MAYA

Building for speed: Part one

3D computer games have traditionally been hampered by limitations in hardware, but things are changing. In the first of two tutorials, we discover how to create a highly detailed 3D model fit for the next-gen road **BY STEVEN KENT**

FACTFILE

FOR
Maya

DIFFICULTY
Intermediate

TIME TAKEN
3–4 days

ON THE CD
- Full-size screenshots
- Initial, intermediate and final *Maya* scene files

ALSO REQUIRED
N/A

Games artists are never satisfied: they always want more. More power, more polygons and bigger budgets. But while developers might previously have scoffed at these requests, in the world of next-gen games, is possible to render more polygons per frame than ever before, meaning that highly detailed objects can enter play. Take *Project Gotham Racing 3* (*PGR3*), for example: each vehicle weighs in at around 80,000 polygons (40,000 for the exterior and 40,000 for the interior) and the environment swallows up obscene amounts of data. Yet the game itself is still silky smooth.

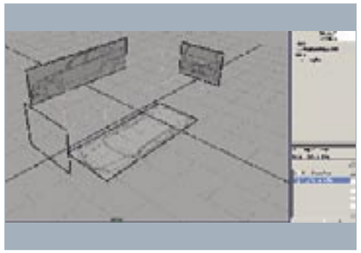
In this two-part tutorial, we will be creating the exterior of a vehicle with similar specifications to *PGR 3*. The first part will focus on modelling, using a combination of 'traditional' polygonal box modelling and subdivision surfaces. Of course, a model as detailed as this could be created using only polygonal tools, but that could take a considerable amount of time, and the result would hardly be as smooth as that created by subdivision surface algorithms – an important factor with an object as finely honed as a vehicle body.

To start with, the scene will be set up in an 'environment' suitable for vehicle creation, using the blueprints provided on the CD. Then we'll create a relatively low-resolution mesh using polygonal modelling tools. This polygonal model will be converted to subdivisions (SubDs) for editing, so that we end up with a smooth surface and edges that look exquisitely crafted. Finally, the SubD mesh will then be converted back to polygons, thereby creating a highly detailed model that can be rendered at very high resolutions for box shots, or as low as your game engine requires for in-game use. The walkthrough assumes that you are already familiar with *Maya*'s modelling workflow, so if you're new to the software, we advise you to check out Autodesk's supporting material first.

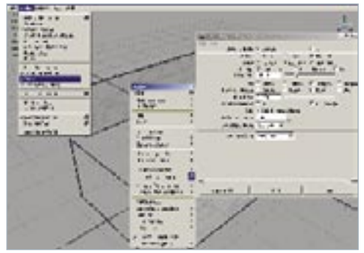
Next issue, we'll explore the process of laying out UVs, setting up materials, and arranging the meshes in a hierarchy ready for export.

Steven Kent is Vehicle Artist at Sony Computer Entertainment Europe, and is in R&D on a next-gen title. His portfolio includes *The Getaway: Black Monday* and *Need for Speed: Underground*
www.scee.com

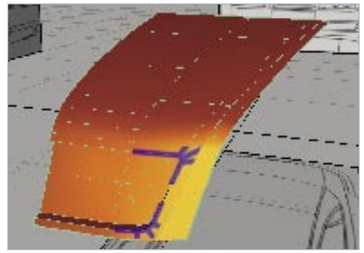
STAGE ONE | Modelling the car



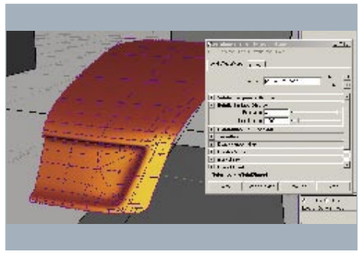
01 Create a new orthographic camera, name it Rear, and set it to Back in the View > Predefined Bookmarks tab. Then create a cube (Create > Polygon Primitives > Cube) with the dimensions of the car, oriented toward +Z. Snap the bottom to the XZ-plane and duplicate the front, side, rear and bottom polygons. Move these polygons out slightly, and assign simple Lambert materials to them containing the cut-out blueprints. (These can be found on the CD as '3dw_blueprints.jpg'.)



