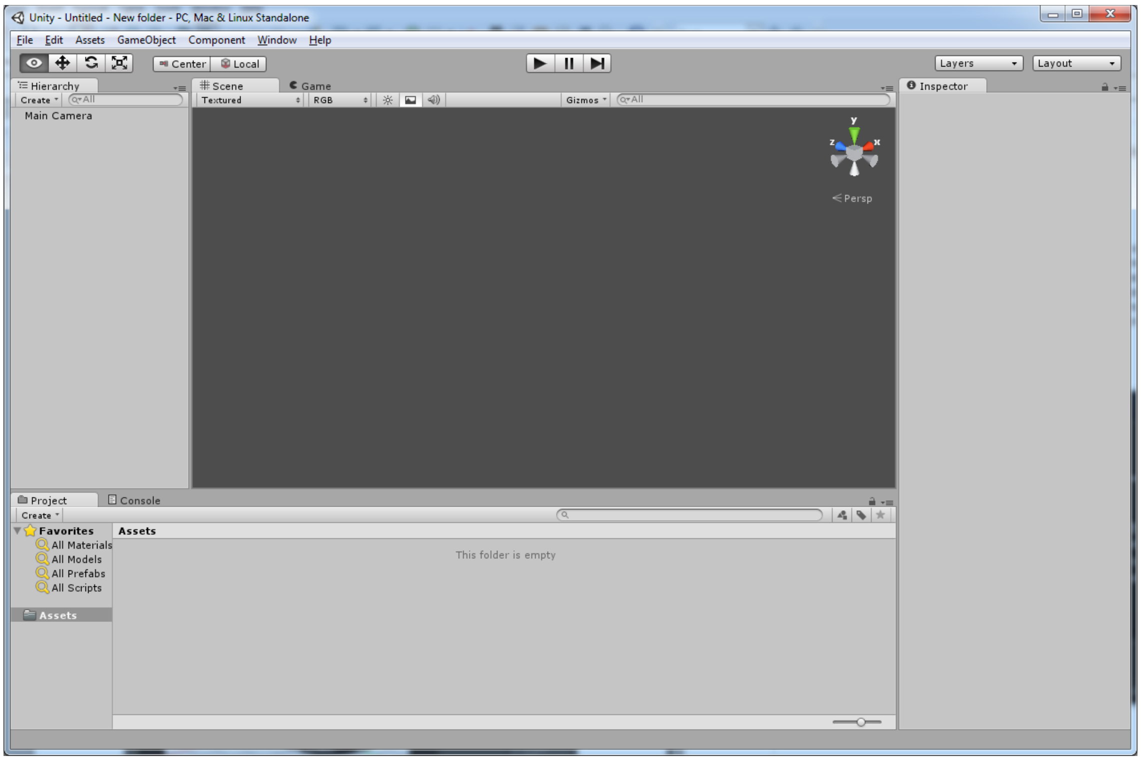


FIGURE 11: Unity game development tool

It uses the same kind of 3D development view as Blender with the exception of having different Y and Z directions but these are reversible in the Blender import options.

Unity also has native support for .blend file format.

3 PRACTICAL WORK

3.1 Acquisition of reference material

The object of the thesis project is a Mitsubishi Galant produced through 1996-1998. This reference material provides a benchmark for measurements and scale in the 3D environment.

To be able to trace the shapes and forms of the model high quality images are needed from several angles, especially from every 90° degree angle that can be later transferred to Blender to work as a blueprint for the exterior model. Most reference material can be found through search engines and hobbyist websites but for small details I used my own car as a reference when it was applicable. There are also a few games that also possess a model of this car such as Gran Turismo-game series for PlayStation which was also at disposal.

There are also free available blueprints online with real world measurements that greatly help with scaling of the model and getting measurements right.

