CRC Press
Taylor & Francis Group

Jingtian Li
Kassandra Arevalo
Matthew Tovar

Creating Games with Unreal Engine, Substance Painter, & Maya

Models, Textures, Animation, & Blueprint

Creating Games with Unreal Engine, Substance Painter, & Maya

Creating Games with Unreal Engine, Substance Painter, & Maya

Models, Textures, Animation, & Blueprint

Jingtian Li, Kassandra Arevalo, and
Matthew Tovar

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2021
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 978-0-367-51267-5 (hbk)
ISBN: 978-0-367-51263-7 (pbk)
ISBN: 978-1-003-05310-1 (ebk)

Typeset in Myriad Pro
by codeMantra

To our amazing colleagues, supportive family,
and my beautiful fiancée Tong.

– Jingtian Li

To my family and colleagues. Thank you for all the support.

– Kassandra Arevalo

Dedication to my parents, Alejandra & Manuel Tovar.

– Matthew Tovar

Contents

Acknowledgments xvii
Authors xix
Introduction xxi

Chapter 1: Maya Modeling 1
- Basics of Navigation 1
- Rendering 2
- What is a 3D Model? 3
- Translation 3
- Anatomy of a Model 4
  - Edge 4
  - Vertex 4
  - Face 4
  - Object Mode 5
  - Normal 5
- Modeling Rules 5
  - Polycount 6
  - Topology 6
  - Size and Proportion 7
- Basics of Modeling 7
- Tutorial 1.1: Modeling a Security Camera 7
- Other Useful Commands 32
- Grow and Shrink Selection 32
- Extract Faces 33
- Combine and Separate 33
- Create Cables or Pipes 34
- Extrude Along a Curve 37
- Duplicate, Duplicate with Transform 38
- Duplicate Special 39
- Mirror 39
- Center Pivot 40
- Change Pivot 40
- Snapping 41
- Hide Model 41
- View Control 41
- Assignments 41
- Geometry Errors 42

Tutorial 1.2: Modular Set Pieces 43
Grid. 44
Create a Base Floor 44
Conclusion. 55

Chapter 2: Maya Set UV. 57
The UV Editor 57
UV Points 59
UV Tiles 60
Cut UV 62
The Problem 62
UV the Floor 63
Texel Density. 71
Chose the Right Texel Density 72
UV the Pod. 74
Conclusion. 89

Chapter 3: Set Texturing. 91
PBR 92
Baking 93
Tutorial 3.1: Texturing Modular Pieces. 93
The Substance Painter UI 96
Navigation 97
Light Direction 97
Ambient Occlusion 98
PBR Material Channels. 98
Generators. 100
Levels. 123
Assignment: Texturing the Rest of the
Models 129
Conclusion. 131

Chapter 4: Level Asset Creation 133
Game Engines. 134
Unreal Engine 134
Tutorial 4.1: Get Unreal Engine 4 and
Visual Studio Up and Running 134
Visual Studio 136
UI of the Unreal Editor. 137
Navigation 138
Tutorial 4.2: Export Our Assets to
Unreal and Build Our Material 139

The Material Editor 145
Color R, G, B, A Channels 146
Material Instance 150
Tutorial 4.3: Set up a Test Hallway 157
Tutorial 4.4: Create Parameters for Our Materials 166
Actors 177
Mobility. 180
Conclusion. 188

Chapter 5: Level Creation 189
Tutorial 5.1: Create the Interior of Our Level 190
Assignment 220
Tutorial 5.2: Create the Landscape of Our Level 222
The Quixel Megascan Ecosystem 224
Tutorial 5.3: Set Up Quixel Bridge 224
Tutorial 5.4: Create the Landscape Material 226
Material Function 228
Tutorial 5.5: Place 3D Assets on the Landscape 234
Conclusion. 242

Chapter 6: Lighting and Baking 243
Baking 243
Lightmaps 244
Lightmap UV. 244
Lightmap Resolution 245
Lightmap Density 247
Volumetric Lightmaps. 247
Mobility. 248
Tutorial 6.1: Optimize the Lightmap Resolution 249
Lightmass Importance Volumes 253
Lower Down Baking Quality for Quick Iteration 253
Tutorial 6.2: Adjust Lighting 256
Tutorial 6.3: Add Post Processing and Other Effects. 261
Post Process Volume 262

Assignment 263
Conclusion 264

Chapter 7: Character Modeling 265
Concept Art 266
Style Sheets 266
Workflow 267
Polycount 267
Setting Up Image Plane in Maya 267
Eyeball 269
Create the Eyelids 271
Create the Eye Socket 275
Forehead and Nose 278
Mouth 283
Rest of the Head 284
Ear 286
Neck 287
Internal Structures 289
Body 290
Hands 296
Hairs 303
Weapon 314
Final Clean Up 317
Conclusion 319

Chapter 8: UV Mapping 321
UV Mapping 322
Tutorial 8.1: Character UV Mapping 322
Mesh Inspection and Cleanup 322
Body UV 324
Eye UV 330
Hair UV 331
Garment UV 334
Conclusion 341

Chapter 9: Character Texture Painting 343
Skin Texturing 345
Hair 353
Eye 355
Upper Body 362
Pants 370
Belts, Straps, Pockets, Holster, and Boots . . . 375

Gloves. 376
Watch 383
Gun 385
Other Details 385
Export Textures. 387
Conclusion 389

Chapter 10: Rigging 391
Joint Behavior 392
Joint Placement—Hip, Spine, Neck, and Head. 393
Tutorial 10.1: Create the Joint Chain for Our Character 393
World Joint. 396
Joint Placement—Left Arm 398
Joint Setup—Right Arm 405
Joint Setup—Legs. 406
Foot Roll Rig. 410
Setting Up the Foot Hierarchy. 412
Tutorial 10.2: Bind and Paint Skin Weighting. 413
Painting Skin Weights. 414
Mirroring the Skin Weights. 419
Copying the Skin Weights. 420
Tutorial 10.3: Set Up Arm Controls. 422
Constrains. 425
IK Arm Setup 425
Tutorial 10.4: Finger Controls. 427
Tutorial 10.5: Clavicle and Body Controls. 429
Gun Joint. 431
Final Hierarchy 432
Conclusion 434

Chapter 11: FPS Animation in Maya 435
FPS Animation Overview. 435
Referencing the Character Rig 436
Save Files. 437
Display Layers 437
Camera Configuration 440
Game Animations 442
Creating a Pose. 443

Weapon Movement Simplified 444
Two-Handed Weapon Setup 446
Frame Rate 449
Idle Animation. 449
Cleaning Up Odd Jitters. 450
Ease-In's and Ease-Out's 451
Graph Editor. 451
Keywords Aside 453
Attack Animation 453
Walk Animation 455
"Got Caught" Animation 456
Keywords Aside 458
Reload Animation 460
Considerations and Conclusion 462

Chapter 12: Unreal Character Asset Creation. 463
Tutorial 12.1: Character Asset Import 464
Skeletal Mesh, Skeleton, and Physics Asset 467
Subsurface Scattering 470
Tutorial 12.2: Export FPS Animations 473
Bake Animation 473
Tutorial 12.3: Motion Captured Data. 476
Tutorial 12.4: Animation Retargeting 480
Conclusion 486

Chapter 13: Basics of Programming 487
Relevant Programming Languages (From Hard to Easy) 488
C++ 488
C#. 489
Python 489
Blueprint 489
Tutorial 13.1: Create an UI 489
Event Graph 491
Execution Pin and the Order of Execution 493
GameMode 494
Functions 495
Machine Code, Source Code, and Compiling 496

Tutorial 13.2: Refactoring Our Load Level Mechanic 498
Variable 499
Variable Types 499
Built-In Variable Types 502
Custom Variable Types 503
Class 503
Tutorial 13.3: Make a Sliding Door Class . . . 505
Parent Class 506
Character 510
Collision 520
Casting 524
Conclusion 525

Chapter 14: Player Character 527
Tutorial 14.1: Create the First-Person Shooter Character 528
Pawn and Character 529
Roll, Yaw, Pitch 536
Controller 536
PlayerController 536
AIController 537
Tutorial 14.2: Set Up the Animation 539
Animation Blueprint 540
Event Graph 540
Animation Graph 541
Vector 541
State Machine 544
Game Mode 548
Player Controller/Controller/AIController 548
Actor 549
Pawn 549
Character 549
Skeletal Mesh 549
Animation Blueprint 549
Widget Blueprints 550

Chapter 15: Weapons 551
Tutorial 15.1: Create a Base Weapon Class 553

Socket. 554
Owner 557
Instigator. 557
Persona 563
Animation Montage 566
Comment Box 577
Tutorial 15.2: Weapon Attack Cooldowns. 582
Enumeration 582
Tutorial 15.3: Weapon Damage. 586
Array. 587
Loop 588
Conclusion 613

Chapter 16: Health and Damage 615
Tutorial 16.1: Create a Health Component. 616
Actor Component 616
Tutorial 16.2: Character Hit and Death . . . 621
Tutorial 16.3: Health Regeneration 629
Conclusion 633

Chapter 17: Inventory and UI 635
Tutorial 17.1: Weapon Pickup 636
Tutorial 17.2: Weapon Switching. 642
Tutorial 17.3: Create the In-Game Weapon UI 649
Widget Switcher. 649
Canvas Panel 649
Anchors 651
Alignment. 654
Tutorial 17.4: Create the Health Bar 663
Tutorial 17.5: Create the Pause and Game Over UI 672
Assignment 679
Conclusion 680

Chapter 18: Security Camera. 681
Tutorial 18.1: Implement an AISeer in C++. 682
The Header and the Source File 687

Tutorial 18.2: Create the Blueprint Version of AISeer by Inheriting from the C++ Version. 694
Tutorial 18.3: Create a Security Camera . . . 699
Tutorial 18.4: Create a BP_Monitor Class that Shows What the Camera Sees 710
Structure 712
Conclusion 715

Chapter 19: Patrolling AI 717
Tutorial 1: Create the Patrol AI Character 718
Tutorial 2: Create AI Controller and Behavior Tree for the Patrol AI. 725
Blackboard and Behavior Tree. 725
Cache Pose 743
Conclusion 749

Chapter 20: Boss 751
Tutorial 20.1: Create the Boss Class 752
Tutorial 20.2: Boss Attack. 757
Tutorial 20.3: Boss Death and Winning. . . 763
Conclusion 767

Chapter 21: Audio and VFX 769
Tutorial 21.1: Add Audio to the Game 770
Tutorial 21.2: Add Extra VFX to the Game 778
Conclusion 783

Chapter 22: Packaging 785
Tutorial 22.1: Package the Game for Windows 786
Conclusion 790

Index 793

Acknowledgments

It takes the effort and support of many people to finish a book like this. We would like to say special thanks to everyone who contributed to this book.

First, many thanks to my most supportive mentor and colleague Adam Watkins. This book becomes much more organized, precise, and informative than it would be without his guidance.

Also, thanks to our Matthew Tovar and Kassandra Arevalo for writing the fantastic rigging and animation chapters; this book would not be complete without their effort.

Finally, special thanks to my amazing sister, Rui, and many other family members of our team. Your support has been of enormous help.

Authors

Jingtian Li is a graduate of China's Central Academy of Fine Arts and New York's School of Visual Arts, where he earned an MFA in Computer Art. He currently is an Assistant Professor of 3D Animation & Game Design at the University of the Incarnate Word in San Antonio, Texas.

Kassandra Arevalo is an instructor of 3D Animation & Game Design at the University of the Incarnate Word in San Antonio, Texas. She previously worked as an animator at Immersed Games.

Matthew Tovar is an industry veteran animator. He has worked at Naughty Dog, Infinity Ward, and Sony Interactive on such games as The Last of Us, Call of Duty: Modern Warfare, and, most recently, Marvel's Avengers with Crystal Dynamics. He is an Assistant Professor of 3D Animation at the University of the Incarnate Word in San Antonio, Texas.

Introduction

Making a game of their own is always the dream of many people since they are teenagers. As new technology emerges, that dream becomes more and more accessible each year. There is an exponential growth of game releases over the past decade. About 10,000 games were released on Steam in 2019, and around 1000 games per day were released on mobile devices.

One of the reasons that more games are coming out is because there are more and better tools to make them. With the release of free game engines like Unreal Engine and Unity, to name a few, start making games is within the reach of everyone. The competition between the game engine developers pushes them to implement new features every year, and we have seen a burst of improvements to the tools.

Outside of the game engines, new developments are happening in every corner of the game industry. Softwares like the Substance Suite solve the texturing process in innovative ways. Newer generations of hardware like Nvidia RTX and Playstation 5 push realtime rendering to new heights. And new categories of devices like Oculus Rift, Steam VR, and Microsoft Hololens are pioneering new user experiences. To add on top of that, services like Quixel Megascan and Adobe Mixamo are providing libraries of reusable assets that significantly improve productivity.

It is the best time than ever for anyone who wants to dip into a game development journey. However, making a game is never an easy task. It requires all kinds of talents to put together a working game that has amazing visuals, engaging gameplay, immersive audio, and an overall well-balanced system. There are many sources where you can learn different ingredients of game development, but only fewer sources explain the whole recipe. This book is dedicated to cover the entire process of making a game, from making assets to programming, and all the way to package a complete game.

Who's It For?

This book is designed for beginners who want to start their game development journey and are unsure where to start and which direction to go. As a reader, you are going to jump into a well-organized learning track that guides you through all aspects of game development. It also shields you from noises and focuses on the fundamentals, which gives you a solid foundation and is able to branch out to nitty-gritty details without losing the whole picture.

For any game enthusiasts or students, this book is a perfect fit to get started with game development. For teachers, this book offers a well-structured solution for your curriculum. For anyone who wants to utilize the game engine for interactive products, this book covers the skill you need extensively as well.

What Does this Book Cover?

This book covers all aspects of game development that includes but not limited to:

Environment Modeling

Environment modeling is the process of making 3D models for environments. We are going to cover what is a 3D model, how to make them, and how to optimize them for your game.

Character Modeling

Character modeling is the process of making 3D characters. We will cover how to approach organic shapes with additional modeling methods.

UV Mapping

We are going to learn how to create a 2D coordinate of a 3D model to map textures to the model. The process we call UV Mapping.

Texturing

Texturing is the process of defining the color and all other aspects of the appearance of the model.

Rigging

Rigging is a technical skill to add skeleton and controllers to animate the character.

Character Animation

We are going to cover the techniques and theories to animate characters.

Game Engine Lighting and Baking

We are going to practice workflows on lighting an environment, which includes how lights work in the game engine, and technical details of baking the lighting.

Game Programming

We are going to cover programming languages, theory, and practices to create gameplay.

We will also explore audio and VFX solutions and many other small details you need to know to create a game. At the end of this book, you should have everything you need under your belt to start making your next awesome game!

Final Notes

It is critical to point out that game development is time-consuming. Please dedicate your energy and time to the learning process, and don't easily give up on any obstacles. With the internet at your fingertip, you can find solutions for just about anything.

It is also important to acknowledge that tools change all the time, and you should always learn new stuff and explore new ideas. Please take away the theories we cover in this book, but don't be religious of the tools we use.

Alrighty, we know you are tired of reading introductions, and many people jump over it. It is time that we start this fantastic journey and start making some awesome games!

Jingtian Li
May 9, 2020
San Antonio, TX, USA

CHAPTER 1

Maya Modeling

We will jump into the production by discussing modeling. 3D models are the foundation of the graphics of modern games. They encompass the environment and characters you see on the screen. An eye-catching visual is one of the key components for a game to succeed. In this chapter, we will discuss in detail about how they are built.