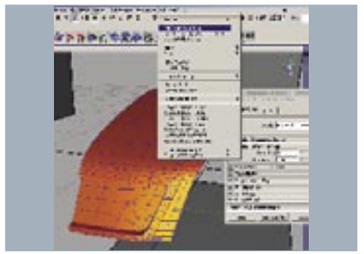
02 Arrange the Bounding Box and the Image Planes in two layers, so the Bounding Box can be set to Template and the Image Planes can be set to Reference. This will make the scene much easier to work with. Set the scene to X-Ray by selecting Shading > X-Ray, so you can see your model through the Image Planes. Finally set the Image Planes to Backface Culling > Keep Wire. (This initial set-up had been done for you in the file provided on the CD: simply load in '1_InitialSetup.ma'.)



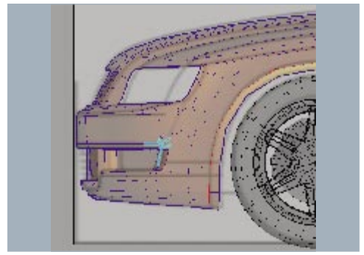
03 Start by modelling the bonnet, using the blueprints as references. Use the Create Polygon tool to outline the shape, then use tools like Split Polygon and Cut Faces to mould a basic shape, by tracing the lines on the front, side and top Image Planes. Keep in mind that a SubD mesh works best with four-sided polygons. The image shows an example of an important spread used to create smooth curves around corners with three subdivisions.



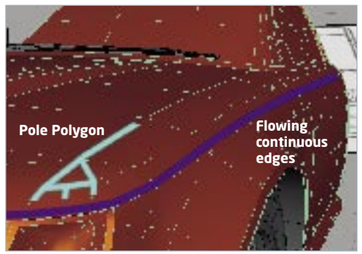
04 As an initial test to see how your mesh smoothes out using SubDs, duplicate your object and convert the shape to a SubD mesh using Modify > Convert > Polygons to Subdiv (using the default settings). Open the Attribute Editor ([Alt]+[a]) and under Subdiv Surface Display change the Resolution to the highest setting (3), which will make the SubD model look very smooth.



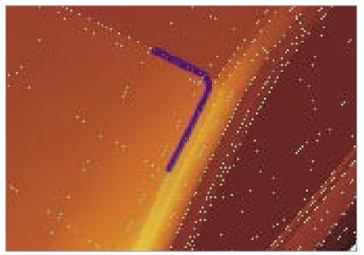
05 You will find the result in Step 4 creates a consistently smooth mesh without taking any hard polygonal edges into account. To solve this, select the edges and vertices you would like to crease, and apply Full Crease Edge/Vertex to them. This will pull the mesh to the edges and vertices, creating a sharp lines and corners.



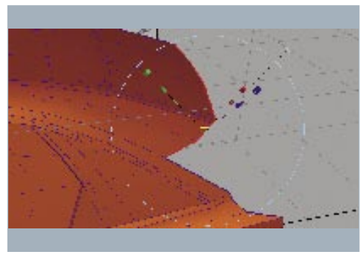
06 Continue modelling using your favourite polygon modelling tools while referencing the model to the Image Planes. Only a few subdivisions around areas like the wheel arch are necessary, as the SubD mesh will create a perfectly round edge if the polygons are sculpted correctly. Remember to use three subdivisions for tight corners.



07 Long continuous edges must be modelled so that they flow without any dents or kinks. This is especially important for shut lines (lines where two body panels meet, but are not joined) and straight panels on the body. To migrate smoothly between high and low resolutions on the mesh, you can use Pole Polygons to create a smooth transition between one and three edges.