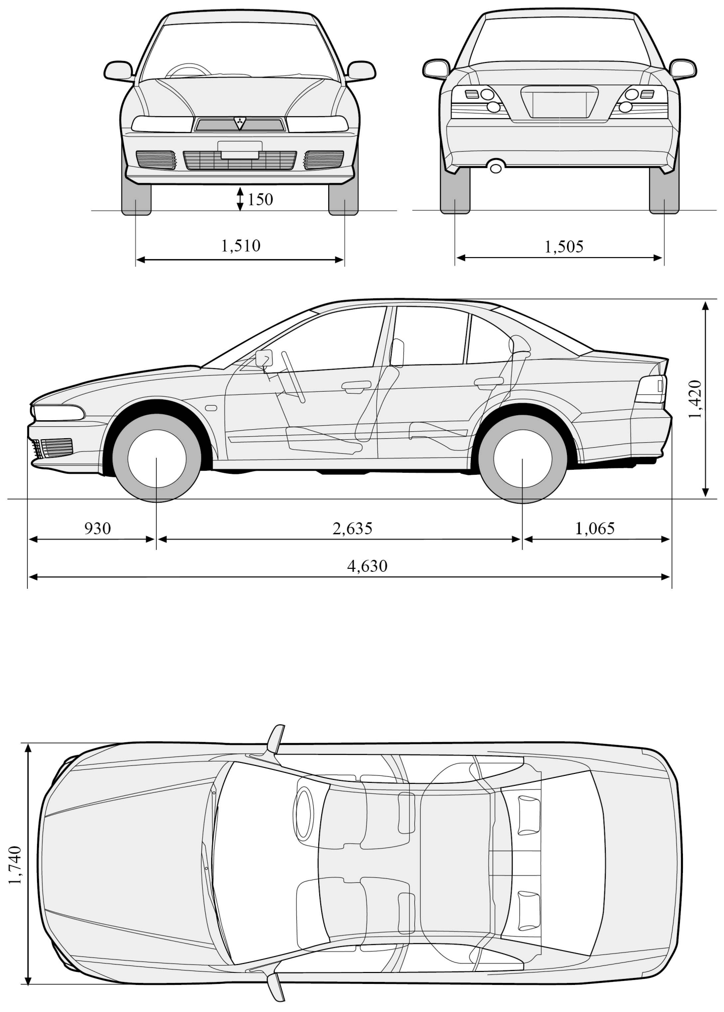


FIGURE 12: Blueprints of the car

3.2 Editing the reference material to a usable form

To be able to replicate the real world model accurately in the 3D environment the reference pictures need to be exact so that the full side, front, and rear of the car are visible so no detail are left out as a result of cropping or bad reference. Using the blueprint as the reference material helps getting the overall shape of the car but it doesn't hold much detail in itself of the depth and curves of the model.

Using the blueprint all the different angles of the car were separated into different image files to fit them later separately into Blenders viewports. Editing out all the unnecessary lines and artifacts and cropping the image for simplicity helps the modelling process later on. The measurements on the blueprint clean are a welcome bonus as they help with scale and measurements of the car.

For small details such as badge, side skirts and others that require depth pictures and measurements, pictures of my own car and photos available on google image search were used.

For reference pictures that aren't line drawing and have a lot of artifacts in them, All the parts that aren't part of the model were cropped out and the referred material was highlighted and made as visible as possible. If the photo is from a different angle than the blueprint it was important to find a reference point so that once the image was imported to Blender, the model can be rotated so that it matches the angle of the reference photo.

One of the best tools in Photoshop for cleaning up reference images is quick selection that allows you to select pixels based on tone and colour, it's somewhat similar to the Magic Wand tool but it allows looking for similar textures in the image which makes it great at detecting the edges of objects.

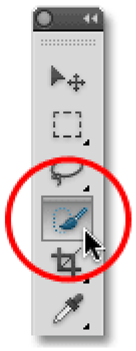


FIGURE 13: Quick Selection tool

Another tool is Transform that was used to scale, rotate, skew and warp selections. Distort is used to remove unwanted artifacts caused by camera lenses, like barrel and pincushion distortion, chromatic aberration and vignetting. Warp to drag control points to manipulate the shape of images, shapes or paths.

There are also vertical and horizontal guides that help lining up the image, any obvious rectangular element or straight element can be utilised to determine the correct vertical or horizontal alignment.

3.3 Creating the model using the reference material

The work is started by using the box modelling or digital sculpting technique. Starting from a single box primitive to get the shape and form in the Z- and X-axis for the car. Starting from the main body panels of the car to achieve the desired shape. Switching between actives to define the basic shape as the edit mode doesn't always show if there are multiple vertices on a single point. (Gahan 2010, 32.)

Using the pictures cleaned up previously by adding them to the Blenders viewport by enabling background images and then selecting the appropriate viewing point for each picture.