Basics of Navigation

Autodesk Maya will be our tool of choice for modeling. It is not the best modeling tool on the market, but it is the most used over the entire production pipeline, especially

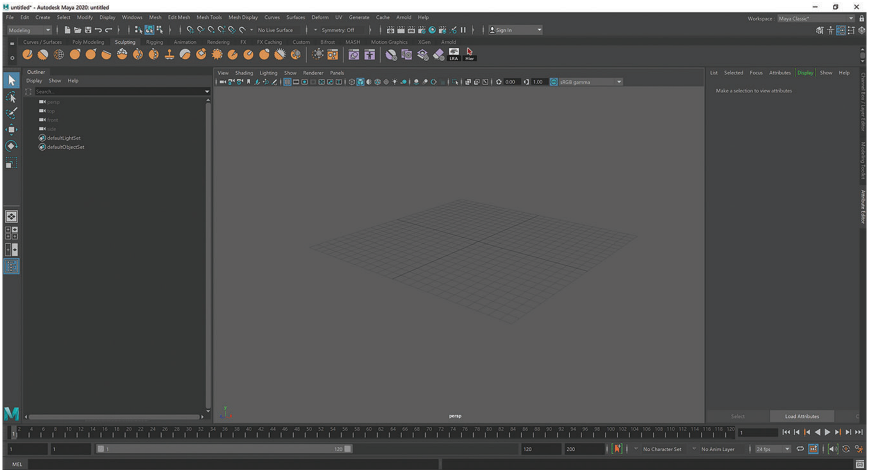

untitled* - Autodesk Maya 2020: untitled
File Edit Create Select Modify Display Windows Mesh Edit Mesh Mesh Tools Mesh Display Curves Surfaces Deform UV Generate Cache Arnold Help
Workspace : Maya Classic*
Modeling
No Live Surface
Symmetry: Off
Sign In
Curves / Surfaces Poly Modeling Sculpting Rigging Animation Rendering FX FX Caching Custom Bifrost MASH Motion Graphics XGen Arnold
LRA
Hier
Outliner
Display Show Help
Search...
persp
top
front
side
defaultLightSet
defaultObjectSet
View Shading Lighting Show Renderer Panels
0.00
1.00
sRGB gamma
List Selected Focus Attributes Display Show Help
Make a selection to view attributes
Channel Box / Layer Editor
Modeling Toolkit
Attribute Editor
persp
Select
Load Attributes
1 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100 102 104 106 108 110 112 114 116 118 120
1
1
1
1
120
120
200
No Character Set
No Anim Layer
24 fps
MEL

FIGURE 1.1 Maya's user interface. The origin is the area at the center of the grid.

for animation. So, let us get Maya up and running on your machine. The UI (user interface) will look like Figure 1.1. The large region in the middle of the UI is the viewport; this is where we see our models. It is currently empty, with just a grid in the middle to indicate the center of the world. The center of this grid is called the origin.

To Navigate around the viewport, hold down Alt key and drag the left-mouse button to look around the viewport. To zoom in and out, hold down the Alt key and drag the right-mouse button. To pan left and right, hold down the Alt key and drag the middle-mouse button.

A 3D space has width, height, and depth, each represented on three axes called the X, Y, and Z axes. The lower left corner of the viewport shows the directions of these axes.

Rendering

The shape is drawn by the Graphic API, but the lighting is calculated by the Fragment Shader written by the game engine programmer. It is a complicated process, and we do not have to understand the details and math behind it. It is enough to know that the *renderer* is the tool drawing whatever you see on screen. Maya's interactive renderer (that shows you what is currently in your scene) is called Viewport 2.0.

What is a 3D Model?

In the menus, go to Create->Polygon Primitives->Plane. This will create a shape in the middle of the viewport. On the right side of the UI, look for the Channel Box. This is a brief list of essential attributes we can tweak for the object. Under the INPUTS section, click on the polyPlane1 to open it and change the Subdivisions Width and Subdivisions Height to 1 to make the plane only one polygon (sometimes called a "face").

What we are seeing now is the building block of any model – a face with four corners that we typically call a rectangle in geometry classes; in 3D graphic terms, we call this a *quad*. Any complicated shape can be composed by assembling many quads together to create 3D forms.

Translation

On the right side of the UI, there is a column of manipulation tools. You can try and use the Q, W, E, and R buttons to switch between these tools: Q for the select tool, W for the move tool, E for the rotation tool, R for the scale tool.

To select the model, simply left-click on it or drag a selection box over it. To deselect the model, click in the empty space, or hold down Ctrl and click on the model, or drag a selection box over it.

To move the model, after selecting it, hit the W button. This will display new handles (called gizmos) that will allow you to move the object. Try dragging the various arrows to move it only along a particular axis. Look carefully at the gizmo, and you will see squares that can be dragged to move it along two axes at the same time; you can even drag the cyan square in the middle to move it freely along all axes in the 3D space.

To rotate the model, after selecting it, hit the E button, drag the circles on the gizmo to rotate it around different axes. You can also drag the yellow one on the outside to rotate it around a plane that is perpendicular to the angle of the viewport.

To scale the model, hit the R button, and drag the various boxes to scale it along their respective axes. You can also drag the various squares to scale it along two axes at the same time; you can even drag the yellow box in the middle to scale it up along all axes, essentially making it bigger.

There are more tricks about this sort of manipulation that we will cover later on when we jump into modeling.

Anatomy of a Model

Edge

Hold down the right mouse button on the model, and you will see a pop-up menu we call a *Marking Menu*. Here, we can see various parts of the form we can switch to. With the marking menu active, slide up and chose Edge; the four edges around the face now appear to be in a lighter blue color. You can click on any of the edges to select them. When an edge is selected, it will be highlighted with orange color. Once selected, you can change to the Move tool (hit W on the keyboard) and drag the three arrows to move the edge along the respective direction.

Vertex

You can also hold the right mouse button again and chose Vertex. Four purple points will show up on the corner of this face. These are the vertices where edges meet. You can click to select any of them and move them around just like how you can move an edge.

Face

Hold down the right mouse button again and chose Face; you can now select the face and move it around as well.

Edge, Vertex, and Face are the three important elements of any 3D form's polygons. We can add and tweak these elements to create any shape we want.

Object Mode

Hold down the right mouse button again. This time, we chose Object Mode. This will allow us to move the model altogether. Object, Vertex, Edge, and Face are the primary modes we keep switching between while making a model.

Normal

Use the alt-left, -middle, and/or -right mouse drag to rotate your camera to look at the bottom of the face. You can see it appears to be black. Any face in 3D has a front side and a back side. The front side will appear normal, while the back side will be black or invisible (depending on the rendering engine). Maya makes the back of the face black in the default setting. To view this, using the top menus find Display->Polygons->Face Normals.

Press the Q button to switch to the select mode to get rid of the Move tool handles. We can now see a green line sticking out from the front face of the model. In general, the front of the polygon should face outwards. It is possible though to render both sides of the face. Consider a situation like rendering a piece of paper. Here we would definitely want both sides of the polygons seen, but otherwise we want to avoid rendering both sides, if possible, to avoid performance overhead. Since games have to draw many frames each second, we want to always ensure that we aren't drawing anything we don't need to (Figure 1.2).

Modeling Rules

Before we start modeling anything, let's talk about a few important rules when modeling for games.

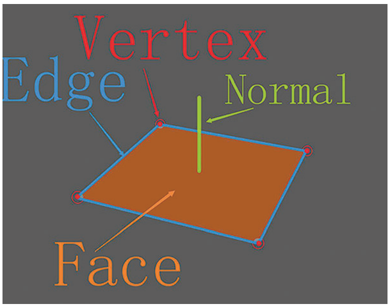


FIGURE 1.2 The elements and normal direction of the quad.

Polycount

Each of those four-sided faces we looked at earlier can be triangulated into two triangular polygons. We typically use the number of *triangles* of a model as the number for polycount, even we use quads to make a model. The reason we use the number of triangles instead of quads is because a triangle is guaranteed to be a flat surface, while this is not guaranteed for a geometric figure with more than three vertices. Thus, the rendering process uses triangles as the basic rendering unit. Fewer polygons means your game is easier to run (less data); so find the balance of including the needed number of polygons to describe a shape, but not extras.

Topology

Topology is how the faces are laid out on the model. Use quads if possible, because quads have a strong sense of directionality and are easy to represent shape evolution and deformation. We want the flow of the quads to represent the change of the surface. Figure 1.3 shows how topology is critical for deforming a face. The loops of faces around the orbicularis muscle, nasolabial fold, and orbicularis oris create an essential structure to support the facial expression. Long story short, topology is for the purpose of better representing the shape of the model and supporting the deformation for animation.

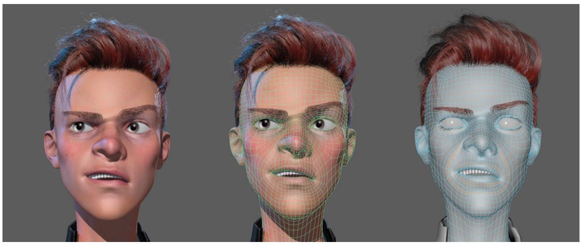


FIGURE 1.3 Effective topology (the flow of polygons) is critical to support the deformation that will come later in animation.

Size and Proportion

Size is a critical aspect in 3D modeling, no matter how detailed a model is. If the size or proportion is off, the model will never look right. In Maya, the default unit is a centimeter. This is the unit across many popular programs including Maya, Unreal Engine, Blender, etc. Other software, like Unity, use the meter as the default unit, but converting between the two scales is an easy math. One should always check sizes and dimensions to ensure things will work with physics simulation, rendering, and animation; for example, if you are modeling a staircase, then you have to know that the general height of a stair is around 18 cm and the depth is 28 cm. Converting to the right scale as you move assets from Maya to your game engine is trivial, but focus on building assets in Maya at the correct scale for its unit size (centimeters by default).

Basics of Modeling

We will jump into modeling right away and introduce various tools along the way. Keep in mind that the only way to improve is to practice; there is no shortcut to get better.

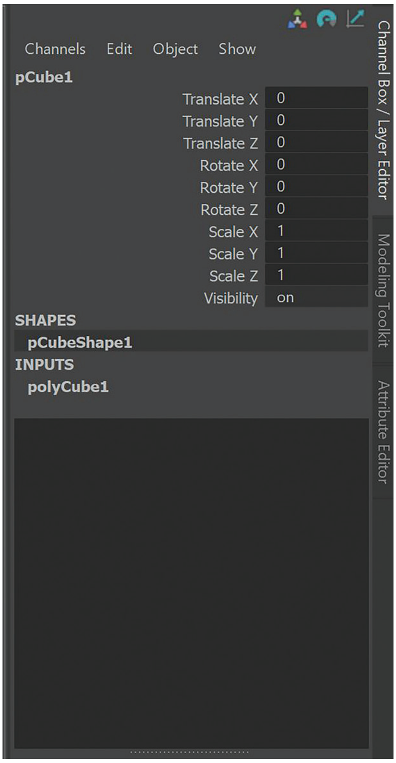
Tutorial 1.1: Modeling a Security Camera

Step 1: Basic Shape. Choose Create->Polygon Primitives->Cube. This will create a cube at the origin. This cube is also referred to as box by 3D artists. In fact, what we are doing now has a nickname called box modeling.

Tips and Tricks

In Maya, with nothing selected, you can hold down the Shift+right mouse button to pull up a type of menu called a *marking menu*. If you do this in the Viewport where there is no other object, the marking menu that will show up allows for the creation of new object. You can use this to create a cube in the same way as Create->Polygon Primitives->Cube. Learning shortcuts like this will drastically improve your modeling speed.

Step 2: Dimension. With a bit of research, you will find that a common security camera is about 18 cm long, 10 cm high, and 10 cm wide. Make sure that the box is selected and look to the right side of the UI. In the Channel Box (Figure 1.4),



pCube1

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1
Scale Y	1
Scale Z	1
Visibility	on

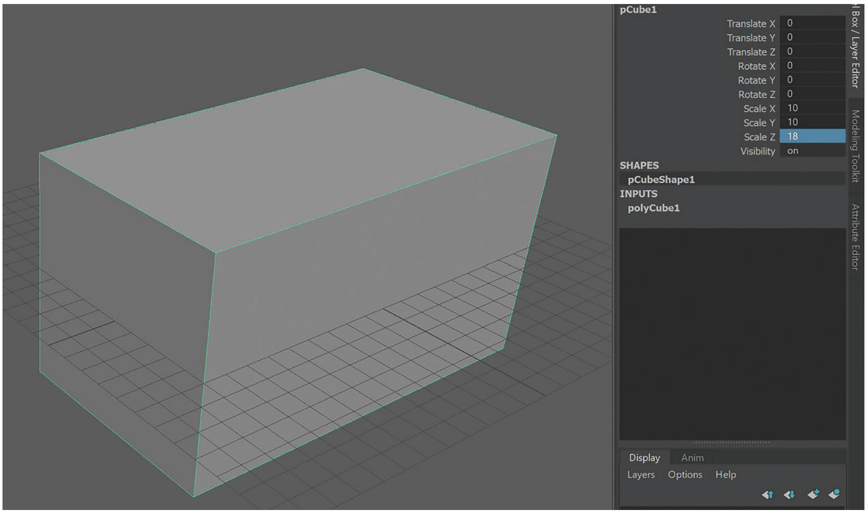
SHAPES
pCubeShape1
INPUTS
polyCube1

FIGURE 1.4 The Channel Box is at the top right of the Maya UI and allows you to change the position, rotation, and scale of a selected object.

change the Scale X and Scale Y to 10 and change the Scale Z to 18 (Figure 1.5).

Step 3: We are making a camera that looks like the one in Figure 1.6; one of the major differences between our box and the image is that the camera's corners are rounded.

Switch to edge mode (right-click and hold on the box, and choose Edge from the marking menu). Select the four edges across the length of the box (seen in Figure 1.7). Go to Edit Mesh->Bevel or press Ctrl+B to bevel these edges. This operation splits the edge you are selecting



pCube1

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	10
Scale Y	10
Scale Z	18
Visibility	on

SHAPES
pCubeShape1
INPUTS
polyCube1

FIGURE 1.5 Adjusting the size of a cube (box) via the Channel Box.

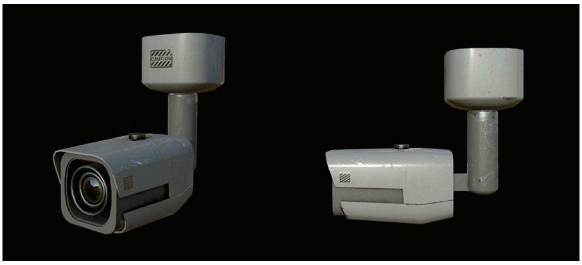
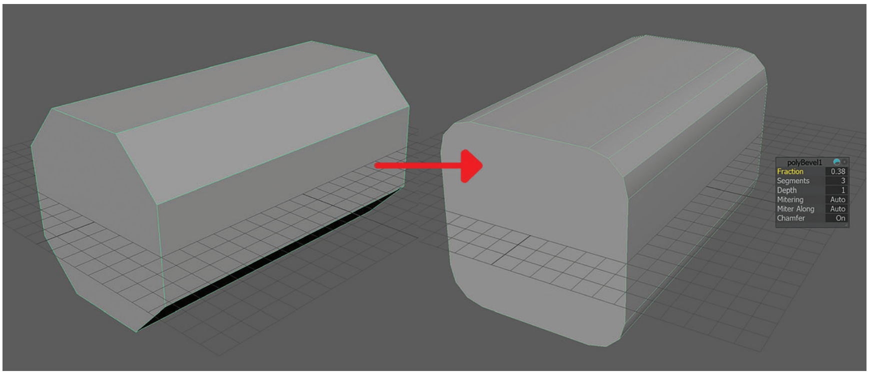


FIGURE 1.6 Our target camera.



polyBevel1

Fraction	0.38
Segments	3
Depth	1
Mitering	Auto
Miter Along	Auto
Chamfer	On

FIGURE 1.7 Using the bevel tool to round the edges of our cube.

to multiple ones. To round off these new edges, look for the pop-up menu (labeled polyBevel1) and change the Segments value to 3. Change the Fraction to 0.38 to shrink the distance between the newly beveled edges.