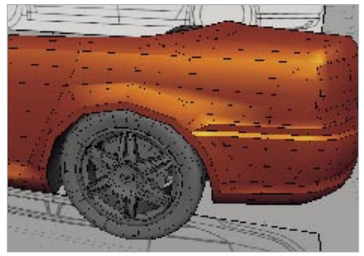


08 To create rectangular corners, use crossing parallel edges to create a pattern similar to a grid, as shown in the image above (a full-size version can be found on the CD). Depending on where you place the fully creased edges after converting to SubDs, a corner can be perfectly sharp, or have a smooth turn.

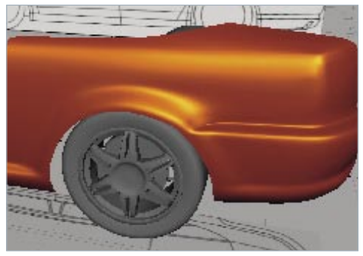


09 Continue modelling using tools like Edit Polygons > Extrude Edge. History must be switched on if you wish to use the Extrude Modifier. The corner of the rear window uses both parallel lines to create a rectangular corner and Pole Polygons to increase the resolution from the centre of the roof to the rear window.

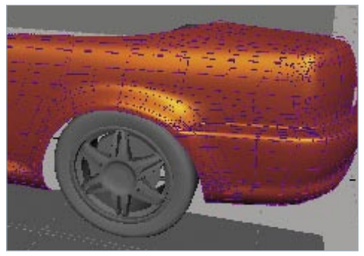
STAGE TWO | Testing the SubD model



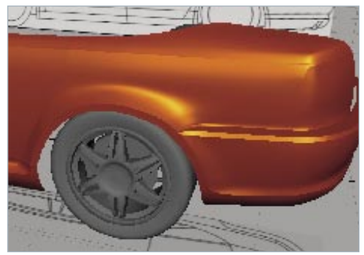
10 Test your wheel arch by converting the rear section to a SubD mesh. The arch shown here has eight main subdivisions, with some edges being followed through to the wheel arch. This ensures the edge loops are set up correctly, so functions like Edit Polygons > Selection > Select Contiguous Edges can be used efficiently. Load in the pre-built wheel ('6_Wheel.ma') to see how it fits into the arch.



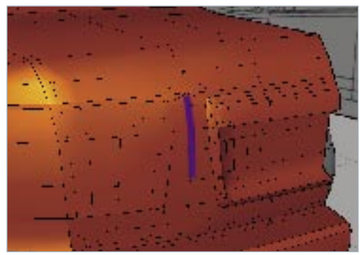
11 Duplicate the mesh and convert to SubDs. As you can see, this creates a smooth, rounded wheel arch. The transition between the rear fascia and fender shows how two edges placed closely together can create a smooth ridge. In fact, the entire car could be created using bevelled edges, so there would be no need to use Full Crease Edge/Vertex, but this would create a very dense mesh.



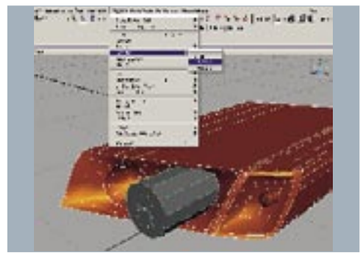
12 Select those edges and vertices that need to be created and use the Subdiv Surfaces > Full Crease Edge/Vertex function to pull the mesh towards the edges and vertices. Partial Crease Edge/Vertex can also be used to pull the mesh only slightly, and this can be applied a number of times to an edge to create an even sharper profile.



13 It's a good idea to repeatedly test any complex areas of your mesh to make sure the SubD mesh flows properly. Here you can see the SubD mesh after all necessary edges and vertices have been creased.

