A picture containing text

Description automatically generated

**CZ2003 COMPUTER GRAPHICS & VISUALIZATION**

**EXPERIMENT 4: IMPLICIT SOLIDS**

**LAB REPORT**

Shearman Chua Wei Jie (U1820058D)

LAB GROUP: SS2

**Complex Implicit Solid**

|  |  |
| --- | --- |
| A picture containing animal  Description automatically generated | The code for the following experiment can be found in the file “lab4.wrl”  On the left is the snapshot of the final implicit solid  generated using the following function:  frep(x,y,z,t){  feet1=min(min(min(z,y),(5/3)\*x - (5/2)\*y - (5/3)\*z + 1),-6\*x - 9\*y - 6\*z);  feet2=min(min(min(z,y),6\*x - 9\*y - 6\*z),-(5/3)\*x - (5/2)\*y - (5/3)\*z + 1);  leg1= min(min(min(0.1^2-(x+0.3)^2-z^2,y),0.4-y),z);  fulleg1= max(min(min(min(0.1^2-(x+0.3)^2-z^2,y),0.3-y),z),min(min(min(z,y),(5/3)\*x - (5/2)\*y - (5/3)\*z + 1),-6\*x - 9\*y - 6\*z));  leg2= min(min(min(0.1^2-(x-0.3)^2-z^2,0.4-y),y),z);  fulleg2=max(min(min(min(0.1^2-(x-0.3)^2-z^2,0.3-y),y),z),min(min(min(z,y),6\*x - 9\*y - 6\*z),-(5/3)\*x - (5/2)\*y - (5/3)\*z + 1));  body = 0.4^2-x^2-((0.7\*(y-0.2))-0.25)^2-(z/0.9)^2;  bod= 0.427^2-x^2-((0.7\*(y-0.06))-0.25)^2-(z/0.9)^2;  body2=0.3^2-x^2-((0.7\*(y-0.15))-0.25)^2-((z-0.2)/0.9)^2;  wing1=1-(5\*z)^2-((10/3)\*y-1.2)^2;  wingfinal=min(min(1-(5\*z)^2-((10/3)\*y-2.1)^2,0.42+x),0.42-x);  wingtap=min(wingfinal,bod);  head=max(min(min(0.3^2-x^2-(y-1.2)^2-z^2,-min(0.05^2-(x+0.15)^2-(y-1.3)^2,z-0.1)),-min(0.05^2-(x-0.15)^2-(y-1.3)^2,z-0.1)),beak);  eye1=min(0.05^2-(x+0.15)^2-(y-1.3)^2,z-0.1);  eye2=min(0.05^2-(x-0.15)^2-(y-1.3)^2,z-0.1);  beak=min((z-0.6)^2 - ((x)^2)/0.2^2 - ((y-1.2)^2)/0.2^2,min(min(0.1^2-(x)^2-(y-1.2)^2,z-0.1),0.5-z));  final = max(max(max(max(max(fulleg1,body),fulleg2),wingtap),head),body2);    return final;}  The complex implicit solid is a penguin and in order to make the final function less complex, each body part of the penguin is broken down and given a label. |
| A picture containing fish, animal  Description automatically generated  Feet    Leg  A picture containing furniture  Description automatically generated  Full Leg | The “feet” of the penguin is created from a pyramid solid object with the plane halfspace z >= 0.  feet2=min(min(min(z,y),6\*x - 9\*y - 6\*z),-(5/3)\*x - (5/2)\*y - (5/3)\*z + 1);  The “leg” of the penguin is created using a cylinder solid object intersected with the plane halfspace z >= 0.  leg2= min(min(min(0.1^2-(x-0.3)^2-z^2,0.4-y),y),z);  By using the union of the “feet” and “leg” objects, we obtain the full leg of the penguin with the equation:  fulleg2=max(min(min(min(0.1^2-(x-0.3)^2-z^2,0.3-y),y),z),min(min(min(z,y),6\*x - 9\*y - 6\*z),-(5/3)\*x - (5/2)\*y - (5/3)\*z + 1)); |
| Body  A picture containing object  Description automatically generated  Full body | The “body” of the penguin is fist created using an ellipsoid according to the following equation:  body = 0.4^2-x^2-((0.7\*(y-0.2))-0.25)^2-(z/0.9)^2;  Then, to make the body look more like the body of a penguin, we union the larger body with another smaller ellipsoid to obtain the final body as shown on the left.  body = 0.4^2-x^2-((0.7\*(y-0.2))-0.25)^2-(z/0.9)^2;  body2=0.3^2-x^2-((0.7\*(y-0.15))-0.25)^2-((z-0.2)/0.9)^2;  fullbod = max(body,body2); |
| Sphere  A close up of a logo  Description automatically generated  Beak  A picture containing object  Description automatically generated  Head | In order to make the “head” of the penguin, we first create a solid sphere object using the equation:  sphere= 0.3^2-x^2-(y-1.2)^2-z^2;  Next, we create the “beak” of the penguin using a cone solid object with the equation:  beak=min((z-0.6)^2 - ((x)^2)/0.2^2 - ((y-1.2)^2)/0.2^2,min(min(0.1^2-(x)^2-(y-1.2)^2,z-0.1),0.5-z));  Finally, the head is obtained by the union of the head and the beak, and the “eyes” of the penguin are obtained by the subtraction of two cylindrical objects from the sphere.  eye1=min(0.05^2-(x+0.15)^2-(y-1.3)^2,z-0.1);  eye2=min(0.05^2-(x-0.15)^2-(y-1.3)^2,z-0.1);  head=max(min(min(0.3^2-x^2-(y-1.2)^2-z^2,-min(0.05^2-(x+0.15)^2-(y-1.3)^2,z-0.1)),-min(0.05^2-(x-0.15)^2-(y-1.3)^2,z-0.1)),beak); |
| A picture containing object  Description automatically generated  Cylinder  A picture containing furniture  Description automatically generated  Wing | In order to make the “wings” of the penguin, we first create an elliptical cylinder solid object as seen on the left with the equation:  wingfinal=min(min(1-(5\*z)^2-((10/3)\*y-2.1)^2,0.42+x),0.42-x);  In order to taper the cylinder so that the wings look more curved, we used the cylinder obtain and intersect it with an ellipsoid object to obtain the figure shown on the left with the equation:  bod= 0.427^2-x^2-((0.7\*(y-0.06))-0.25)^2-(z/0.9)^2;  wingtap=min(wingfinal,bod); |
| A close up of a logo  Description automatically generated | We set a variable colour through the function-defined  diffuse colour according to the equation:  diffuseColor "r=abs(1-2\*u); g=abs(1-2\*v); b=abs(1-2\*w);" |