Tips and Tricks

To select the four edges, you can rotate the camera view to look at the side of the box, and then drag a section box over these four edges. Alternatively, you can select one of them, hold down the Shift button, and double-click the next one. Maya will select all edges that are between the same loop of faces; we call this selection of edges an edge ring.

Step 4: Soften edge. Swap out of Edge mode into Object mode by right-clicking (and hold) and choosing Object from the marking menu. Click in an empty space in the viewport to deselect the rounded cube. See a harsh line on the rounded-out corner? This is due to that edge being "hard." To soften it, swap to Edge mode and then select that edge and hold down the Shift button and double-click the next one to select the entire edge ring. Use Mesh Display->Soften Edge to make all the lines of this ring a soft edge (Figure 1.8).

Step 5: Frontal opening. Go to Face mode and select the front face of the camera. Go to Edit Mesh->

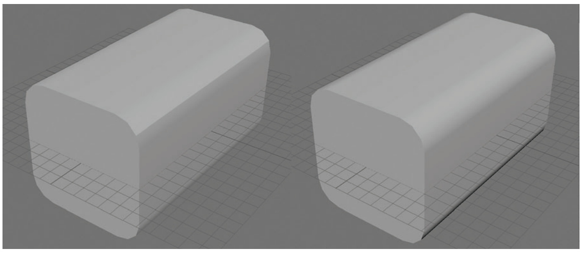


FIGURE 1.8 Softening the edges.

Extrude, or hold Ctrl+E to extrude the face. This creates another segment right at the faces we selected. Press the R button to switch to the Scale tool. Drag the yellow box in the middle of the Scale tool to scale the new face down to make the thickness of the shell. Take a closer look, and you can see the left and right contour of the opening is rounded. With the Scale tool, scale with just the red box handle (it will turn yellow when you are using it) to scale the face down across the X axis. Once done, we do not need this in the middle anymore, so press the delete button on the keyboard to delete it (Figure 1.9).

Step 6: Add Curvature to the side edges. To round the contour, we need more geometry. Go to Mesh Tools->Multi Cut. Hold down Ctrl, and hover the cursor on the side edge. Maya will give a preview of the edges that will be created if you click the mouse. Before clicking though, hold down the Shift button, to snap where the previewed ring will be created. This preview will snap every 10% across the length of this edge. Move the cursor until the preview lands at the middle of the edge, and click to finish adding the new subdivisions

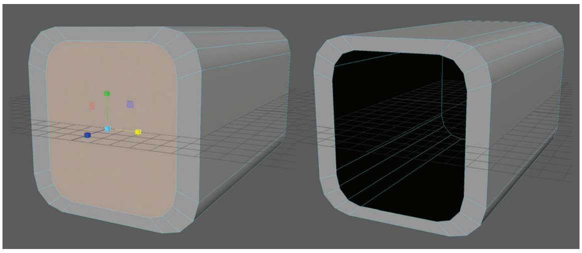


FIGURE 1.9 Using the Extrude tool to create an opening at the front of the camera.

(new edges). These edges have their tip and end connected. We call this kind of line an edge loop. Repeat and add the same edge loop on the other side (Figure 1.10).

Step 7: Turn on symmetry. Modeling is time-consuming, so we want to save time if possible. To do this, we can turn on symmetry, so we do not have to manually add the edge loop on the other side. The setting is located on the second row of buttons (Figure 1.11). By default, the setting is at Symmetry: Off. Click on the drop-down arrow on the right and choose Object X to toggle symmetry on across the X axis (Figure 1.11). After toggling symmetry on, selecting and performing commands on one side of the geometry will affect the other side.

Step 8: Add Curvature to the camera opening. Double-click on any edge of the edge loops we created in Step 6 to select the entire edge loop. Press Ctrl+B to bevel the edge loop and change the Segments to 2. Go to Vertex mode (right-click and hold on the shape, and choose Vertex from

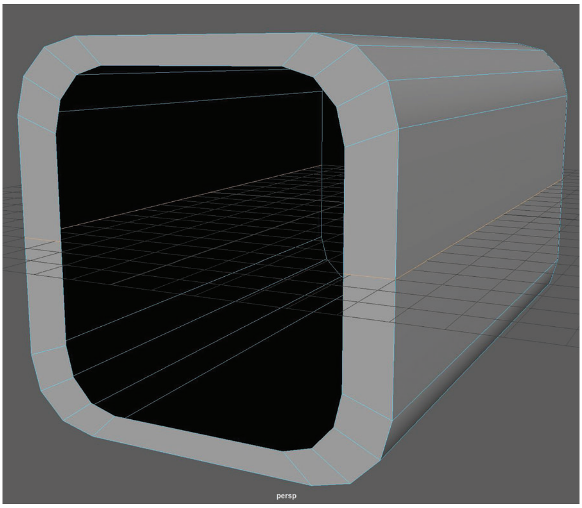


FIGURE 1.10 Added edge loops on either side of the camera chassis.

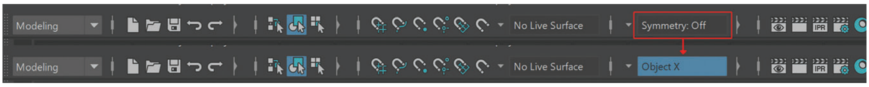


FIGURE 1.11 Turning symmetry on to allow us to mirror our modeling work.

the marking menu) and select the vertex in the middle on the edge of the hole. Use the Move tool (W) to drag it away from the center a little. Select the vertex above the middle vertex, hold down shift, and click on the vertex below the middle vertex to add it to the selection. Drag them also away from the center. Work your way around the opening and adjust the vertices until you get a proper curvature for the side (Figure 1.12).

Step 9: Extrude the inner face. Double-click on any edge of the hole to select the edge loop around the hole. Hold down shift and left-mouse-button and drag the loop inward a little; this is a quick shortcut to extrude a new ring of polygons. Select the edge ring along the newly extruded edge and hold down the Shift + right mouse button. In the resulting pop-up menu, select Soften/Harden Edges->Soften Edge; this will make the inner edges soft. This command is the same command in the Mesh Display-> Soften Edge. Hit the R button to switch to the Scale tool. Hold down Shift again and drag the yellow box in the middle to extrude a new small ring of polygons. Switch to Move tool, hold down shift and drag the new edge ring toward the back of the form to fill out the inside (Figure 1.13).

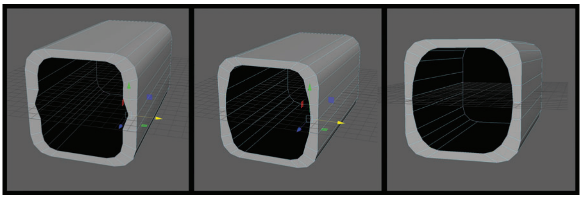


FIGURE 1.12 Using Vertex mode and symmetry to adjust the opening to create a round opening.

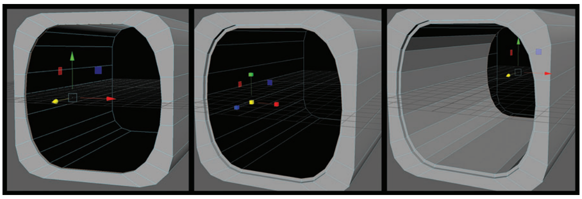


FIGURE 1.13 Using the Extrude tool (Shift-drag) to create polygons for the inside of the form.

Tips and Tricks

Shift+right mouse button is a very common shortcut. Basically, it will pull up tools or commands to the current element you have selected. If nothing is selected, doing this will pull up a wide selection of primitive polygons. Almost all commands we need can be found in this pop-up marking menu.

Step 10: Camera lens. Click in some empty area of the Viewport to deselect the camera body. With nothing selected, hold down the Shift+right mouse button and chose Cylinder. Go to the Channel box and set the Rotate X to 90. This will rotate the cylinder 90 degrees in X and lay the cylinder down. Scale and move the cylinder so that it is roughly the size of the lens of the Camera.

Step 11: Lens frontal rims. Switch to Vertex mode. Select the vertex at the center of the front faces, hold down the Ctrl+right-mouse button, and in the resulting pop-up marking menu, chose To Faces->To Faces. This will select all faces that share this vertex. Turn off the symmetry (remember up in the second row of the interface). Press R to go to the Scale tool and hold down the Shift button and drag the yellow box to extrude the face in. Using the Move tool, hold down the Shift button and drag the face back in; keep on extruding with Scale and Move tools to create all the rims of the lens (Figure 1.14).

Step 12: Bevel the rim. Select the harsh edge loops on the rims of the lens (remember, you can do this by double-clicking on an edge while in Edge mode) and press Ctrl+B to bevel them. Select all the edges in the front of the lens, do a Soften edge command to soften the edge of the lens (Figure 1.15).

Step 13: Curvature of the lens. Select the vertex at the center of the lens. Hold down Ctrl+right mouse button and chose To Faces->To Faces. Switch to the Scale tool and hold down the Shift button while you drag the yellow box to extrude the faces down to about half of the original size. Use the Move tool to drag the faces forward a little. Grab the vertex at the center again and move it forward a bit more. Select the edge loop around the center vertex and press Ctrl+B to bevel it. This will give us the curvature we need for the

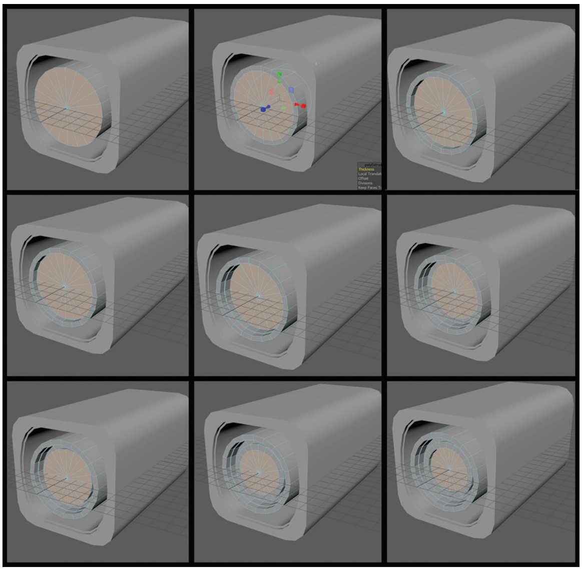
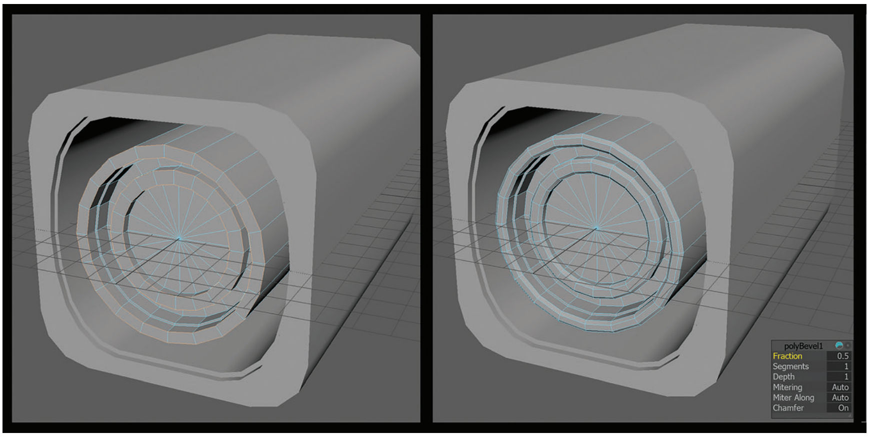


FIGURE 1.14 Using the Extrude, Move, and Scale tools to create the front rim of the lens.



polyBevel1

Fraction	0.5
Segments	1
Depth	1
Mitering	Auto
Miter Along	Auto
Chamfer	On

FIGURE 1.15 Beveling and softening the edges to create the rim of the front lens.

lens. Finally, soften the edge loops we created to make the lens feel smooth (Figure 1.16).

Step 14: Clean up history. Maya remembers everything we've done and stores this in the Input stack under the Channel Box (Figure 1.17).

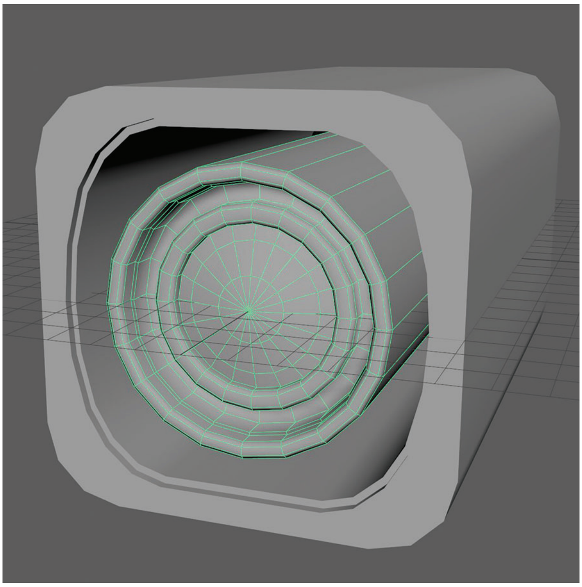
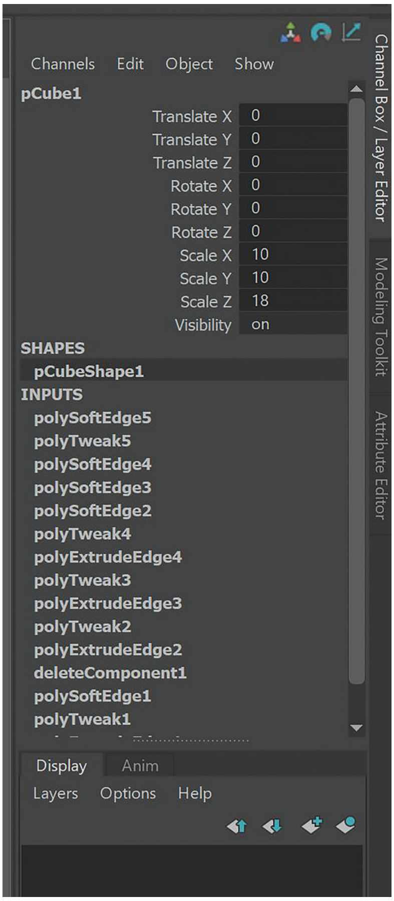


FIGURE 1.16 Finishing off the lens by adding curvature to the glass portion.

Go to Object mode and drag a big selection box to select both the shell and the lens of the camera. Go to Edit->Delete by Type->History to clean up the history. This will make all the construction history disappear (the shortcut for this operation is Alt+Shift+D). It is important to delete the history of the model regularly to ensure the model is stable and the scene is not getting heavier and heavier.