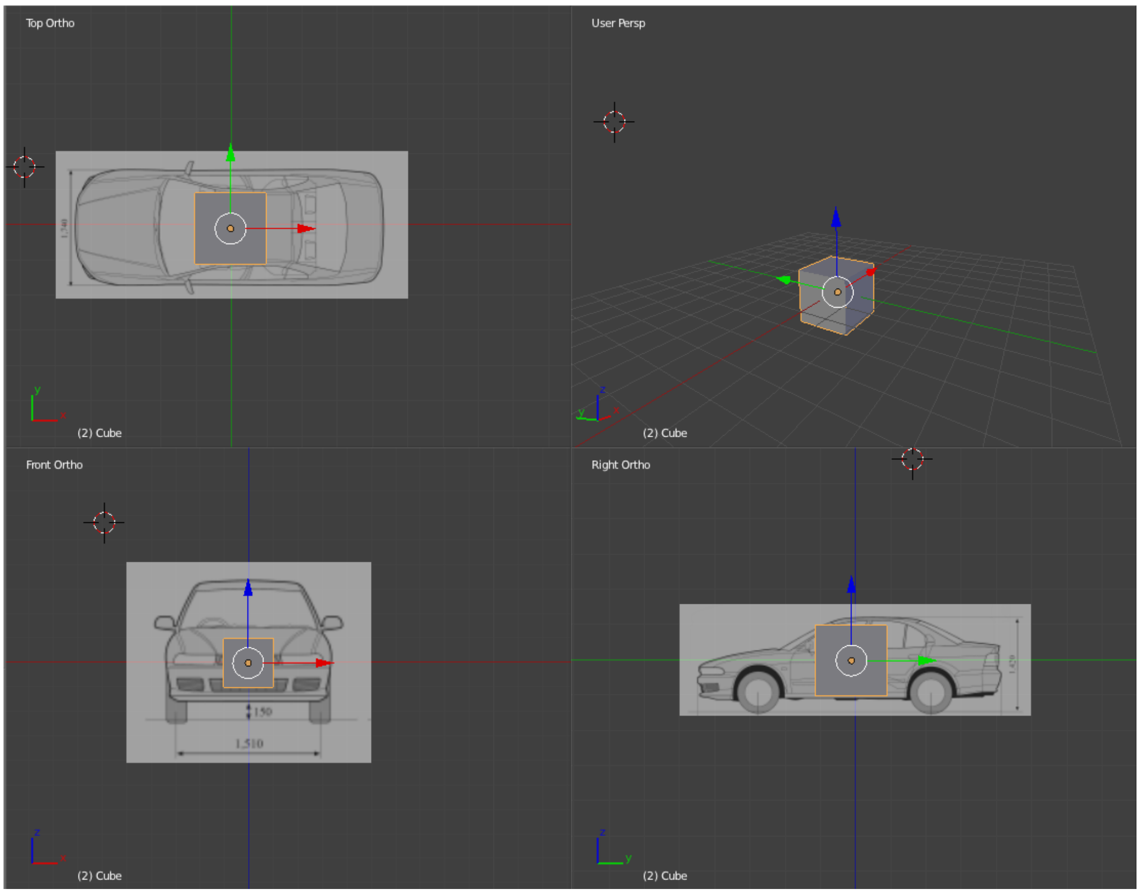


FIGURE 14: Four viewports with background images set in Blender

I used 3 orthographic viewpoints to show me top, front and side view of the model with a reference image in the background and one user perspective that allows me to rotate around the model to see how it's forming up and prevent any early mistakes. Using the alpha channel can define the opacity of the background image to help me differentiate between the model and the background image.

Using the orthographic viewports tracing along the blueprint to get the basic shape of the car, extruding, moving and adding vertex points based on the level of detail the blueprint shows. Having a blueprint in vector graphics helps here as they have mathematical data points and are indefinitely scaleable so you're not hindered by the quality of the blueprint, rather your patience and attention to detail.

Detail can be added later on by smoothing lines but using these tools requires caution as they can add unwanted hidden geometry that greatly increases the polycount and decreases the performance when rendering the model in a game environment.

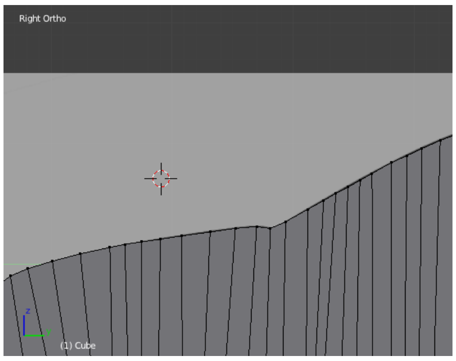


FIGURE 15: Tracing model along the background image

When the model starts getting it's longitudinal silhouette in Y-space I go through the vertex points to remove any unnecessary points that could cause problems later, having a clean and simple model early on before doing width and depth is crucial to have a good base to work on.

Now that the model has it's longitudinal shape I change my view to the frontal orthographic viewpoint where I can start working on the cars shape when looking it from the front, using the same technique I add vertex points from the basic shape along the lines of the blueprint to get the general shape. I only follow the basic body shell lines as the windows, side mirrors and others can be added later as separate objects making the whole process easier modelling and converting wise.

Because the car is symmetrical in design in X-axis (width) I only need to trace half of the car as I can duplicate and mirror it to the other side with Blender.

It's advisable to remove any faces in places where there is no solid surface needed or the car doesn't have any, such as the windows, the air vents and scoops in the front bumper. The reason is that when the model is added to a game later or you want to have them empty as you can add these parts as a separate objects or the game requires you to use stock content for light fixtures or wind shields as they often have damage modelling built in. So for now the lights, windows, air scoops and wheel arches are left empty and the main focus is the body shell of the car.