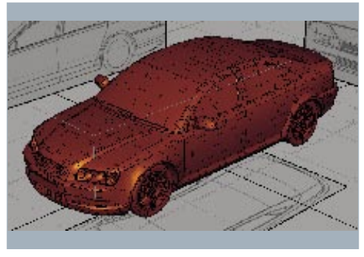


14 One particularly complex area on this model is the part between the rear lights and the extrusion around the centre licence-plate zone. If you have corners close together like this, add a central parallel line so the SubD mesh can converge after it flows around the corners. Load '4_FinishPolyModel.ma' from the CD to test the sample model and convert it to a SubD mesh.

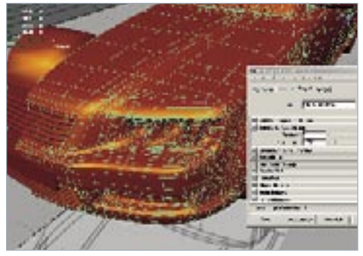


15 To create a round shape for the front light mirrors, use Booleans to cut a cylinder from the mesh. First create a solid/closed object from the area you're cutting into. Then intersect a cylinder (Create > Polygon Primitives > Cylinder) into this, and choose the object to cut into. Now select the object you want to cut with and select Polygons > Booleans > Difference.

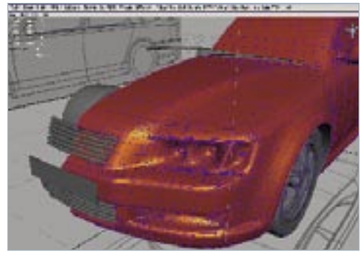
STAGE THREE | Increasing the resolution



16 In its current state, a model at this resolution could be used for action/adventure games where high-resolution objects are inappropriate, as these games require a very high number of different vehicles and other objects to be drawn at the same time. High-resolution models could make the frame rate drop or make objects disappear as the game engine compensates to keep the frame rate up.

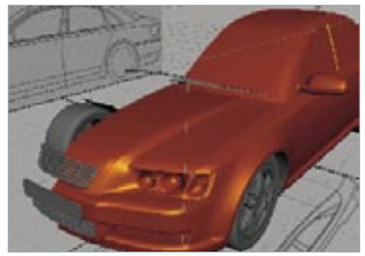


17 You will by now have tested your model enough to be sure that the SubD conversion doesn't hold any surprises. Small inaccuracies can be rectified by right-clicking on the model and selecting Polygon mode in the Hotbox and editing as normal. However, this is very slow, as the SubD mesh is calculated on the fly. Return to Standard mode to use the SubD mesh normally.

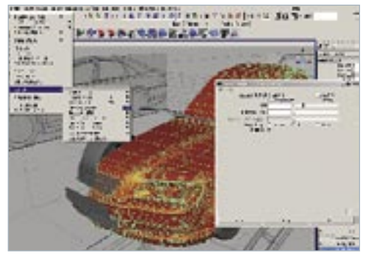


18 Duplicate your poly mesh and keep it as a backup. Now convert your poly mesh to SubDs and go around the model marking all the edges you would like to crease: again, use Subdiv Surfaces > Full Crease Edge/Vertex. Another approach would be to separate your vehicle into panels (as in real-world vehicle construction), and then carry out this step. This, however, is down to personal preference.

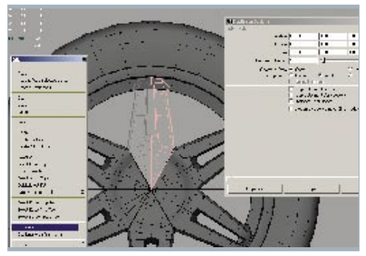
STAGE THREE (Continued) | Increasing the resolution



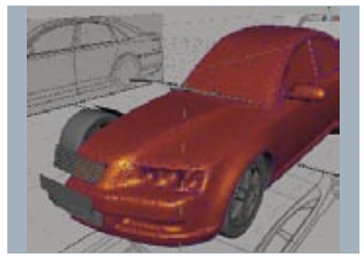
19 Now your model should look something like this. The model is perfectly smooth with sharp corners and edges where the edge crease was set to Full. Experiment with Partially Crease Edge/Vertex to find out if some areas can be pulled close to the edges without creasing them completely.