Step 15: Outer shell. Select the outer layer of faces of the lens that we made from a box. To do this, go to Face mode and grab one of the faces that goes across the depth of the model. Hold down Shift and double-click the next one to grab the whole loop across the depth of the model. Hold down Shift+right-mouse button and chose Duplicate Faces. Dragging the arrow that is facing away of the face that the arrows are sitting on, this Duplicate Faces command creates a new model from the faces selected. This allows you to shift the faces away so we can easily create a shell (Figure 1.18).



pCube1

Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 10
Scale Y 10
Scale Z 18
Visibility on

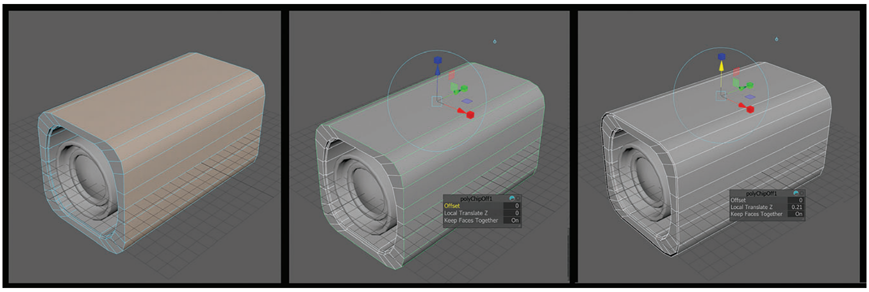
SHAPES

pCubeShape1

INPUTS

polySoftEdge5
polyTweak5
polySoftEdge4
polySoftEdge3
polySoftEdge2
polyTweak4
polyExtrudeEdge4
polyTweak3
polyExtrudeEdge3
polyTweak2
polyExtrudeEdge2
deleteComponent1
polySoftEdge1
polyTweak1

FIGURE 1.17 The Input stack of the Channel Box. This shows the History of steps created thus far.



polyChipOff1
Offset 0
Local Translate Z 0
Keep Faces Together On

polyChipOff1
Offset 0
Local Translate Z 0.21
Keep Faces Together On

FIGURE 1.18 Creating a shell by duplicating faces.

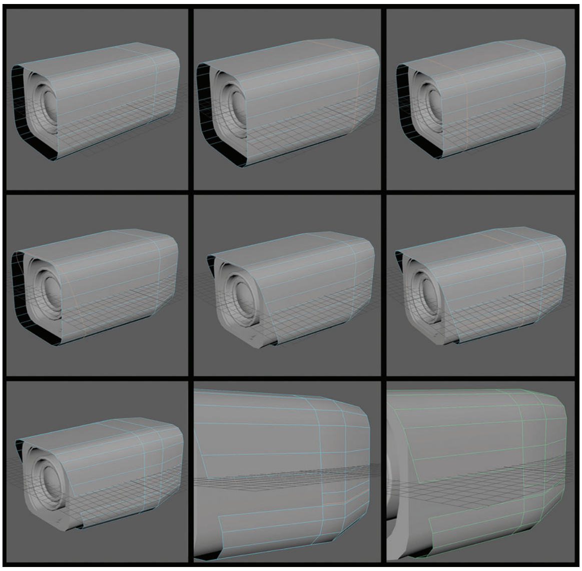


FIGURE 1.19 Try and follow this visual guide to tweak the shape to match the research.

Step 16: Tweak the shape. Figure 1.19 shows a sequence of steps using the techniques introduced in earlier steps. Try following the images to match the shape.

If you need help, the steps are: Grab the Outer shell we created in Step 15 and use the Scale tool to stretch it longer. Hold down Shift+right mouse button and chose Multi Cut. Hold down Ctrl and click to add an edge loop closer to the back end of the shell. Press Q to switch to selection tool and double-click on any edge of the newly created edge loop to select the whole loop. Scale this loop up and drag it slight down to create the wider portion of the shell. Add another loop closer to the front of the shell. With this loop still selected, press E to switch to the Rotate tool. Hold down Ctrl+Shift and rotate the loop to tilt it forward. (Note: you can see how the edge is constrained on the surface of the model when rotating, which is great to create the tilted frontal shape.) Select the front loop of faces and delete them. Toggle symmetry on and add edge loops to mark out the edge of *the opening in the middle of the shell. Select the corresponding faces and delete them. Add an edge loop really close to the edge where the seam between the upper and lower shells is. Finally, delete the face loop in-between to open the seam.*

Step 17: Upper shell hole. Add an edge loop at the center of the model. Then select the new loop, press Ctrl+B to bevel it and change the fraction to 0.32. Switch to Move tool and use the Ctrl+Shift trick to slide the edge in the center forward to mark the front edge of the opening. If you are not sure if the face is gone or not, you can go to Object mode and grab the shell and press Ctrl+1 to isolate it. You can press Ctrl+1 to toggle the isolation (Figure 1.20).

Step 18: Add thickness. Grab all the faces of the model (using Face mode and either double-clicking on any polygon or marquee-selecting). Press Ctrl+E and drag the arrow to extrude the faces out to add the thickness.

Step 19: Back arm. Create a cube. Move and scale it to create the basic shape of the back arm. Add an edge loop in the middle and bevel it to give it curvature. Don't forget to smooth the edges of the rounded back (Figure 1.21).

Step 20: Connect back arm. Grab the back arm and the inner shell of the camera. Go to Mesh->

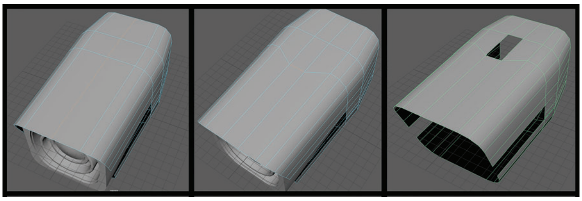


FIGURE 1.20 Creating an upper shell hole, and finally isolating just the shell.

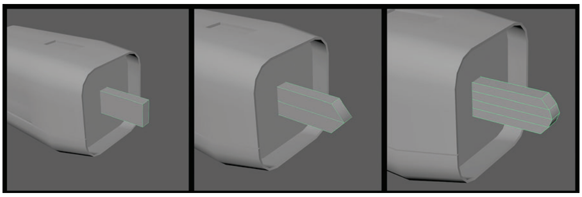


FIGURE 1.21 Creating the back arm by creating a simple cube, adding new edge loops, and tweaking those to create the desired shape.

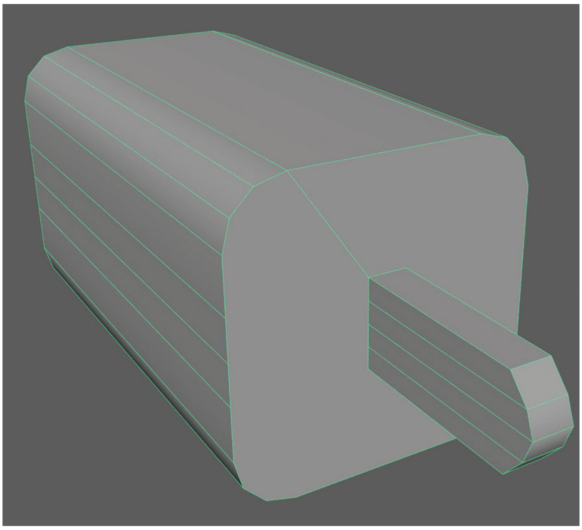


FIGURE 1.22 Using Booleans (Union) to merge two shapes together into one.

Booleans->Union. This will combine the selected meshes, blast out the overlapped part, and fuse the contact surface (Figure 1.22).

Step 21: Fix N-Gon. An N-Gon is any face with more than four edges. This can be a problem in 3D because it is unclear how the face should be divided into triangles for the rendering process. This can sometimes yield undesired output at the time of rendering. So it's best to rebuild N-Gons to either four-sided polygons (quads) or triangles. The big back face is a typical N-Gon. Switch to Multi Cut tool, and click and drag on one of the outer edges until it stops at one vertex. Click and drag on one of the inner edges until it hits another vertex. Maya will connect these two vertices with a new edge. Press the G button to commit the current operation and re-initiate the same tool again. Keep clicking and dragging to connect lines until there are no N-Gon anymore (Figure 1.23).

Why?

Notice that we had to end up with some triangles, and this is totally fine; otherwise, we need to add new edge loops

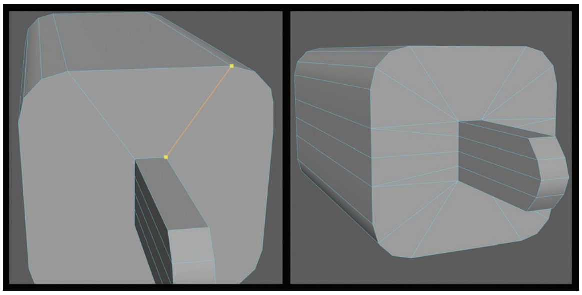


FIGURE 1.23 Using the Multi Cut tool to create new edges to rebuild the N-Gon into three- or four-sided polygons.

to the rest of the body, which takes more performance, and the render result will be the same.

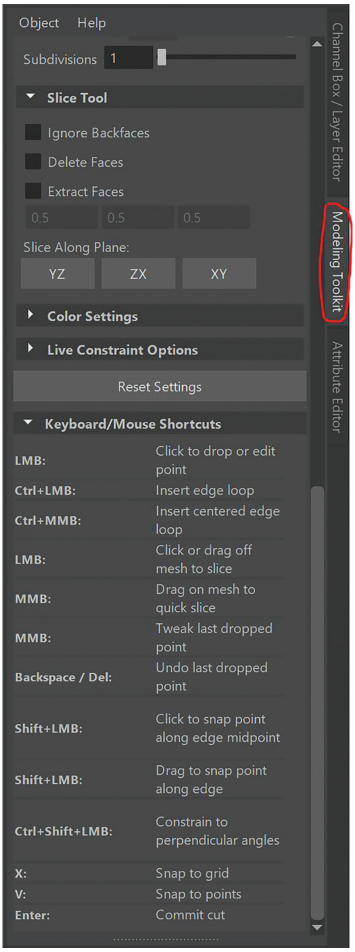
Tips and Tricks

Click on the Modeling Toolkit button (Figure 1.24) on the right edge of the UI to switch to the Modeling Toolkit. Under the Tools section, you can also see Multi Cut. Click on it to toggle it on, and various settings of the tool will appear underneath. Scroll all the way down and open the Keyboard/Mouse Shortcuts section. You can see how versatile this tool is. Experiment with these different shortcuts to speed your workflow.

Step 22: Base. Create a cube, set the Scale X and Scale Z of the box to 13, and set the Scale Y to 8. Grab the vertical edge of the cubes and press Ctrl+B to bevel them. Change the Fraction to 0.62 and Segments to 3. Move it to the back of the camera body and drag it higher (Figure 1.25).

Step 23: Base bottom shell. Select the bottom face and extrude it down. Scale the new faces down to match Figure 1.26. Next, go to Multi Cut tool, and in the Modeling Toolkit, toggle on Edge flow under the Cut/Insert Edge Loop Tool section. Add an edge loop to the middle of the newly extruded segment. In Figure 1.26, you can see how edge flow automatically added the curvature.

Step 24: Base bottom arm. Grab the bottom face again. Hold down the Shift+right mouse button and choose Circularize Components. This will round the shape up to a perfect circle.



Subdivisions 1

□ Ignore Backfaces

□ Delete Faces

□ Extract Faces

0.5 0.5 0.5

Slice Along Plane:

LMB:	Click to drop or edit point
Ctrl+LMB:	Insert edge loop
Ctrl+MMB:	Insert centered edge loop
LMB:	Click or drag off mesh to slice
MMB:	Drag on mesh to quick slice
MMB:	Tweak last dropped point
Backspace / Del:	Undo last dropped point
Shift+LMB:	Click to snap point along edge midpoint
Shift+LMB:	Drag to snap point along edge
Ctrl+Shift+LMB:	Constrain to perpendicular angles
X:	Snap to grid
V:	Snap to points
Enter:	Commit cut

FIGURE 1.24 The Modeling Toolkit can provide faster ways to work with advanced modeling tools.

Unfortunately, it is tilted, but we can fix this by changing the Twist value to make it straight again. Extrude the face in the center down to create the length of the arm. Using the same technique used in Step 23, we can create a small rounded bottom for the arm. Finally, use the Multi Cut tool to fix the N-Gon (Figure 1.27).

Step 25: Create the arm bending socket edge. Select the edges across the bottom of the arm and press the R button to switch to Scale tool. This time,

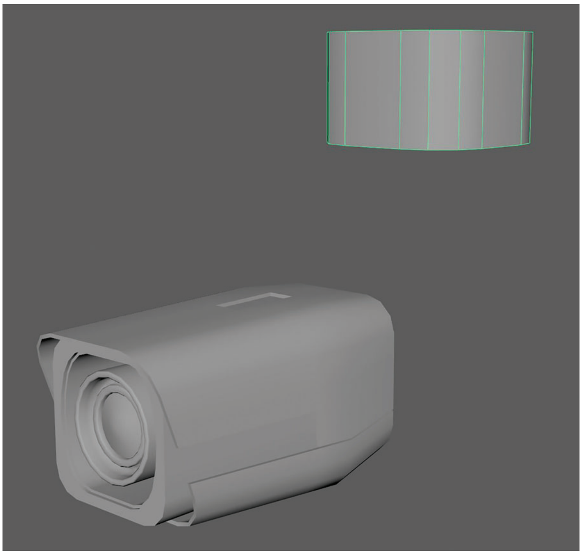


FIGURE 1.25 Using the same techniques covered above to create the base of the camera.

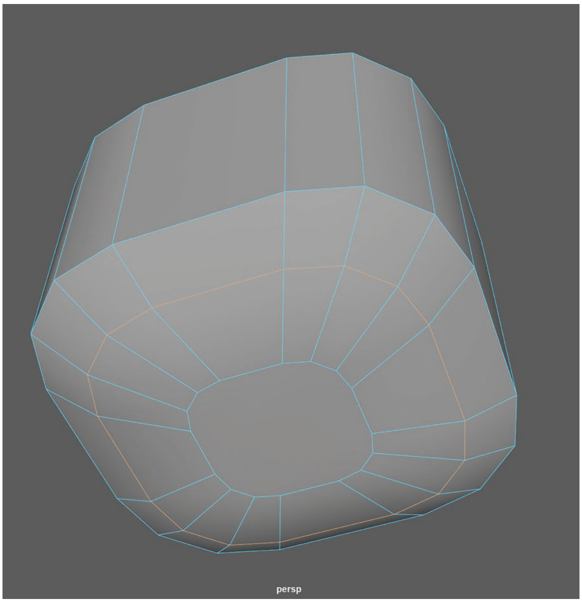
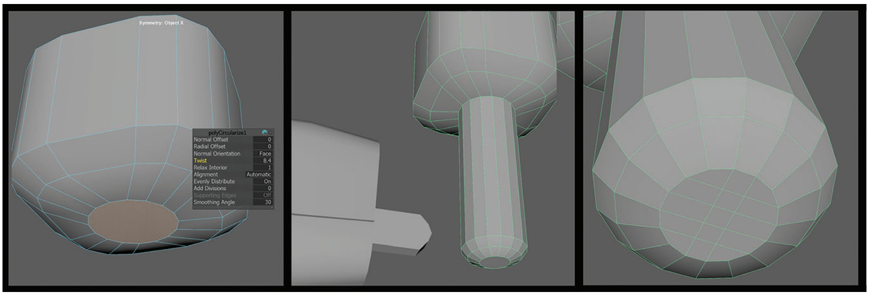


FIGURE 1.26 Adding new faces and edges to round off the bottom of the base.



polyCircularize1	
Normal Offset	0
Radial Offset	0
Normal Orientation	Face
Twist	8.4
Relax Interior	1
Alignment	Automatic
Evenly Distribute	On
Add Divisions	0
Supporting Edges	Off
Smoothing Angle	30

FIGURE 1.27 Creating the bottom arm and cleaning up the topology to eliminate N-Gons.

on the left side of the UI, look for the column of buttons we call the Toolbox. Try pressing Q, W, E, and R and you can see how to switch between these tools with the keyboard shortcuts. Double-click on the button that is highlighted when you press R to pull out the Scale tool settings. Check the Prevent Negative Scale option. Scale the lines on the X axis until they are flattened (they will not overshoot). Switch to Move tool and hold down the V button to turn on Vertex Snapping. While you are holding down the V button, drag the arrow of the move tool along the X axis (red-cone gizmo) and move your cursor to the point lying on the outer rim of the handle to snap the flattened line to that point only on the X axis. Do the same thing on the other side. Add another loop around the length of the handle to mark out the upper edge of the opening socket. What we are trying to achieve here is to mark the opening edge of the socket. The opening of the socket is highlighted in the last figure of Figure 1.28.