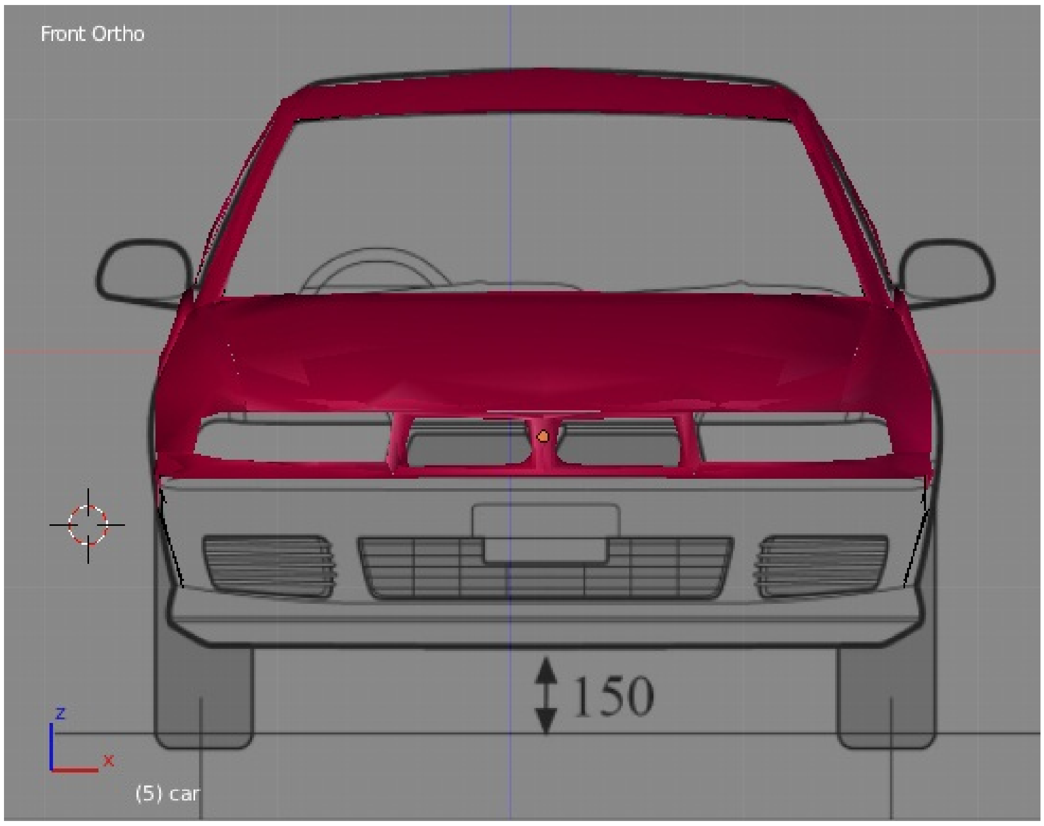


FIGURE 16: Front top half of the cars front aligned to background image

The basic shape of the car starts to form as the model starts to resemble the real car, the work was started from the car's front first to give a good scale of all the details that go into it and it's arguably the most recognisable things in the model as well as my favourite thing about it. The blueprint helped with the general shape of the car but for the front bumper I used a more modern ”facelift” version that I'm more accustomed to, you can notice the two vertical spokes going up in the lower part of the bumper in the middle with air scoops on the side as the original blueprint and older version had a uniform model for this.

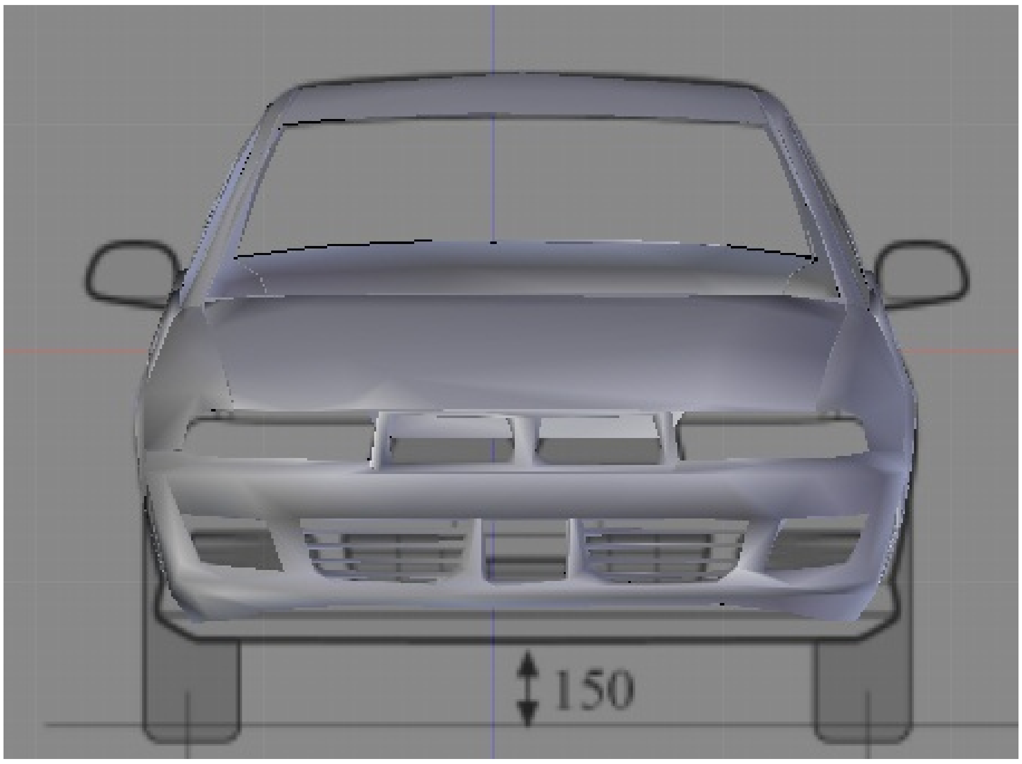


FIGURE 17: Cars front bumper done

As the car now had it's basic shape in X,Y and Z axis and only it's outer shell is present, it has 1174 faces so it's not very detailed and requires minimal computational power to render, model of this accuracy would be ideal for a portable game device as their processing power is limited.

Next step is adding detail to the model as modern racing games use well over 100 000 faces for each car depending on the graphical fidelity.

Majority of the detail will come from smooth rounded surfaces and added detail to smaller parts of the car. Bezier curves were used in this instance to allow simple control points that have handles for rotation, the curves are mathematical so they have infinite fidelity so transforming them to a polygonal form gives you the option to define the accuracy and amount of polygons used for the shape.

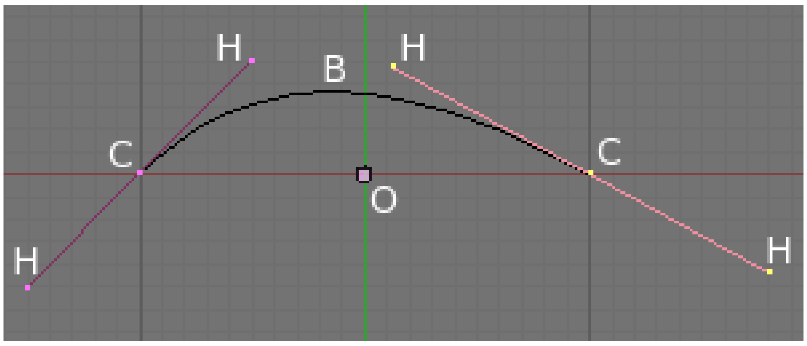


FIGURE 18: Bezier curve

For the model's depth in places where the blueprint doesn't get like front bumpers deep air scoops real world measurements were used from pictures and my own car. The headlight gaps that are visible and measurable on the blueprint for the front of the car, this portion can be measured and compared to the distance and depth of the air scoops which give a close estimate on how deep the bumpers details sink in. In many cases the same method was used of measuring something that's available in blueprint and then comparing it to something that isn't available in an accurate form. For most gaps and round surfaces bevel tool was used to create chamfered or rounded corners to the model. It's an effect that smooths out edges and corners. Bevel is a useful tool to add realism to non-organic models as blunt edges on objects catch the light and change the shading around the edges.

