A BUG'S LIFE

How the Bug's world was captured By Mengu Sukan (ms3774) Swati Kumar (sak2144)

DESCRIPTION:

Our scene describes a bug's life which has 2 bugs – ant and butterfly that move about the scene. These are our animated objects. We have 2 light sources one is a directional light source and the other one is a positional one.

Details of all the other requirements are given below:

MODELING:

Requirement 1: You must have enough objects for your scene to be considered 'complete'. If your scene is a barnyard, you should have a windmill, barnhouse, some animals, trees, and moon or sun.

Our Scene: We have the following objects – Ground, SkyBox, Cloud, Tree, Ant(s), Bench, blades of grass, butterfly, sun, mound of soil, flowers, mushrooms, pebbles, coin

Requirement 2: One object must be created by hand. You must explicitly type in the data for vertecies (normals, texcoords, etc.) between a glBegin() and glEnd(). Do not use any tools to help you with this. It will obviously be a rather simple object, but make it more interesting than a box or rectangle.

Our Scene: Ants, pebbles, grass, butterfly, sun, clouds, mound of soil and coin are the objects made by hand.

Requirement 3: One object must be loaded from a Maya .obj file (read in at runtime). You'll have to go online and find a unique object that fits well into your scene. The challange here is to scale and place the object well (no giant cats sticking out of the roof of a car.)

Our Scene: Mushrooms and Bench are Maya objects loaded from a .obj file

Requirement 4: One object must be created from a .raw triangle file (read in at runtime).

Our Scene: The flowers are created using the .raw format.

Requirement 5: You must do all placements, and scaling by hand (or with a code loop). You are not to create a complete scene in some modeling program and then load it all at once.

Our Scene: Everything has been placed by hand and all the transformations, lighting calculations and shadow effects are done by hand

Requirement 6: At least one object must be textured with an image read from a .tga file (or other image file format of your choice)

Our Scene: The Bench has been given a single texture using a .tga file.

Requirement 7: You must texture at least two objects with two different textures. The textures must be placed to fit well on the objects.

Our Scene: The mushrooms and the butterfly (the back and front wings have different textures) have different textures that fit well together.

Requirement 8: At least one object must be shiny, and one dull in appearance.

Our Scene: The prominently shiny object is the coin on the ground. The other objects like butterfly are dull and there are other like flowers and tree leaves that are somewhere in between.

Requirement 9: Your scene must have at least one directional light

Our Scene: There is a directional light that approximates the sun and lights up the entire scene.

Requirement 10: Your scene must have at least one point light source.

Our Scene: We have a spot light just above the coin that projects on to the tree trunk. This is to give the coin an effect like its emitting light from itself. The tree trunk will be brighter than when lit only by the directional light.

Requirement 9: You must pick at least one object to be instantiated more than once. You must do this in a way that does not duplicate the entire object, but merely draws it twice with different ModelView matrices.

Our Scene: There are many objects that repeat in the scene like grass, pebbles, tree branches and leaves, they are all drawn using display lists where the list is called again and again as required.

Requirement 10: You must be able to switch between fill and wireframe rendering for exactly one object.

Our Scene: The ant object can be switched between the solid and the wireframe models that are drawn by hand.

Requirement 11: At least half of the objects in your scene (not counting duplicates) must have correct normals so they interact with the lights correctly.

Our Scene: All our objects have correct normals and interact correctly with light.

Requirement 12: Use double buffering. hidden surface elimination, and a perspective projection.

Our Scene: We are using all 3 as can be seen in display, using <code>glutSwapBuffers()</code> and <code>depth buffer</code>

Requirement 13: You must have some mechanism (a key press is most convenient) to toggle on and off all lights and textures.

Our Scene: Please refer below for details on all the keys used and their functions for controlling the scene.

USER CONTROL AND ANIMATION:

Keyboard:

Key '+': Increases the amount by which camera transformations take place.

Key '-': Decreases the amount by which camera transformations take place.

Key 'w': Moves the camera forward by 1 unit specified by "amount".

Key 's': Moves the camera back by 1 unit specified by "amount".

Key 'a': Moves the camera left by 1 unit specified by "amount".

Key 'd': Moves the camera right by 1 unit specified by "amount".

Key 'c': Straightens up the camera and points to the default original values.

Key 'm': Provides a top view of the entire scene.

Key 'l' (small el): Rotates the camera in the clockwise direction at its given position by "amount" degrees.

Key 'j': Rotates the camera in the anticlockwise direction at its given position by "amount" degrees.

Key 'i': Rotates the camera towards the up direction at its given position by "amount" degrees.

Key 'k': Rotates the camera towards the down direction at its given position by "amount" degrees.

Key 'I': Shifts the eye upwards by 1 unit specified by "amount".

Key 'K': Shifts the eye downwards by 1 unit specified by "amount".

Key 'W': Zoom in towards the current look at point.

Key 'S': Zoom out from the current look at point.

Key '3': Changes the cloud from a 3 D object to a 2 D object.

Key '@': Turns all lighting on and off

Key '1' (one): Turns the directional light on and off.

Key '2': Turns the spot light (special case of a positional light) on and off.

Key 'p': Turns all the animation on or off. It toggles between on and off with the initial value as off.

Key 'Z': Toggles the camera in movie mode or user control mode. By default the camera is in the user control mode.

Key '#': Toggles between wireframe mode and solid mode for the ant. By default the ants are solid.

MOUSE:

Left Click + Drag = Turn 360 degrees about the look at point to view the scene from all angles.

Shift + Left Click+ Drag = Move closer/farther from the look at point and then turn 360 degrees. The motivation is that we can control the details in the scene we are looking at.

ADDED FEATURES

Shadows: We have added shadows that are blended with the ground.

Anti aliasing: To remove some of the jagged effects we are using the built in anti aliasing feature of opengl

Transparency: The clouds and the sun have transparency effects that are controlled using the alpha value.

Curved Surfaces: We use NURBS to draw the mound of soil.

Translation using sine waves: The butterfly flies in a sine wave while flapping its wings and also translates to whatever co-ordinates are specified. This is achieved using vectors and sine equation.

The tree used in the code has been drawn using opengl routines with some modifications using the code given at http://graphics.ethz.ch/teaching/former/imagesynthesis 06/miniprojects/p1/index.html

Realtime rendering of a fractal plant - By Jessy Kiriyanthan, Marcel Meili

SCREENSHOTS:

Figure 1: The opening scene



Figure 1

Figure 2: We can see the grass moving due to animation. The bench using 3D Maya objects is also visible



Figure 2

Figure 3: Notice the mushrooms with double texture.



Figure 3

Figure 4: The shadows for the entire tree and the bench interact with the light correctly. The red flowers are drawn using raw files.



Figure 4

Figure 5: The coin has a shiny appearance with a spot light over it, pointing to the tree.



Figure 5



Figure 6: The sky with clouds and the sun using transparency and double buffering

Figure 6