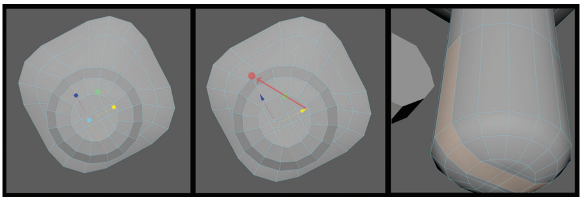


FIGURE 1.28 Building out the bottom of the arm using a few new tricks in each tool's options.

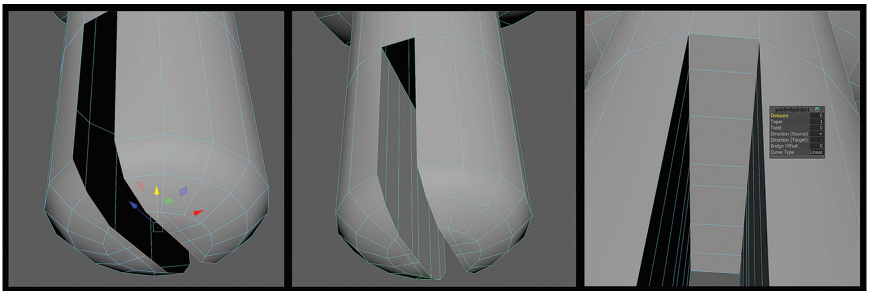
Tips and Tricks

Ctrl+Shift+right mouse button will also pull up the settings of the current tool. You can find the Prevent Negative Scale setting if you press R and then hold down the Ctrl+Shift+right mouse button. To quickly select part of a loop, select the beginning of the part of the loop and hold down Shift and double-click on the end of the part of the loop. This trick works on face loops, edge loops, and edge rings.

Step 26: Opening the socket. Delete the faces highlighted in the last figure of Figure 1.28. Grab the bottom edges and extrude them up. Scale them on the Y axis to flatten them. Switch to the Move tool and hold down V while dragging the edges up to snap the edges to the upper corner of the opening. Do not change the selection and hold down Shift+right mouse button and select the Bridge tool from the marking menu. This will bridge the two loops with faces. This command requires an equal number of polygons on the two loops (Figure 1.29).

Step 27: Merge vertices. Select the edge on the upper corner of the opening and move it just a little bit in any direction. Notice that there are two vertices overlapping instead of one merged vertex (Figure 1.30); this creates a tear in the mesh.

To fix it, we need to merge these vertices together. Press Ctrl+Z to undo the moving of the vertex. Then hold down Ctrl+Shift and drag over the two overlapping points to select both. Check to ensure that you are not



polyBridgeEdge1	
Divisions	0
Taper	1
Twist	0
Direction (Source)	+
Direction (Target)	-
Bridge Offset	0
Curve Type	Linear

FIGURE 1.29 Creating the notch of the arm by deleting faces, extruding edges, and bridging.

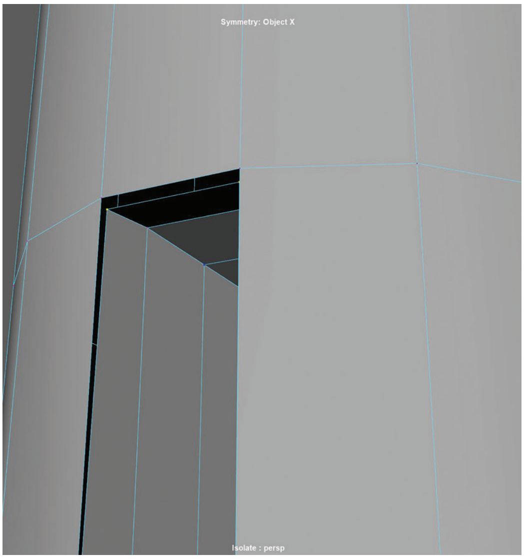


FIGURE 1.30 Our previous steps have created a form with holes in the mesh.

selecting anything else on the back of the form. You can choose Edit Mesh->Merge to merge these two vertices to one single vertex. Alternatively, you can hold down Shift+right mouse button and select Merge Vertices, but this time, the Marking Menu will show a nested sub-menu. We just keep dragging up to select Merge Vertices to Center.

Step 28: Shrink and attach the camera to the base. Grab the faces of the back arm of the camera body and scale it up or down to make the size fit with the opening of the base. Move the base to attach the arm with the socket (Figure 1.31).

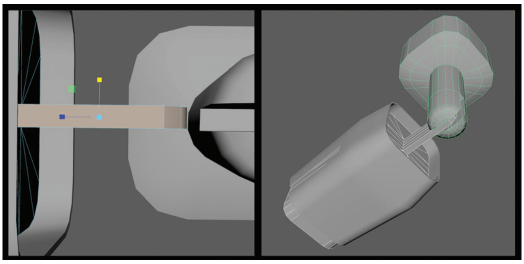


FIGURE 1.31 Scaling the faces of the back of the camera to fit the mounting base.

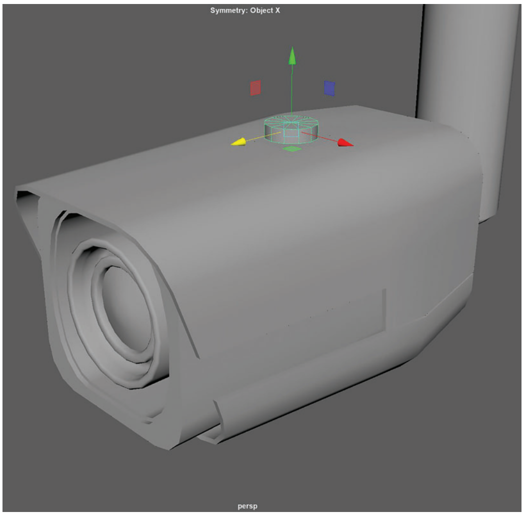


FIGURE 1.32 Roughing out the switch at the top using a cylinder.

Step 29: Top switch. Create a cylinder (Create->Polygon Primitives->Cylinder). Move and scale it to the opening of the top shell of the camera. In the input section of the Channel box, click on the polyCylinder1 and change the Subdivision Axis to 12 (Figure 1.32).

Why?

We made the Subdivision Axis smaller to lower the polycount of the little top switch. It is such a small part that we do not need the same number of loops as the lens. In games, polycount is important, and trimming away those we don't need as we work will generate cumulative benefits in the long run.

Step 30: Reduce polycount. It is always possible to reduce polycount of a model to save a little bit of performance. Go to Display->Heads Up Display and check on Poly Count. You can see on the upper left corner of the viewport that we have 1736 Tris in total. There are two ways we can reduce polycount:

1. Delete edge loops that seems unnecessary. Grab the outer shell of

the camera, select the edge loop in the middle, and hold down Shift+right mouse button and chose Delete Edge. Notice that there is no difference in the form after deleting it. Similar cleanups are shown in Figure 1.33.

2. If a loop cannot be completely deleted, triangulate parts of the loop. We clearly need no extra edge loop for the top flat surface of the outer shell, but we have two for the purpose of opening a hole on the top. To fix this, go to Object mode and hold down Shift+right mouse button and select Target Weld Tool. Click and drag the vertex in the middle area of the upper edge of the shell to the point next to it to weld it to that vertex. Using this technique, we can weld a lot of points without affecting the shape of the model. We may end up with some triangles, but it is totally fine for most non-deforming

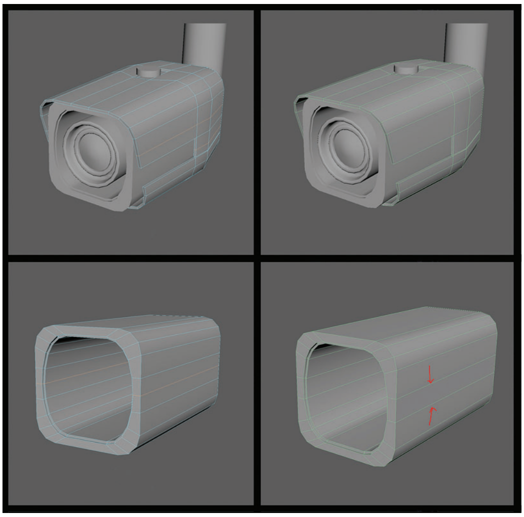


FIGURE 1.33 Strategically deleting unnecessary edges reduces our polycount without sacrificing form.

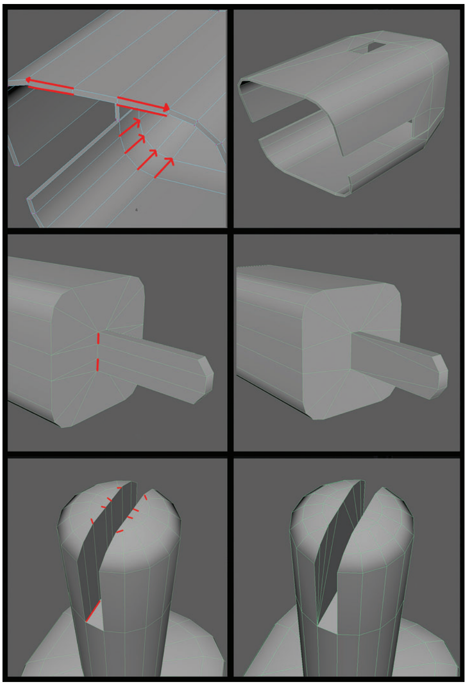


FIGURE 1.34 Reducing polycounts with careful vertex welding.

(not bending) forms, especially for a game model. Similar reducing results are shown in Figure 1.34.

Keep in mind that you still want to avoid too many triangles; triangles are harder to manage for the UV process (an important part of the texture process that we'll cover later), and they make it harder to do high-resolution sculpting (if you need to do so for complex forms including organic shapes). You can always consult the supervisors of your team to get their suggestions on the polycount if you do work for a studio. The final polycount after these optimizations is 1494 tris for our camera.

Tips and Tricks

The Delete Edge command should be what you use all the time to get rid of edges. The delete button on the keyboard does delete the edges but not delete the vertices on the edge, which means that when the renderer triangulates, it still produces the extra triangles from those left-over points.

Step 31: Clean up. On the top of the base, there is a big, flat N-Gon. Grab that top face and extrude it in. Hold down the Shift+right mouse button and select Merge Faces To Center. Select all models we created, press Alt+Shift+D to delete all the history. Finally, do a Modify->Freeze Transformation to clean up the transform.

Why?

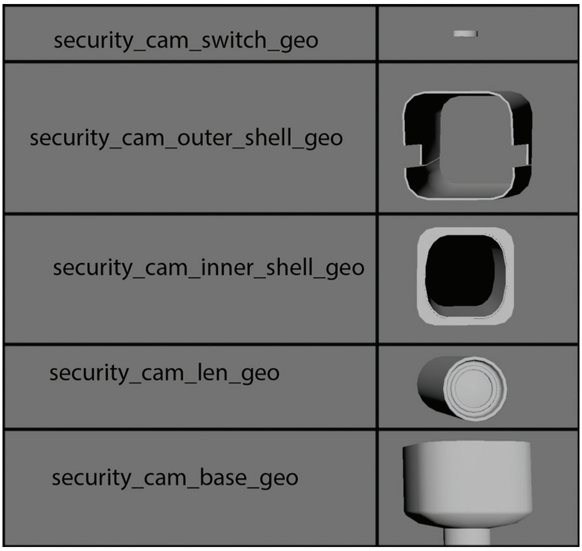
You may wonder what this Freezes Transformation does. Well, in Maya, a model has two primary components: transform and shape. Transform governs where the model is, how it is tilted, and how it is scaled; these are reflected in the translate, rotate, and scale values in the Channel Box. Shape governs the vertices, faces, and edges and how they are combined together to form the shape of the model; the final appearance of the model is the shape of the model moved, rotated, and scaled by the transform of the model. If you do recall, we have scaled the outer shell of the camera, and that scale value will appear in the Channel Box; freeze transform will clean that up and bake the scale we did to the transform of the model to the shape of the model. Many processes later (Rigging, UV Mapping) do require the transform of the model to be baked to the shape of the model through Freeze Transformation so that the final look of the model is the actual shape instead of a shape getting scaled, rotated, and moved by a transform.

Step 32: Naming and organization. On the left side of the UI, there is a pallet with a tab called Outliner; this is a list of the currently existing objects in the scene. Select anything in the *viewport and you can also see it highlighted in the list of the Outliner. Alternatively, you can also select an object by clicking its name in the outliner. In the Outliner, you can hold down Shift to select multiple objects or hold down Ctrl to deselect.*

Tips and Tricks

If you cannot see the Outliner, go to the column of buttons where we have the Move, Rotate, and Scale tools; the last button in that column is the toggle to show or hide the Outliner.

Select everything in the viewport in Object mode and press Ctrl+G to place them into a group. Something called group1 will appear in the outliner. This is a group (really a parent object). You can press the plus sign in front of it to open the group and see the children models inside of it. Double-click to rename any object there. For now, rename the group to security_cam_geo_grp. Spend some time renaming all other objects; the final naming is shown in Figure 1.35.



security_cam_switch_geo

security_cam_outer_shell_geo

security_cam_inner_shell_geo

security_cam_len_geo

security_cam_base_geo

FIGURE 1.35 Names to use in naming the shapes build so far.

Tips and Tricks

Anything inside of a group will follow the group. You can now grab the group in the outliner and move the whole collection of shapes. You can put any object or objects under a group by grabbing the objects first and selecting the group last and pressing the P button on the keyboard. This is also something we call parenting. If you select something inside of a group and press Shift+P, this will get the object or objects out of the group (called unparenting). You can also parent one object to another object instead of a group. In the outliner, you can click the middle mouse button to drag anything around or drag one object to another to parent one object under another.

Step 33: Clean up the outliner. We do not need any other objects in the scene; there could be other empty groups in the outliner due to some operations we did to the model. We can grab anything outside of security_cam_geo_grp and delete them. Alternatively, we can go to File->Optimize Scene Size and let Maya clean these up for us.

Step 34: Save the file. Go to File->Save Scene, in the pop-up Save window, change the File name to game_set_models, navigate to a folder that is safe and easy to find, and press the Save button to save it.

Other Useful Commands

We have introduced some of the most important commands for modeling. Let’s start a new scene and go over a few more before we do some assignments.

Grow and Shrink Selection

Create a sphere (Create->Polygon Primitives->Sphere) and select the top vertex. Hold down Ctrl+right mouse button and choose To Faces->To Faces to select the top faces. Hold down Ctrl+right mouse button again and chose Grow Selection->Grow to select all the direct neighbor faces. Press the G button three times to redo Grow Selection three more times. You can also find Shrink Selection in the Ctrl+right mouse button marking menu.

Extract Faces

With the top four rows of faces selected, hold down Shift+right mouse button, and choose Extract Faces. Drag the blue arrow to shift the face away; you can now see how Maya separated the model into two objects. Notice that in the outliner, you can see the pSphere1 becomes a group, and there are two objects inside of it. That transform1 is the remaining construction history that you can use Delete History to get rid of.