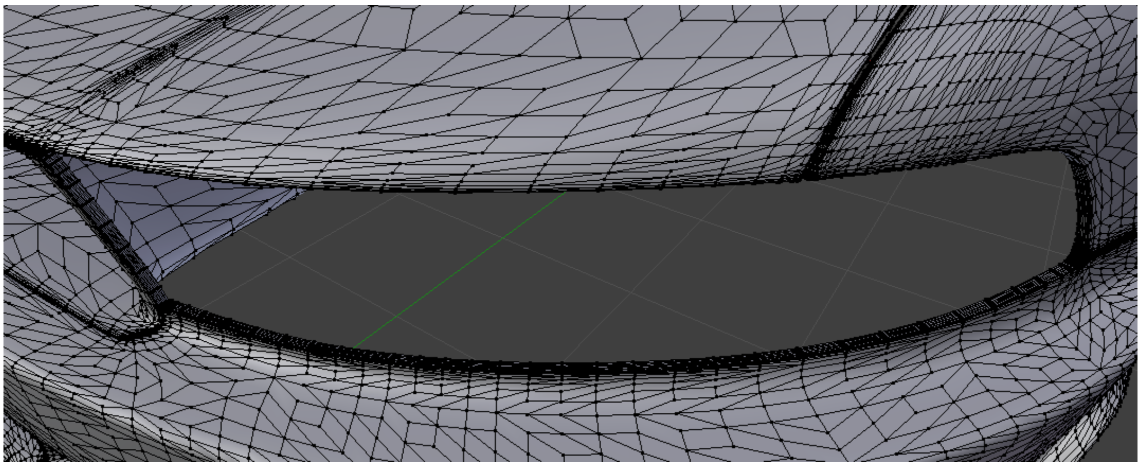


FIGURE 19: Visible polygons in the front lamp housing

After doing one side of the car with higher detail and focusing on the small parts linked duplicate was created which could then be mirrored on one axis to produce a perfect mirror-image copy that updates in real time as it was being edited, halving the work. If there are points from using the mirror modifier on some parts where vertices would overlap each other clipping was used so as soon as the vertices are within a predefined merge limit, they are clipped together and cannot be moved beyond the mirror plane. If several vertices are selected and are at different distances to the mirror plane, they will one by one be clipped to the mirror plane. It's sort of sowing the whole model together from the middle.

For detail on large surface areas like the roof, hood and windows, NURBS surfaces were used to get the correct amount of curvature and detail to match the reference. Just like NURBS curves, it has a resolution modifier that controls the detail of the surface.

As the car gains detail it also racks up a lot of faces and require more processing power but having a high detail model in the end is better as it's harder for end-users to add detail to the model rather than remove.

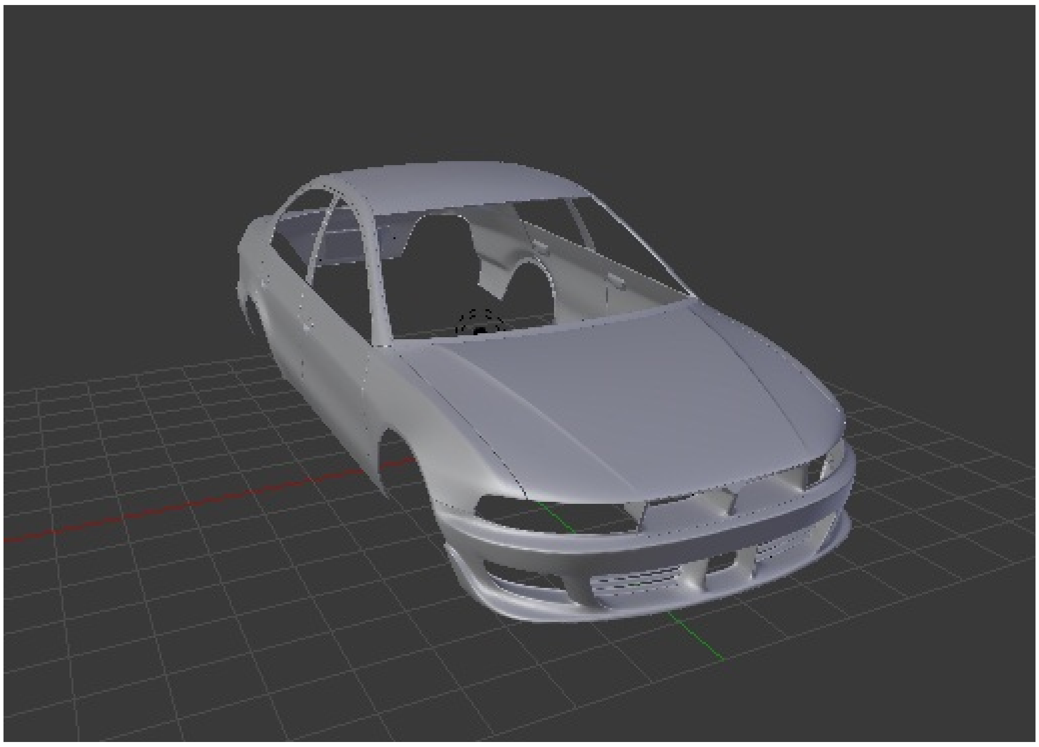


FIGURE 20: Full body shell

At this point the car's outer shell is almost complete, aside from side view mirrors and some skirts and decals there needs to be a bottom half and innards so that when brakes, tires and other features are added they fit correctly and it helps giving the model it's overall look rather than looking like an empty eggshell.