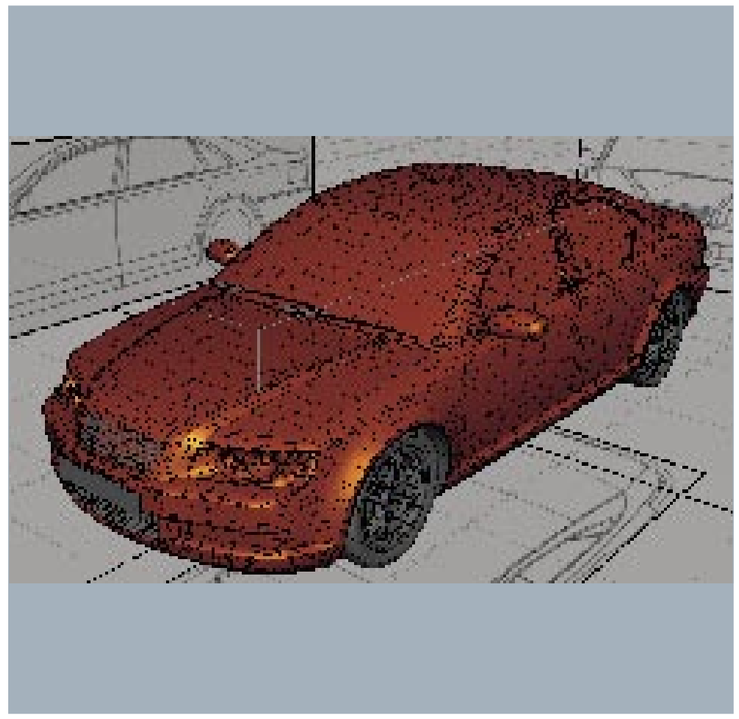
20 Convert the model back into polygons, as this is the way a game engine normally processes 3D objects. Do this by highlighting the model and selecting Modify > Convert > Subdiv to Polygons. This will open a window that allows you to specify the amount of detail on the polygon mesh. Use the settings shown above.

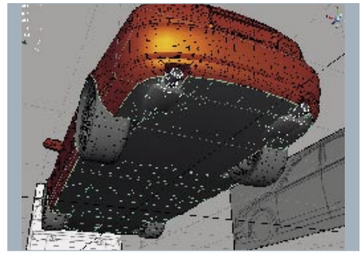


21 A brief note on creating wheels. First create a 'wedge' – half of one spoke – using poly modelling techniques. This can be mirrored along the XY-plane by selecting the options in Edit > Duplicate and setting the Z-scale to -1. Combine the two halves (Polygons > Combine) and duplicate the spoke five times with a rotation around X of 60 degrees. The pivot should be centred to the world.



22 Converting your model back to Polygons will not take Full Creased Edges into consideration, so another step is required to make those creased edges hard. Select those edges you want to make hard and use the Polygon Soften/Harden Edge Options tool under Edit Polygons > Normals > Soften/Harden, setting the Angle to 0.





23 Mirror the mesh to create the missing flank of the car, and create a basic underside that matches up to the wheel arch, running board and front and rear fascias. Also model a simple exhaust object by creating a cube and using Bevel to round off edges and the Boolean tool described earlier to combine a cylinder to the cube and place it in the exhaust cut-outs.

24 For a model to work well in a game environment, it requires a solid and optimised mesh because the number of polygons the gaming environment allows is extremely limited. SubDs are a great way to create accurate objects, while having to model only a lo-res mesh.

This concludes part one of the tutorial. Part two next issue will cover how to optimise the model further and how to lay out UVs for the different body panels and accessories like the front and rear lights, windows and wing mirrors, and how to set up materials for car paint and different textures, including emissive textures, for lights and light glows. It will also cover how to arrange all the meshes in the hierarchy so the entire scene could be exported to a game engine without a hitch. ●