Combine and Separate

Some commands, like Bridge, can only be used for component on the same object. So to bridge the upper shell with the lower shell (Figure 1.36), you have

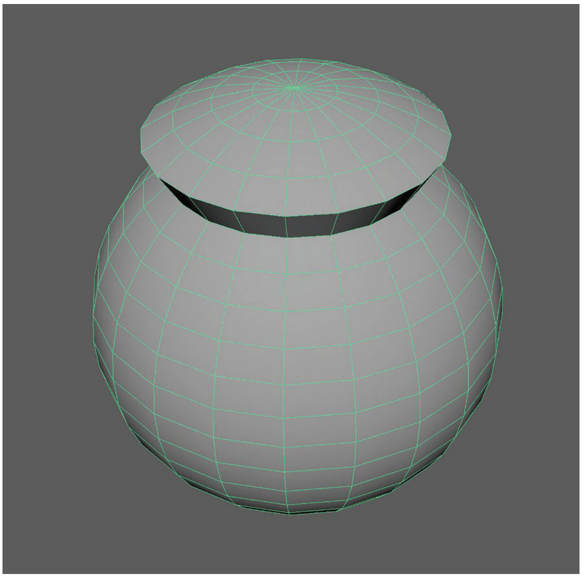


FIGURE 1.36 In order to bridge collections of polygons like this, you first must ensure that the polygons are parts of the same single object.

to combine the models together into one object. To combine models, grab all models you want to combine and use Mesh->Combine. By the way, you can also see the Separate command right below Combine. Separate will separate the model into multiple ones based on their connectivity.

Create Cables or Pipes

From time to time, we may want to create a cable or a pipe. Go to Create->Curve Tools->CV Curve Tool, and click and drag in the viewport to drop down a CV point. Click and drag again to add a new one; keep doing this and you will see a curve getting created. You can hit backspace to roll back and drag the middle mouse button to refine a placed CV point. When you are happy with the shape, hit Enter to finish the creation (Figure 1.37).

CV stands for Control Vertices. Maya will interpolate between the vertices to form a curve. This type of model is called NURBS, which uses mathematical interpolations between control vertices to create a form. These are fundamentally different from the camera model (polygon) we created earlier.

After creating the curve, you can still edit it by holding the right mouse button on it and choosing Control Vertex. Then you can move the CV to refine the shape as desired. The curve will be created on the grid by default. You can go to the front, top, or side view to create your curve

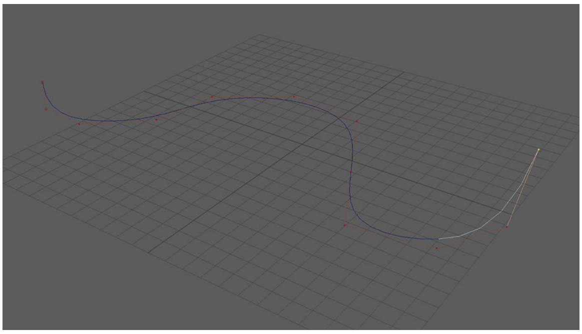
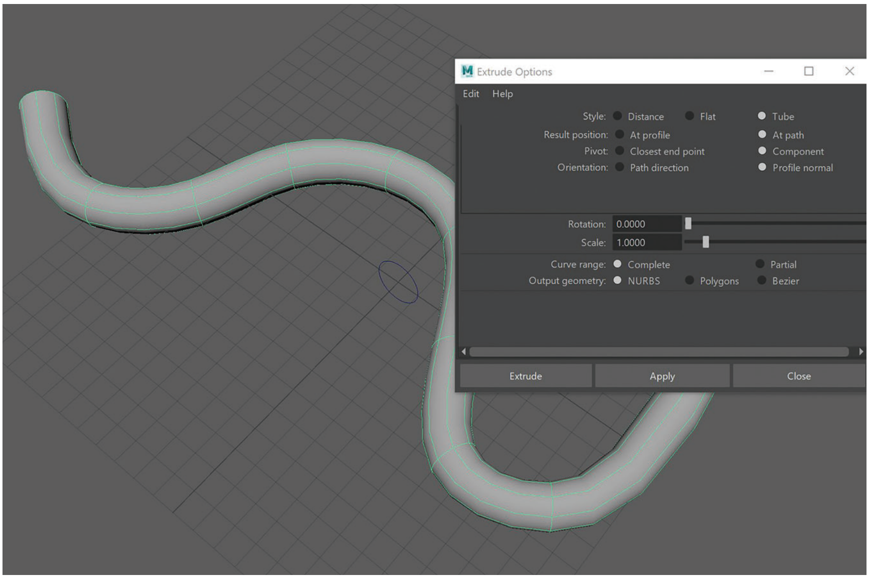


FIGURE 1.37 Using the CV Curve Tool to create a series of CVs that define a curve.

so that the curve snaps to the grid of that view. Go to Create->NURBS Primitives->Circle to create a circle. Grab the circle and the curve created previously, and go to Surfaces->Extrude□ (be sure to click the square to pull up the Extrude Options window). There, change the Result position setting to “At path” and change the Pivot setting to “Component”. Press the Extrude button, and you will see that a tube is created (Figure 1.38). This Extrude is not the same Extrude we did with polygons; it basically places the circle along the curve to create a frame and then interpolate a shape out of it.

You can scale the nurbsCricle1 to change the radius of the tube, and you can still tweak the shape of curve1 to change the shape of the tube (Figure 1.39).

However, this tube is not a polygon or polygon-based (which we will need for games). So to convert it into a polygon-based form, go to Modify->Convert->NURBS to Polygons□. Change the tessellation method to “Control points”. Press the Tessellate button to convert the tube to a polygon (Figure 1.40).



Style: ○ Distance ○ Flat ● Tube
Result position: ○ At profile ● At path
Pivot: ○ Closest end point ● Component
Orientation: ○ Path direction ● Profile normal
Rotation: 0.0000
Scale: 1.0000
Curve range: ● Complete ○ Partial
Output geometry: ● NURBS ○ Polygons ○ Bezier

FIGURE 1.38 Creating a tube using NURBS extrude.

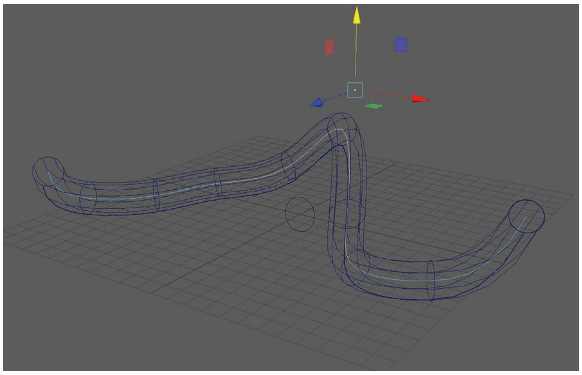
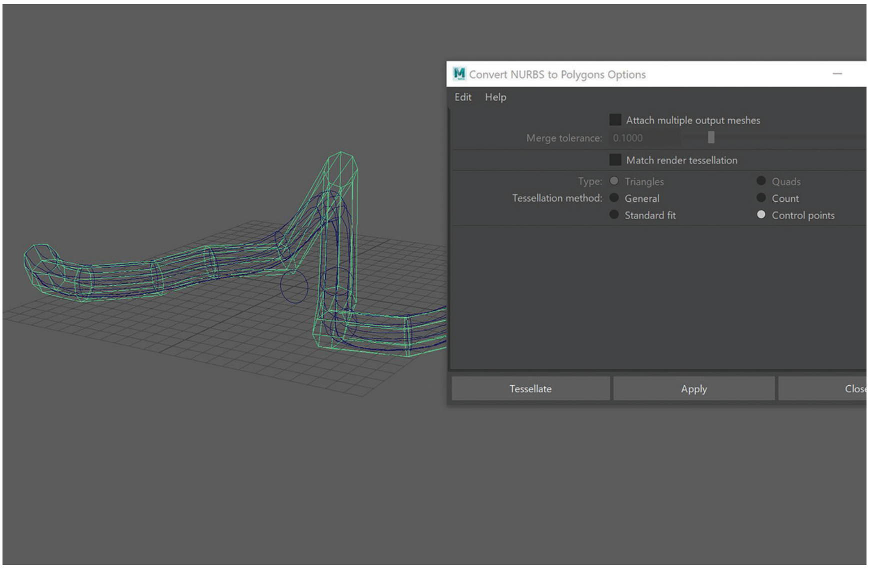


FIGURE 1.39 Once a NURBS form is created, the form can be adjusted by editing the curves used to create it.



☐ Attach multiple output meshes

Merge tolerance: 0.1000

☐ Match render tessellation

Type: ○ Triangles ○ Quads

Tessellation method: ○ General ○ Count ○ Standard fit ● Control points

FIGURE 1.40 Tessellated NURBS form that is now a polygonal object.

Until now, you can still tweak the curves to change the radius and the shape of the curve. When you are happy with the form, select the polygonal shape and delete the history and delete all curves and the original NURBS surface as they're no longer needed.

Tips and Tricks

If, at any time, the tube model appears black, you can rotate the circle on the X axis until it flips back to normal. NURBS curves and surfaces are a different type of model that are mathematically interpolated between the control points we created. They are primarily used for architectural or industrial design. Most times, we don't use this type of model in game scenarios. However, they can be very useful to construct a form originally (that we then convert into polygons).

Extrude Along a Curve

Another variant of creating a tube is to create a curve in front of a face and then extrude that face along the curve. To do this, select both the face and the curve (Figure 1.41), and press Ctrl+E.

In the pop-up dialog boxes, increase the number of the Divisions setting to create a smooth extrusion along the curve (Figure 1.42). If the extrusion is backwards, you can grab the curve, do a Curves->Reverse Direction to fix it.

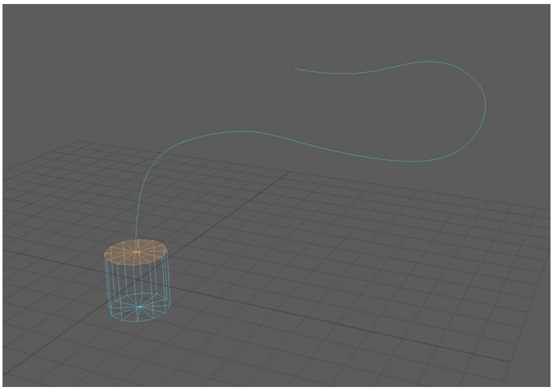
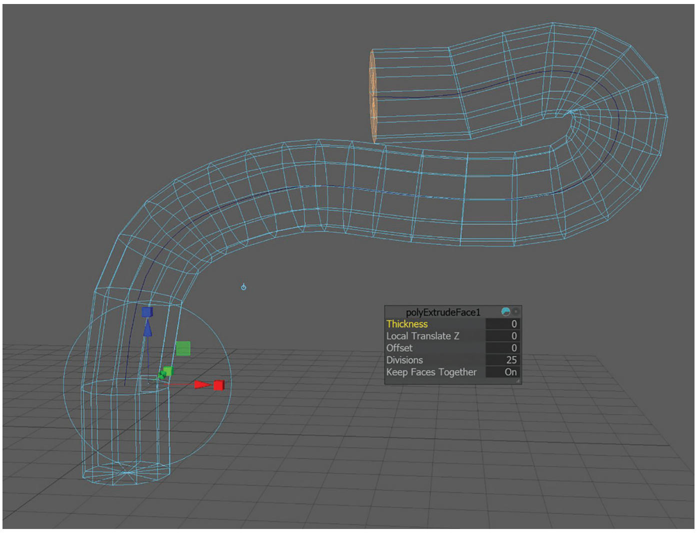


FIGURE 1.41 Extruding along a curve.



polyExtrudeFace1

Thickness	0
Local Translate Z	0
Offset	0
Divisions	25
Keep Faces Together	On

FIGURE 1.42 Tweaking the Extrude along curve options to get the resolution you desire.

Duplicate, Duplicate with Transform

You can grab any model and press Ctrl+D to duplicate it. The duplicated model will be at the same location as the original (although you'll see the name of the new form in the Outliner). Right after duplicating, you can use the Move tool to move the new duplicate away (Figure 1.43).

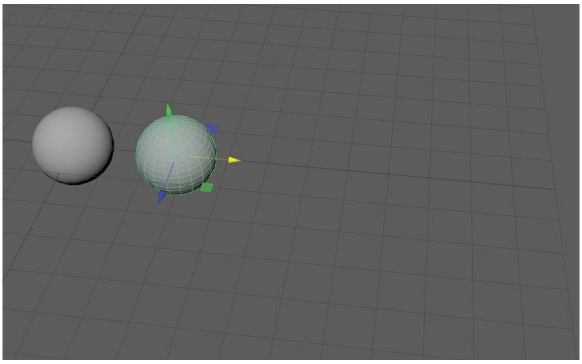


FIGURE 1.43 Duplicate allows for a quick copy of an original.

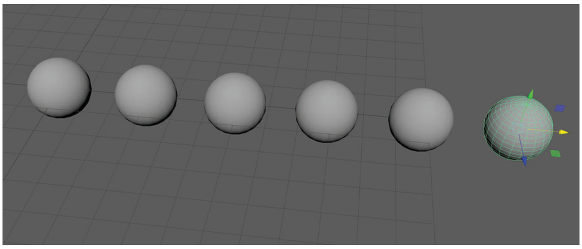


FIGURE 1.44 Duplicate with Transform (Shift+D) duplicates and transforms (moves) the object in the same command.

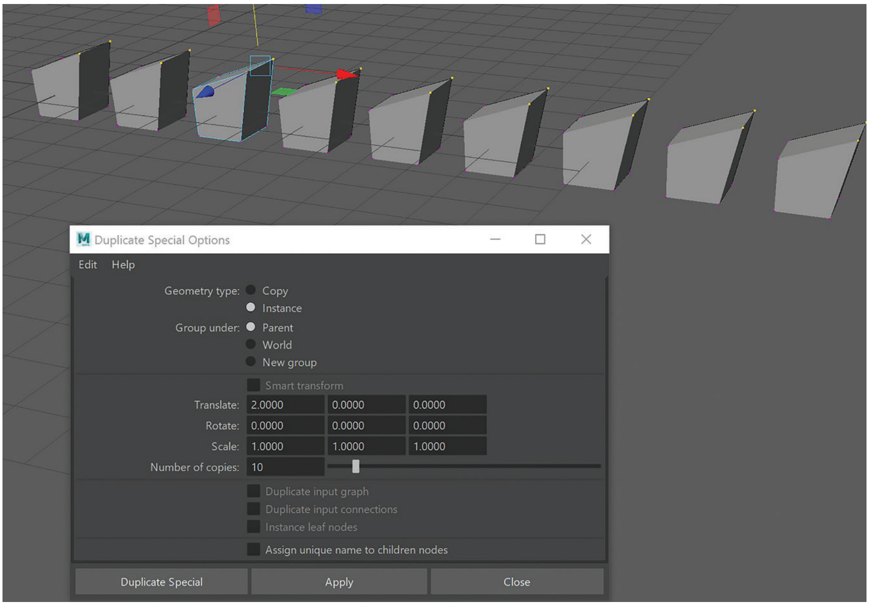
If you want to create another duplicate and have it move the same distance (or rotate the same amount), you can press Shift+D. You can keep pressing Shift+D to have multiple duplications, each offset the same amount as last time (Figure 1.44).

Duplicate Special

From time to time, we may want to create multiple duplications but as *instances*. An instance is a copy that keeps the link of the shapes between the original and the copy; we can adjust any one of the duplications to update the shape of all others (but not transform). Grab your model and go to Edit->Duplicate Special□. Change the Geometry type to "Instance". Change the first number of the Translate to 2 (the X axis), and the Number of copies to "10". Finally, press the Duplicate Special button, and you will see ten duplications of your model, each two units away from each other, and more importantly, editing any one of them will affect all others (Figure 1.45).