Because the cars bottom and underparts will hardly ever be visible, I'm going to leave them almost completely without detail as the only time they would be visible was if the car was turned upside down or viewed from beneath. In this situation I didn't want to sacrifice any more processing power for detail that doesn't haven to be represented in a 3D space when most of it could be done with a good texture and bump mapping. For the wheel arches a simple space that's enough to clear the wheels and brakes even when turning is enough, detail isn't really needed.

For the bumpers air scoops and interior a simple floor was created as well as door panels and a straight cube dashboard to accommodate the necessary gauges in racing games. I create these as separate objects so that if someone wants to continue creating an interior for the car they can.

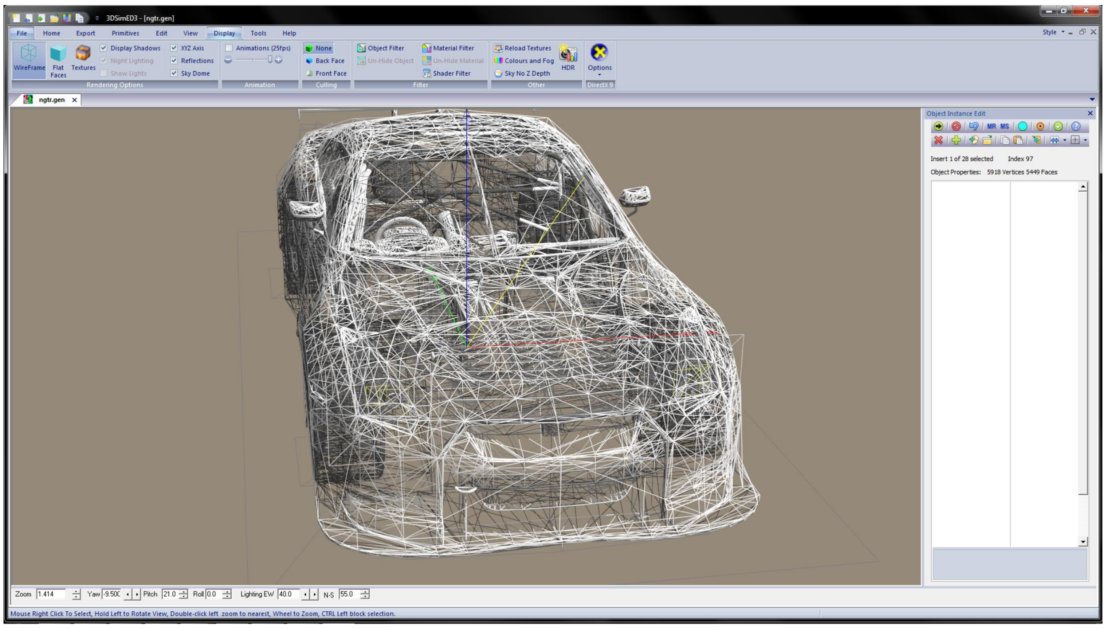
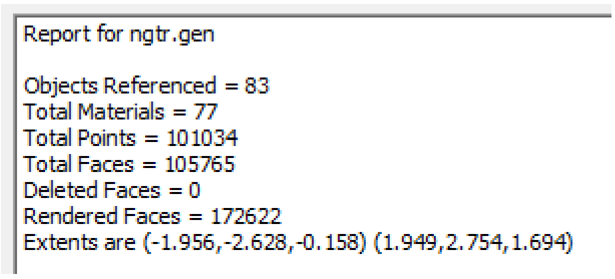


FIGURE 21: Polygon model from rFactor 2

In it's finished form the exterior of the car sits at 236144 triangles which is almost double the number of triangles that modern day racing simulator games like rFactor 2 use. So the model has plenty of detail that can be shed to be used in a modern game.



```
Report for ngtr.gen

Objects Referenced = 83
Total Materials = 77
Total Points = 101034
Total Faces = 105765
Deleted Faces = 0
Rendered Faces = 172622
Extents are (-1.956,-2.628,-0.158) (1.949,2.754,1.694)
```

FIGURE 22: Polygon count from rFactor 2 model

3.4 Finishing the model to a usable asset

Making sure that the model doesn't have any hidden geometry or surfaces that unnecessarily increase the polygon count of the model or cause geometrical bugs in the game environment. Blender has a few easy ways of doing this, selecting all faces, remove doubles, reselect interior faces and delete those. The number of triangles or quads depending on the game engine defines how many polygons the model has. The polycount of model is the total number of these triangles or quads that it takes to draw the model in 3D space. The higher the polycount for model, the longer it takes for the system to render it. The shorter rendering times, the better performance you get. All the parts of the car can be separated into their own objects in the model, such as side view mirrors, bumpers, lights and wheels. This allows end-users to have these parts fall off in damage modelling and remove unwanted parts of the cars if they wish to modify it further.

For wheels and tires generic looking circles were created using Blenders circle tool, adding some width and thickness to create the inner rim. Tires are just copies of the wheels with a larger circumference with additional thickness and bevel on the edges to give them a round look. For side view mirrors generic models were used that look close to the original ones as they widely vary depending on the domestic market model. The mirrors are close resemblance of Japanese domestic market model that's slightly bulkier but has more aerodynamic look to it.