**Additional figure made with multiple FShape of different colours**

In addition to the compulsory part, I have created a VRML file consisting of many FShapes for which requirements do not have to be observed and is built from different Fshapes individually coloured.

**Complex Implicit Solid**

|  |  |
| --- | --- |
| A picture containing drawing  Description automatically generated | The code for the following experiment can be found in the file “robot.wrl”  The complex implicit solid is a robot and in order to make the final function less complex, each body part of the robot is broken down into different Fshapes and given respective labels. |
| A picture containing object, light  Description automatically generated | The “feet” of the robot consists of 2 box shapes that are defined in the Fshape:  Transform{  translation 0 0 0  children[  FShape {  geometry FGeometry {  # Function script defining the CSG solid.  # Change to some other formulae to see how the solid geometry changes  # based on the parameters domain and the sampling resolution defined below  definition "function frep(x,y,z,t){  box1 = min(min(min(min(min(x+0.25,-0.05-x),y),0.1-y),z+0.2),0.2-z);  box2 = min(min(min(min(min(x-0.05,0.25-x),y),0.1-y),z+0.2),0.2-z);  final = max(box1,box2);    return final;}"  # Adjust the tight bounding box and an optimal resolution  bboxCenter 0 0 0  bboxSize 2 4 2  resolution [100 150 100]  }  appearance FAppearance {  material FMaterial {  # Variable color is defined for the CGS solid  diffuseColor "r= 0.3; g=0.4; b=0;"  } }  }  # Displaying the bounding box with a standard Box object of VRML  # Size of the bounding box goes to size <x y z> while its centre  # goes to translation <x y z>  Transform {translation 0 0 0 children [  Shape {geometry Box {size 2 4 2}  appearance Appearance {material Material  {diffuseColor 0 1 1 transparency 1}}}  ]}  ]  }  The feet are coloured green. |
| A picture containing drawing, computer  Description automatically generated | The “body” and “head” of the robot is defined by the Fshape:  Transform{  translation 0 0 0  children[  FShape {  geometry FGeometry {  # Function script defining the CSG solid.  # Change to some other formulae to see how the solid geometry changes  # based on the parameters domain and the sampling resolution defined below  definition "function frep(x,y,z,t){  box1 = min(min(min(min(min(x+0.25,-0.05-x),y),0.1-y),z+0.2),0.2-z);  box2 = min(min(min(min(min(x-0.05,0.25-x),y),0.1-y),z+0.2),0.2-z);  final = max(box1,box2);    return final;}"  # Adjust the tight bounding box and an optimal resolution  bboxCenter 0 0 0  bboxSize 2 4 2  resolution [100 150 100]  }  appearance FAppearance {  material FMaterial {  # Variable color is defined for the CGS solid  diffuseColor "r= 0.3; g=0.4; b=0;"  } }  }  # Displaying the bounding box with a standard Box object of VRML  # Size of the bounding box goes to size <x y z> while its centre  # goes to translation <x y z>  Transform {translation 0 0 0 children [  Shape {geometry Box {size 2 4 2}  appearance Appearance {material Material  {diffuseColor 0 1 1 transparency 1}}}  ]}  ]  }  The body is coloured grey. |