Mirror

Anytime you forget to have symmetry on and want to make the model symmetrical again, you can select the model in Object mode and do a Mesh->Mirror. Try different axes and directions to make sure you got the correct side mirrored. The merge threshold should be as low as 0.001 if you wish to only have the vertex in the center mirrored. You can also change the Border setting



Geometry type: ○ Copy
● Instance

Group under: ● Parent
○ World
○ New group

□ Smart transform

Translate:	2.0000	0.0000	0.0000
Rotate:	0.0000	0.0000	0.0000
Scale:	1.0000	1.0000	1.0000
Number of copies:	10		

□ Duplicate input graph
□ Duplicate input connections
□ Instance leaf nodes
□ Assign unique name to children nodes

FIGURE 1.45 Using instances (as opposed to copies) to make copies of an object that will change when the original is manipulated.

to bridge or do not merge the geometry along the axis of symmetry to have a different result.

Center Pivot

You can grab any model in Object mode and do a Modify->Center Pivot; this will move the pivot of the model to the center of its bounding box. The Pivot is the location where the object is rotating around. It is needed whenever you want to be able to rotate or scale a model from its geometrical center.

Change Pivot

In any mode, you can hold down the D button on the keyboard and drag the gizmo to adjust the location or orientation of your pivot; you can also click on any elements on the model to snap your pivot to that element.

A good example that we want to do this is to change the pivot of the body of the camera to the hinge of the arm so that we can rotate it around the hinge.

Snapping

When moving an object or its various elements, you can hold down the X button to snap to the grid and V button to snap to vertices. The snapping toggles are on the Status line, which is the row of buttons under the main menu. The snap toggles are the six buttons with a magnet in their icon. Try these toggles and see what they do.

Hide Model

You can grab any model or its other elements and press Ctrl+H to hide them. To unhide, press Ctrl+Shift+H; this will only unhide the object you are selecting (probably in the Outliner) if you have something selected. It will unhide everything if you have nothing selected. After making a model, name it properly, freeze transformation, and hide it so you can move on to the next one without the other models blocking your view.

View Control

At any time, if your cursor is in the Viewport, you can press the spacebar to go to the Four View layout, and this will show you the Top, Perspective, Front, and Side views; you can then move the cursor to any view, and press the spacebar to maximize that view. However, we recommend to just hold down space and drag up, down, left, and right to go to these views.

Assignments

We have covered enough commands that you are now able to create models of your own; go ahead and start modeling some of your own models in Maya,

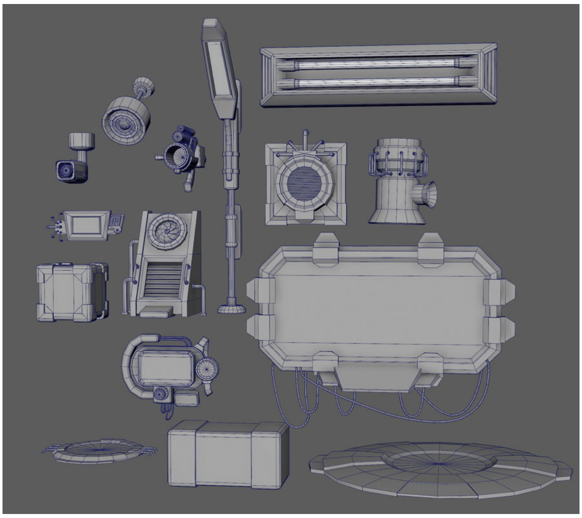


FIGURE 1.46 A selection of props that can be built using the techniques covered in this chapter.

and make sure you find references and get the correct measurement. Figure 1.46 shows a few examples of what we are looking for.

Geometry Errors

Sometimes in the modeling process, some errors can emerge. These errors might not even be readily visible in your model, but without fixing them, you can run into some serious problems later in a game engine. While these can be pretty technical, and the hard-core specifics are a bit outside the scope of this book, it's worthwhile to talk about them for a minute and – more importantly – evaluate how to fix them. Here are some typical geometry errors that we can now check on your model:

Non-Manifold Geometry. *This geometry cannot be unfolded and flattened to a 2D surface. Typically, there is an edge shared by more than two faces or inconsistent normal directions. This type of model will confuse the renderer on which side is the outside of the geometry.*

Lamina Faces. *Two faces that share all of their edges. Typically, this is caused by duplicating and combining meshes that have the same faces.*
Zero Length Edge. *A self-descriptive situation in which an edge has no length.*
N-Gon. *We have covered N-Gon already; any face with more than four sides is a N-Gon.*

Luckily, even though the theory behind these errors is abstract, fixing them is usually pretty easy. To clean up the models, go to Mesh->Cleanup□. Under the Fix by Tessellation section, check on "Faces with more than 4 sides". Under Remove Geometry, check on "Lamina faces and Nonmanifold geometry". Press the Cleanup button; this will, in theory, clean up all the errors. Maya may choose to delete some of the faces because they are error geometries; make sure you check around the model and recreate any missing models.

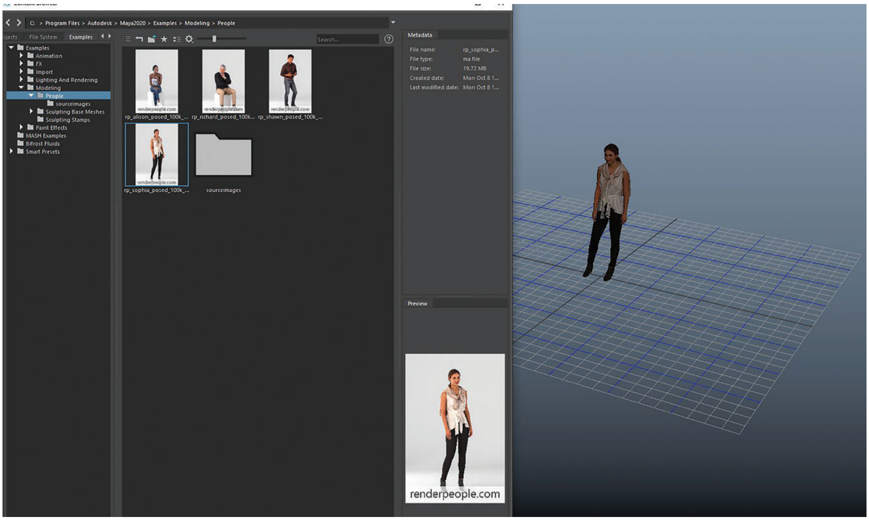
Tutorial 1.2: Modular Set Pieces

Making a compelling and complex environment is a daunting task. To ease the pain, we are going to adapt to a modular workflow. This means we will make reusable pieces that are easy to combine with each other, like a system like Lego building blocks. The props we have made previously as assignments are already designed for that purpose, but to make the foundation of our game level, we need a more unified system. This means we need to have a chart of sizes that our models will have to exactly match, so they can be assembled seamlessly.

There are two size systems we can use: decimal and binary.

For decimal, we will have sizes like 10, 20, 30, 50, 100…
For binary, we will have sizes like 16, 32, 64, 128, 256, 512… (both in cm)

Both systems are popular, and we are going to follow the binary system. The author has found that it is easier to combine modules seamlessly and easier to match with textures sizes, which is also binary.



File name: rp_sophia_p...
File type: ma file
File size: 19.72 MB
Created date: Mon Oct 8 1...
Last modified date: Mon Oct 8 1...

FIGURE 1.47 Setting up our grid to build modularly. The person acts as a size and scale reference.

Grid

Go to Display->Grid□. Set the Length and width to 256, and set the Grid lines to 64 and Subdivisions to 4. Drag the slider of Grid lines and numbers to make it a blue color and press Apply and Close. This will create a grid that has its edge 256 cm away from the center and a blue grid line every 64 cm with four extra divisions in-between every blue grid line, which makes every grid 16 cm long.

To verify our sizes, go to Windows->General Editors->Content Brower. Under the examples category on the left side of the window, choose Modeling->People. Drag a standing character to the viewport to import the human model; the height of the model should be slightly shorter than half of the grid length (Figure 1.47). If your character appears gray, hit 6 on your keyboard to have Maya show the materials as well.

Create a Base Floor

Step 1: Base floor dimension. Let’s hide our other models, so we can start our new model with nothing else visible. Create a cube, set its

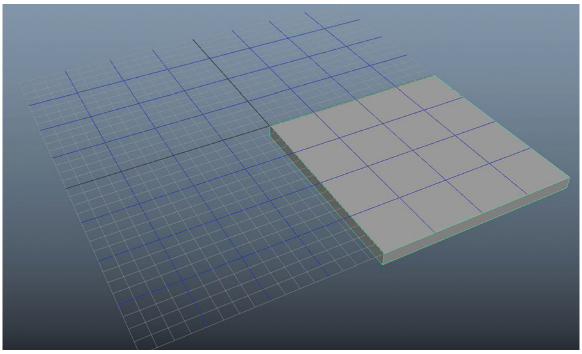


FIGURE 1.48 Creating and snapping the first-floor module to our grid.

Translate Y to –8, set its Scale X and Scale Z to 256, and set its Scale Y to 16. Switch to the Move tool and hold down both D and V buttons, and drag the pivot of the box to the upper corner on the negative X and Z quadrant. Release all the buttons. Now, hold down X button and drag the center of the gizmo to snap the model to the positive X and Z quadrant (Figure 1.48).

Why?

We want the model to be easily snapped together. Positioning the pivot to the corner of the box is extremely helpful for the snapping. We also want the pivot to be at the center of the world to avoid any offset.

Step 2: Floor edge trim. To help in adding trims to the side of the floor, grab the top face, and press Ctrl+E. Set the Offset setting to 16. This will give us a rim on the outside of the floor (Figure 1.49). Name this model floor_01 and hide it.

We are now done with this module. Every time we finish a module model, we can name it, hide it, and move on to the next one. This way, all of our models are created in one Maya file for easy access. This also allows us to maintain the scale of our game. We are not planning on making a whole lot of models for our environment, so keeping them all in one scene file is manageable. But if you'd rather, you are more than welcome to create new files for extra models instead. But be sure you

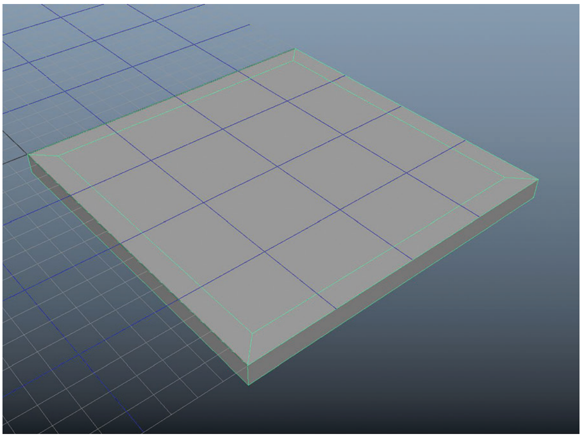


FIGURE 1.49 Creating the base trim of the module using Extrude.

maintain consistent Grid settings across the various scenes if you do so.

Step 3: Base wall dimension. Create a cube and set its Scale X to 256, Scale Y to 512, and Scale Z to 32. Snap its pivot to the lower back corner and then move it to the center of the grid (Figure 1.50).

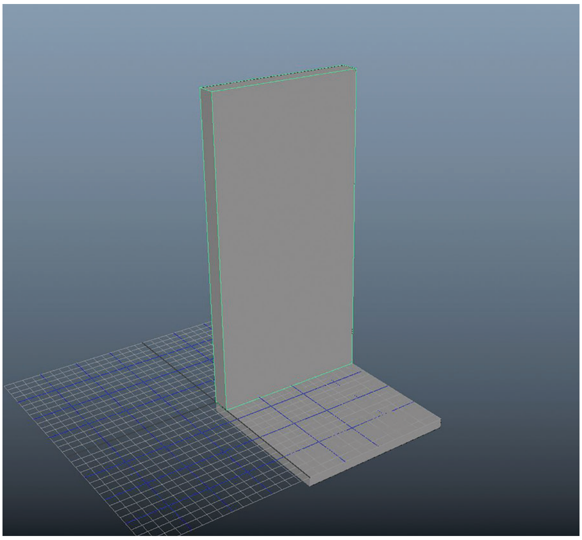


FIGURE 1.50 Creating a base wall module.

Step 4: Add bottom trim. Add an edge loop toward the bottom of the wall module. Extrude out the bottom face and move the top edge of the extruded face down to create a bottom trim (Figure 1.51); name this model wall_01.

Step 5: Arch wall. Follow the steps of Figure 1.52. Create a pipe (Create->Polygon Primitive->Pipe). Under the Input section of the Channel Box, click on polyPipe1 and set the Radius to 128, Height to 512, and Thickness to 32. Set the Rotate Z of the model to –90. Delete the frontal and bottom

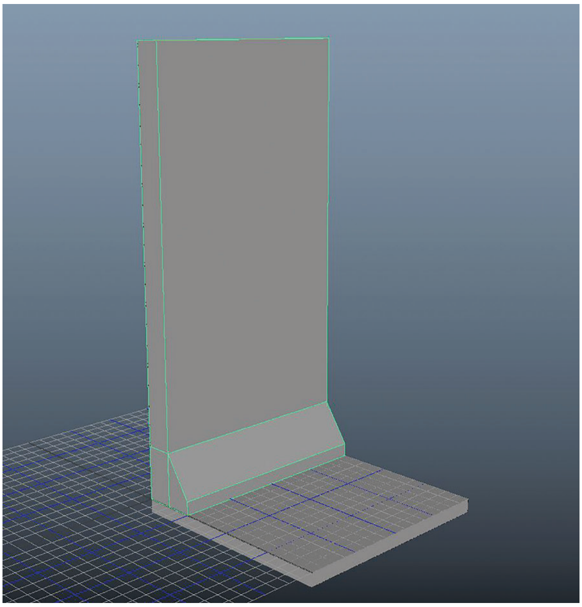


FIGURE 1.51 Creating trim for our wall_01.

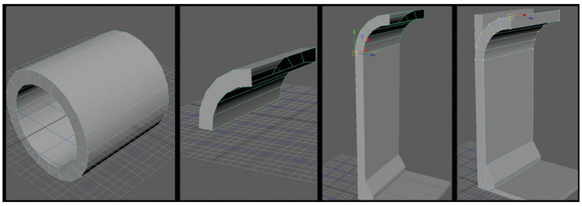


FIGURE 1.52 Creating the arched top of the wall by combining a pipe segment.

quarters of the pipe. Next, hold down D and V, and snap the pivot of the pipe to its back side corner. Hold down V and snap the pipe to the top of the wall we created in the previous two steps. Duplicate the wall and delete its top face. Combine it with the pipe by selecting both the wall and pipe and choosing Mesh->Combine. Grab the vertices of the pipe and the top of the wall and while holding down X drag them down until the top of the pipe is the same height as the height of the original wall. You can go to the side view to check out the alignment. Grab all the vertices and hold down Shift+right mouse button, go up and up again (or choose Edit Mesh->Merge). This will merge the vertices between the top of the wall and the bottom of the pipe. Double-click on one of the edges of the hole in front of the pipe and hold down Shift+right mouse button. Choose "Fill hole". Name this model wall_02.

Tips and Tricks

Step 5 has many steps, but the idea is simple. We want an arch on the top of the wall. Whenever we need something complex, we can break it up to smaller primitives. When we create these primitives, we can snap them together, combine them, and merge the vertices.

Step 6: Wall frame. Copy the arch wall we created and move its pivot to the origin. Change its Scale X to 0.25. Grab the front faces and press Ctrl+E. Change the Local Translate Z of the extrude to 16. Extrude the same amount again, but this time, scale the faces in on the X axis to create a little taper. Use scale or snapping to flatten the top front faces (Figure 1.53).