While in Blender it's important to assign all objects with their own material, these materials define how light is reflected from them and how they cast shadows, sometimes these values can be modified unrealistically. (Lehtovirta & Nuutinen 2000, 39.)

Now that all the duplicates and hidden faces are removed from the car it can be rendered to see how it would look in an actual game environment.

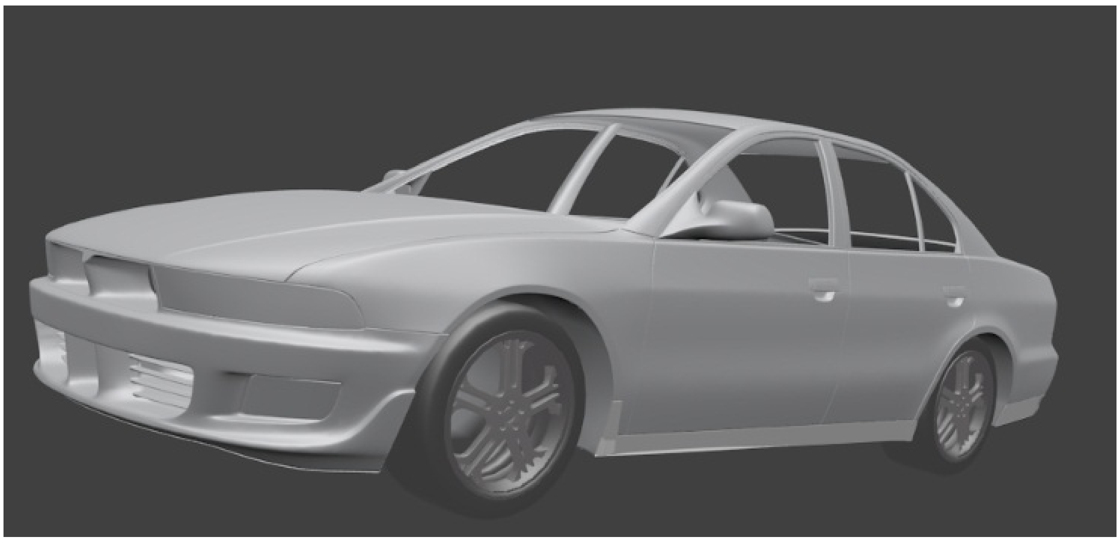
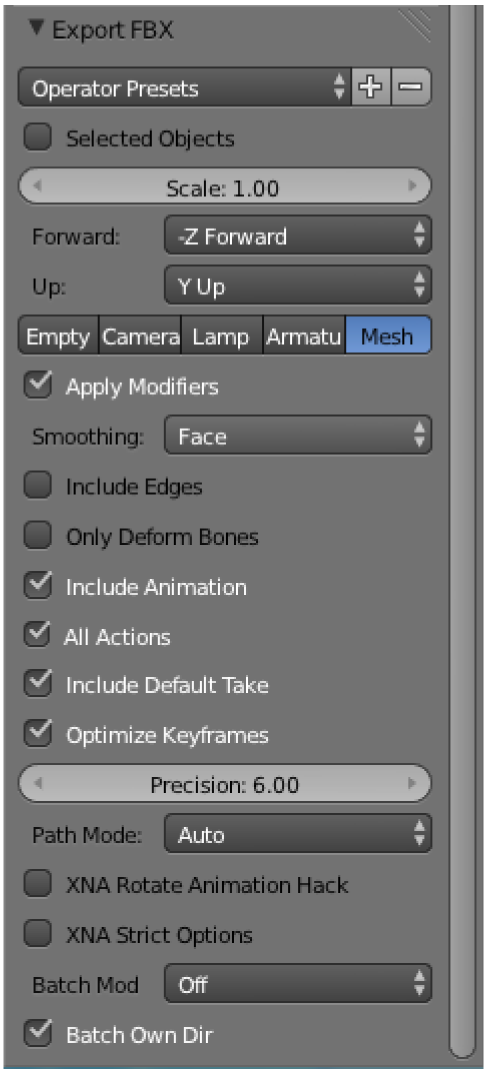


FIGURE 23: Finished rendering of the model with all objects

For rendering Blenders 3D-camera was used, it is almost like a virtual copy of a real camera, it saves the view it sees. The camera has several modifiers and adjustments almost like a real camera. When the model is set to its place the camera is set to a desired angle and position for rendering. (Wissler 2013, 5.)

3.5 Transferring the created model

Using Blenders import tools and settings to create an import-ready asset to be used in Unity game development environment. Unity game engine natively supports .blend models but having the model in multiple formats helps as all games don't support .blend or they need to be exported to an entirely different format. A popular file format that's almost universally supported is .fbx. For exporting .fbx files you need to set certain parameters for the model to work properly. Scaling is set to 1.0 as the models scale doesn't need to be changed before it's in the game engine. Forward and up refers to the models axis compared from Blender to the axis setup that the game development environment uses, axles were set to -Z forward and Y up as they are reversed in Unity compared to Blender. Next on the menu of 5 different selections is the definition of export targets that get exported to the file. For this model only mesh is chosen as it's the only part of the project that needs to be exported. Tick box with "apply modifiers" is self explanatory, this defines if the exported


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

☐ Only Deform Bones

✓ Include Animation

✓ All Actions

✓ Include Default Take

✓ Optimize Keyframes

Precision: 6.00

Path Mode: Auto

☐ XNA Rotate Animation Hack

☐ XNA Strict Options

Batch Mod: Off

✓ Batch Own Dir

FIGURE 24: Blender export options

model will include any modifiers applied to the model, as I used mirroring and duplicate modifiers it's applied. Include edges means all the orphaned edges that don't serve any purpose and only taking extra time to render, I leave it off. Deforming bones is not needed as the model doesn't have a skeleton or bone structure. I included animations as this can sometimes cause issues with models in Unity if no animations are present, even when the model doesn't include. All actions means all actions that affect the models skeleton, while not required in this case I use it as some game development environments require at least an empty set of data of this, the same goes for using optimised key frames. Precision is just a given value of comparing double key frames and the sensitivity of the procedure. Rest of the given export options are XNA-specific or just exporting the model to it's own folder.