| A picture containing object, light  Description automatically generated | In order to make the “arms” of the robot, we created two cylinders that are union with two spheres in the Fshape:  Transform{  translation 0 0 0  children[  FShape {  geometry FGeometry {  # Function script defining the CSG solid.  # Change to some other formulae to see how the solid geometry changes  # based on the parameters domain and the sampling resolution defined below  definition "function frep(x,y,z,t){  sphere2 = 0.1^2-(x+0.4)^2-(y-0.75)^2-z^2;  sphere3 = 0.1^2-(x-0.4)^2-(y-0.75)^2-z^2;  arm1 = max(min(min(0.1^2-(x+0.4)^2 -(z)^2,y-0.4),0.75-y),sphere2);  arm2 = max(min(min(0.1^2-(x-0.4)^2 -(z)^2,y-0.4),0.75-y),sphere3);  final = max(arm1,arm2);    return final;}"  # Adjust the tight bounding box and an optimal resolution  bboxCenter 0 0 0  bboxSize 2 4 2  resolution [100 150 100]  }  appearance FAppearance {  material FMaterial {  # Variable color is defined for the CGS solid  diffuseColor "r= 0.502; g=0.502; b=0.502;"  } }  }  # Displaying the bounding box with a standard Box object of VRML  # Size of the bounding box goes to size <x y z> while its centre  # goes to translation <x y z>  Transform {translation 0 0 0 children [  Shape {geometry Box {size 2 4 2}  appearance Appearance {material Material  {diffuseColor 0 1 1 transparency 1}}}  ]}  ]  }  They are coloured grey as well. |
| A picture containing clock, light  Description automatically generated | In order to make the “hands” of the robot, we created the Fshape with 2 cylinders and minus off 2 smaller cylinders:  Transform{  translation 0 0 0  children[  FShape {  geometry FGeometry {  # Function script defining the CSG solid.  # Change to some other formulae to see how the solid geometry changes  # based on the parameters domain and the sampling resolution defined below  definition "function frep(x,y,z,t){  hand1 = min(min(min(0.12^2-(x+0.4)^2 -(y-0.35)^2,z+0.1),0.1-z),-min(min(0.08^2-(x+0.4)^2 -(y-0.33)^2,z+0.1),0.1-z));  hand2 = min(min(min(0.12^2-(x-0.4)^2 -(y-0.35)^2,z+0.1),0.1-z),-min(min(0.08^2-(x-0.4)^2 -(y-0.33)^2,z+0.1),0.1-z));  final = max(hand1,hand2);    return final;}"  # Adjust the tight bounding box and an optimal resolution  bboxCenter 0 0 0  bboxSize 2 4 2  resolution [100 150 100]  }  appearance FAppearance {  material FMaterial {  # Variable color is defined for the CGS solid  diffuseColor "r= 0.502; g=0.502; b=0.502;"  } }  }  # Displaying the bounding box with a standard Box object of VRML  # Size of the bounding box goes to size <x y z> while its centre  # goes to translation <x y z>  Transform {translation 0 0 0 children [  Shape {geometry Box {size 2 4 2}  appearance Appearance {material Material  {diffuseColor 0 1 1 transparency 1}}}  ]}  ]  } |
| A picture containing object, light, clock  Description automatically generated | In order to make the “legs” of the robot, we created the Fshape with 2 cylinders in the color purple:  Transform{  translation 0 0 0  children[  FShape {  geometry FGeometry {  # Function script defining the CSG solid.  # Change to some other formulae to see how the solid geometry changes  # based on the parameters domain and the sampling resolution defined below  definition "function frep(x,y,z,t){  leg1 = min(min(0.1^2-(x+0.15)^2 -(z)^2,y),0.4-y);  leg2 = min(min(0.1^2-(x-0.15)^2 -(z)^2,y),0.4-y);  box1 = min(min(min(min(min(x+0.25,-0.05-x),y),0.1-y),z+0.2),0.2-z);  final = max(leg1,leg2);    return final;}"  # Adjust the tight bounding box and an optimal resolution  bboxCenter 0 0 0  bboxSize 2 4 2  resolution [100 150 100]  }  appearance FAppearance {  material FMaterial {  # Variable color is defined for the CGS solid  diffuseColor "r= 0.3; g=0; b=0.5;"  } }  }  # Displaying the bounding box with a standard Box object of VRML  # Size of the bounding box goes to size <x y z> while its centre  # goes to translation <x y z>  Transform {translation 0 0 0 children [  Shape {geometry Box {size 2 4 2}  appearance Appearance {material Material  {diffuseColor 0 1 1 transparency 1}}}  ]}  ]  } |
| A picture containing drawing  Description automatically generated | The final “robot.wrl” looks as the figure in the left column. |