Step 7: Wall frame detail. Grab the faces in the front middle part of the model and hold down Shift+right mouse button and chose Duplicate Faces. Set the Local Translate Z to 16. Grab the bottom vertices and drag them up. Bridge the bottom edges and bevel the primary turning edges. This will give us extra volume; you can create additional ones to make the model more complex (Figure 1.54). Name this model wall_frame_01.

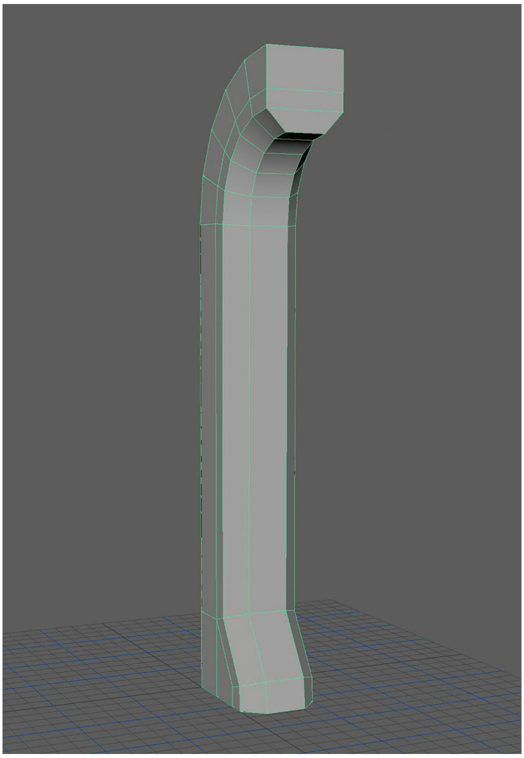


FIGURE 1.53 Creating a tapered wall section.

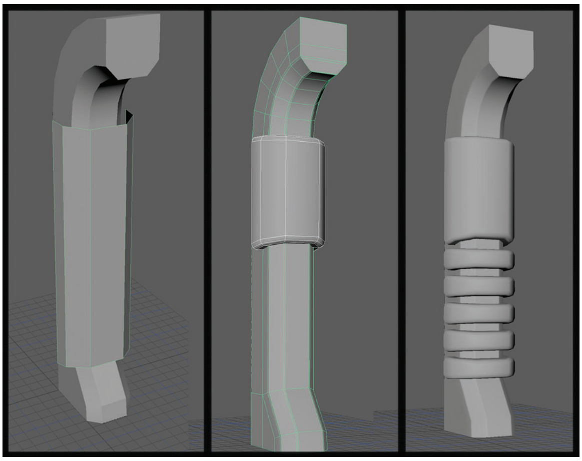


FIGURE 1.54 Extra detail on the walls.

Tips and Tricks

Always name and clean up your models when you have finished them. Your future self will thank you for making everything clean and tidy.

Step 8: Wall corner. We can create rounded corners for our rounded walls for when walls meet as we assemble them. Duplicate and snap our modules like the first figure in Figure 1.55. You can hold down the J button while rotating to snap your rotation for every 5 degrees. It is important that the modules are snapped to each other exactly. Have one blue (64 units) grid gap between the two hallways or corridors and the turning portion of the floor. This is to ensure that there is space for the rounded transition part.

Moving on to the second figure of Figure 1.55, select these two walls of the turning point, duplicate (Ctrl+D) them, and combine (Mesh->Combine) them. Grab the two columns of the faces that will connect to the turning portion. Hold down Shift+right mouse button and chose Bridge Faces. The result may look messy, so change the division to 7, and Curve type to Blend. The resulting middle part is going to be our turning module; delete the extra ones on the side and bridge the holes on the side to finish it. The outer corner is done the same way.

Step 9: Floor variations. Create a few varying sizes for the floor, like the gap we need to fill for

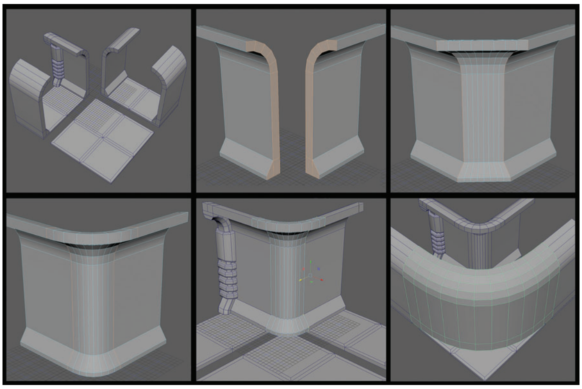


FIGURE 1.55 Creating a transition part for an outer corner.

the turning of the corridor we did for Step 4 (Figure 1.56). The sizes we choose to use are: 256×256×32, 256×128×32, 256×64×32.

Step 10: Stair frame. Create a cube and set the Scale X, Y and Z to 256. Snap its pivot to the back lower left corner and snap the cube to sit at the positive quadrant. Move (and snap) its bottom and top row of vertices to make a tilted frame for the stairs. Its bounding box length is 6×64 units (six blue grids). The footing of the shaft is 64 units, and the thickness of the shaft is 16 units (one gray grid). The results look like Figure 1.57.

Step 11: Stairs. Duplicate the stair frame to create the other side. Snap the duplication so that their whole width together is 256 units. Create a box, make its Scale X 32, Scale Y 8, and Scale Z 200. Move it to the first stairs location. It should be around 18 units high. Bevel all the edges of the box, and extrude from the two side faces to make the connection to the frame. Bevel the bottom edge of the frame to add a little detail. Fix the N-Gon after the bevel. Finally, bevel the edges of the frame (Figure 1.58).

Step 12: Stair handrail. Go to Create->Curve Tools->CV Curve Tool; we have covered this tool

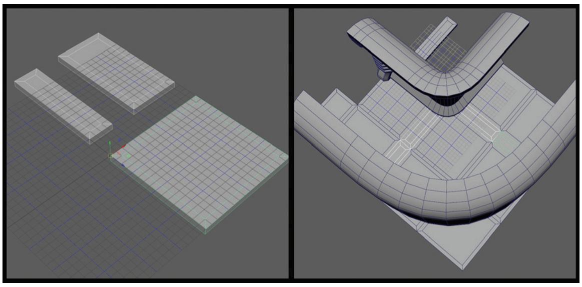


FIGURE 1.56 Creating other modular parts for floors.

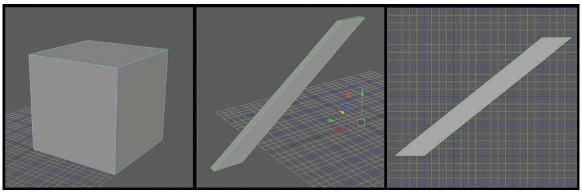


FIGURE 1.57 Building the stairs. Note that for modularity to work, the exact positions of the snapped vertices are important.

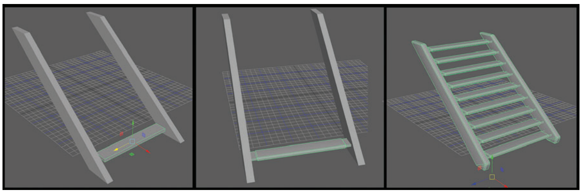


FIGURE 1.58 Creating the stair steps.

previously in the part about extruding along a curve. Use the Curve tool to create the profile of the handrail. Take care to make sure you have enough points on the arcing part; the amount of points you place will determine how many segments you will have on the final polygon shape. Use Extrude along curve techniques to create the handrail. Addition columns can be created using cylinders (Figure 1.59). Remember: be sure to covert the NURBS form into polygons.

Step 13: Other modular pieces. Other modular pieces are made with the same techniques covered previously; here is a list of all the pieces modeled:

- **Walls** – *There are three walls, five wall frames, and some random small blocks. The size of the tall ones is 256×512 with a thickness of 32 (Figure 1.60).*
- **Arcs** – *These arcs are having a radius of 256 units and a thickness of 32; an outside arc,*

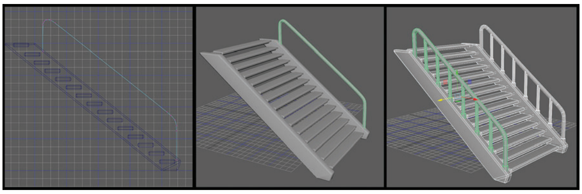


FIGURE 1.59 Creating the handrail using NURBS techniques.

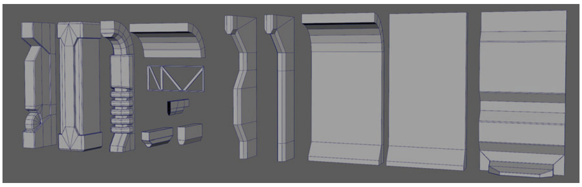


FIGURE 1.60 Completed wall modules.

a wall, and a wall bottom trim are also built (Figure 1.61).

Floor – *Floors with 256×256×32, 256×128×32, 256×64×32 are built to create hallways with different sizes. Two grid modules are also built (Figure 1.62).*

Pipes – *Pipes came with three sizes, each with a radius of 16, 8, and 4. Be sure to build some turning structures to support complicated combination (Figure 1.63).*

Stairs – *We have two stairs, a higher one with a 256 units elevation and a lower one with a 64 units elevation. Handrails were also built to support variations (Figure 1.64).*

Windows – *Window came in four sizes: 256×128×32, 128×128×32, 512×512×256, 96×64×160 (Figure 1.65).*

There are 57 modular pieces. It is hard to determine how many are needed, so it is wise to build less and try creating a hallway or a room and see if more modules are needed.

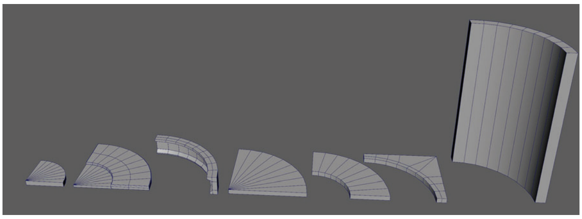


FIGURE 1.61 Completed arc modules.

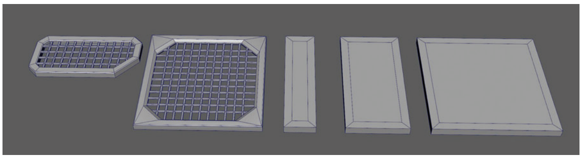


FIGURE 1.62 Completed floor modules.

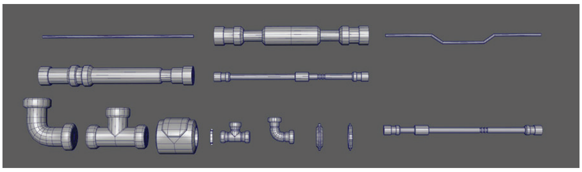


FIGURE 1.63 Collection of completed pipe modules.

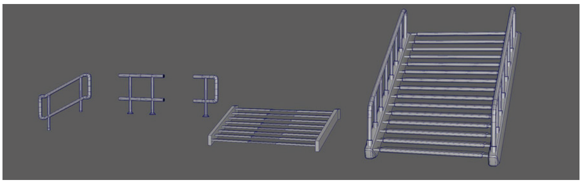


FIGURE 1.64 Stair modules.

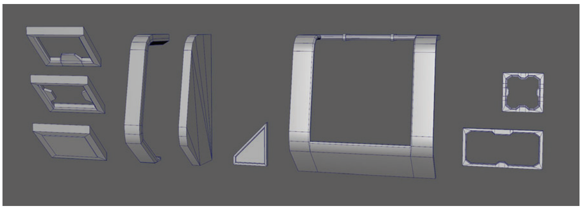


FIGURE 1.65 A variety of finished window modules.

Step 14: Hero assets. Hero assets are the assets that we only use a few times and so might need a bit of extra care and detail. We will create two hero assets for the final scene; the creation process of these hero assets is tedious, expect to spend a lot of time on them and have a higher polycount on these assets. But even though the fidelity might be higher for these assets, the tools and commands used to create them are no more than what we have covered (Figure 1.66).

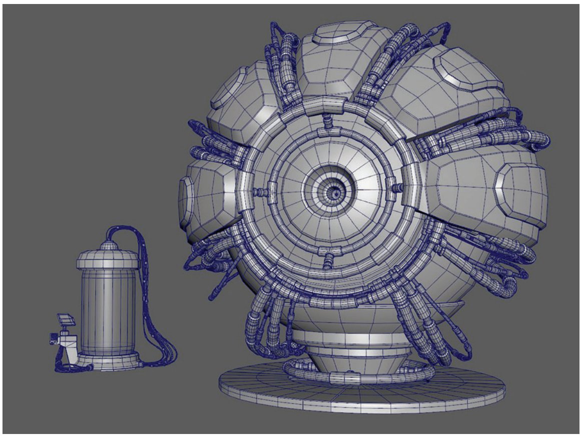


FIGURE 1.66 Hero assets.

Conclusion

We have finished the environment modeling part of our games. Well, of course, we've only created the individualized separated pieces; currently they aren't a level...yet. However, we will move them to the game engine and assemble them into our awesome level later. However, before we do that, we still need to go through UV mapping and Texturing so that our models are not white ghosts.

If you are able to complete these forms in this chapter, you're in good shape. If it is overwhelming and you'd rather move onto other stages, these completed models are available on the support website.

We will move on to the UV Mapping of our assets in the next chapter.

Maya Set UV

UV Mapping is a pretty tricky concept for beginners but quite straightforward after you grasp the essence. It is a 2D coordinate to map a 2D image to the surface of the 3D model. Let us start with creating UVs for our first and simplest modular asset, the 256 × 256 floor piece (Figure 2.1).

The UV Editor

Go to the Workspace at the top right corner of the UI, and in the drop-down list, choose UV Editing; the viewport now splits into two windows. The UV Editor on the right is

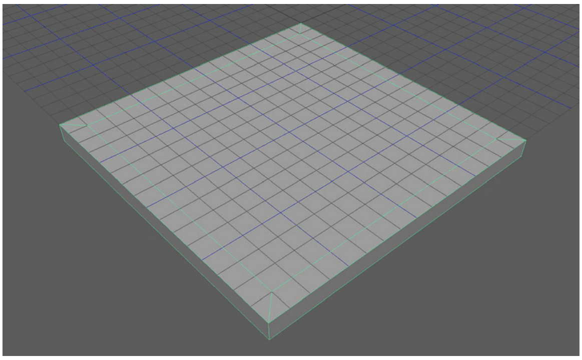


FIGURE 2.1 The 256 by 256 floor piece.

the place we edit our UVs. You can hold down Alt+middle mouse button to pan the view and Alt+right mouse button to zoom in or out. On the right of that window, we can also see a UV Toolkit panel, which contains many useful tools and commands to edit UV.

Select our floor piece; inside the UV Editor, you can see a blue shell that looks like an inverted T, this is the default UV of a cube. If you do recall, we started with a cube (if it is not blue, move the cursor to the UV Editor and press the number 5 button). Click on the checker icon at the row of buttons on top of the UI in the UV Editor. You can now see a checker texture getting displayed in both the UV Editor and on our model in the viewport (Figure 2.2).

Select the top face of the floor, and you can see how a face in the UV Editor is also highlighted. That face in the UV Editor is the UV of the top face of our 3D model. Go to the UV editor, press W to switch to move tool, and move this face in the UV editor to the U letter on the checker texture. You should also see the U letter appearing on the 3D model (Figure 2.3).

This face-to-face match is how UV works. UV is a 2D representation of the 3D model; it defines how an image can be mapped to the surface of the model. UV is also like a flattened shell of the 3D model if you will. This checker is a convenient way to preview how our UV maps texture on our model.