When the model is exported it can simply be drag and dropped to a game development tool like Unity where it shows up as a ready made asset with all parts and materials separated that you used in Blender.

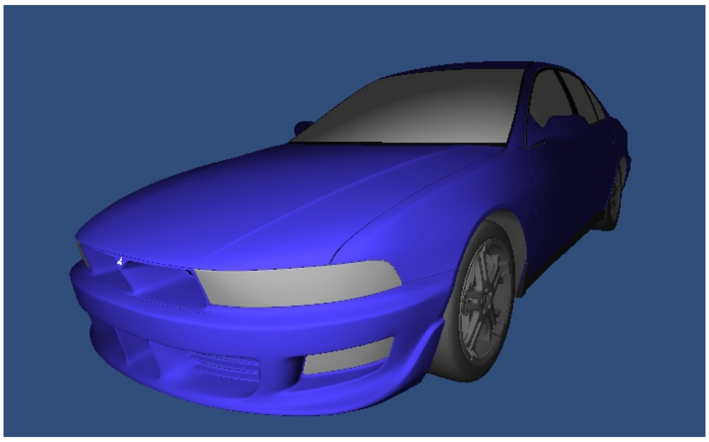


FIGURE 25: Finished model in Unity

4 FINISHING WORDS

3D-graphics have come a long way from portraying simple mazes or just lines drawn in empty space to represent a space station. They are now apart of our every day life and games have become more mainstream media because of it. Fully modelled and simulated cities and worlds that you can interact with are done with thousands of man hours sinked in to them. Seeing how 3D graphics have evolved rapidly in the past 20-years it's not hard to see that a good skillset of 3D modelling can be very useful for anyone who studies technology or plans to employ themselves in the field.

The procedure of modelling a 3D model of a real world model can be daunting at first but once the work is started with good reference pictures. Having good reference pictures saves a lot of time in the modelling process as it's the main source of reference that you have. I learned that using several days for editing the photos to a more usable form saves you a lot of headache when modelling as I often went back to edit the reference material because I wasn't happy with it.

For the modelling process I found that a good workflow is the key feature to quickly model and get satisfying result. Blender has a good variety of hotkeys that are customisable and using them efficiently greatly speeds up the workflow without having to go through menus and navigating for one certain function that you should have at the end of your fingertip. I went as far as making my own keyset where I used W, A, S and D to move around and extra mouse buttons to replace the ones that I removed. At some point I realised that I didn't need to move my hand at all from the keyboard or mouse for several hours to keep working. Also creating a certain recognisable part of the model helped motivate me and gave scale for smaller details that I could add later on as I could see the model coming together at a fast pace with satisfying quality.

For using the model in game environments it was rather easy to export with correct settings as the community around Blender and Unity are quite vast, they have readily available guides and scripts that helped a lot with the work.

SOURCES

Gahan, A. 2010. 3D Automotive Modeling. Taylor & Francis
Vaughan, W. 2012. Digital Modeling. New Riders
Simonds, B. 2013. Blender Master Class. William Pollock

Lehtovirta, P. & Nuutinen, K. 2000. 3D-sisältötuotannon peruskirja. Jyväskylä: Docendo.

Wissler, V. 2013. Illuminated Pixels: The Why, What, and How of Digital Lighting. USA: Cengage Learning PTR.

Blender manual. Retrieved 4.11.2015

http://www.blender.org/manual/

Blender history. Retrieved 4.11.2015

http://www.blender.org/foundation/history

Stories from the Mazer War 30 Year Retrospective. Retrieved 7.11.2015
http://www.digibarn.com/history/04-VCF7-MazeWar/stories/colley.html

Educational Feature: A History and Analysis of Level Design in 3D Computer Games. Retrieved 22.10.2015

http://www.gamasutra.com/view/feature/2674/educational_feature_a_history_and_.php

Electric Escape: The Army Battlezone Q & A. Retrieved 21.10.2015

http://web.archive.org/web/20071031094945/http://www.electric-escape.net/node/561

Bradley Trainer. Retrieved 21.10.2015

http://www.safestuff.com/bradley.htm

The International Arcade Museum. Retrieved 21.10.2015

http://www.arcade-museum.com/game_detail.php?game_id=9063

American McGee on Quake. Retrieved 20.10.2015

http://www.quaddicted.com/interviews/americanmcgee

Techspot: The History of the Modern Graphics Processor. Retrieved 20.10.2015

http://www.techspot.com/article/650-history-of-the-gpu/

IGN: The History of Grand Theft Auto Retrieved 19.10.2015 http://www.ign.com/articles/2008/03/28/ign-presents-the-history-of-grand-theft-auto?page=4

Softpedia: Adobe Photoshop. Retrieved 1.11.2015

http://www.softpedia.com/get/Multimedia/Graphic/Graphic-Editors/Adobe-PhotoShop-Trial.shtml

Unity: Information. Retrieved 1.11.2015

http://unity3d.com/unity