НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №6

із дисципліни «Математичні та алгоритмічні основи комп'ютерної графіки»

на тему «: Анімація тривимірних об’єктів»

Виконав:

студент 3 курсу групи КП-82

Янкевич Станіслав Дмитрович

Варiант №21

Прийняла:

викл. Шкурат О.С.

КИЇВ — 2021

**Завдання**

Виконати анімацію тривимірної сцени за варіантом

**Варіант** — 1 (21 % 10)

Анімація павука «чорна вдова» black\_widow.obj. Рух ніжок, пересування по екрану.

**Хід виконання**

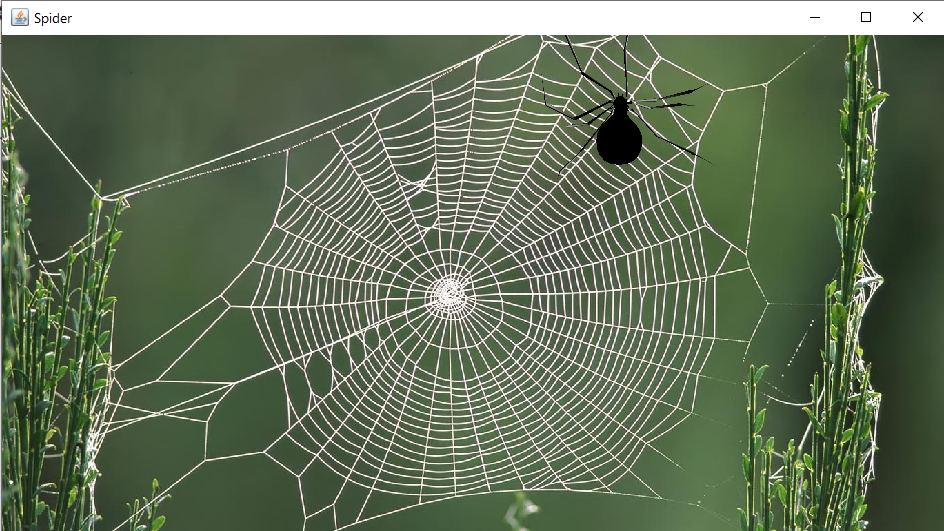
1.  
Налаштування моделi павука

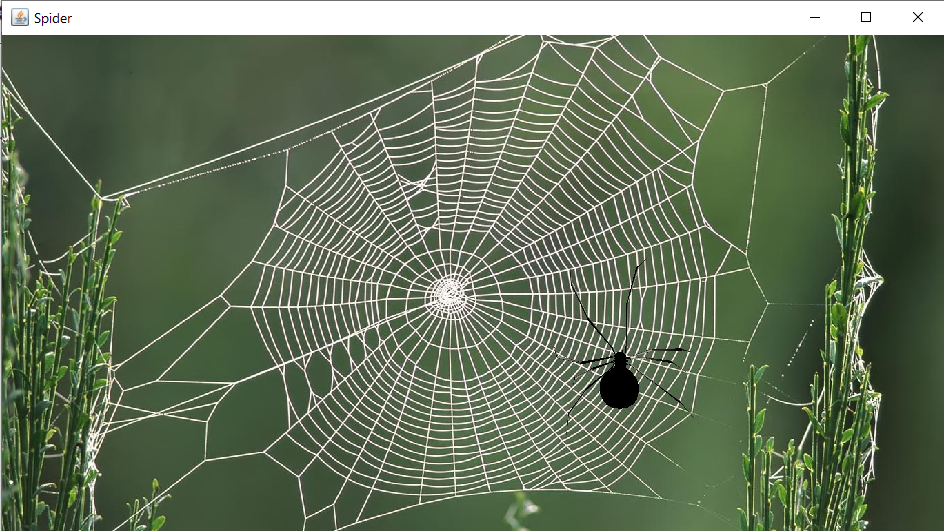
|  |
| --- |
| ObjectFile f = new ObjectFile(ObjectFile.RESIZE);  Scene bugScene = f.load("resources/black\_widow.obj");  roachNamedObjects = bugScene.getNamedObjects();  TransformGroup sceneGroup = new TransformGroup();  BranchGroup scene = new BranchGroup();  Transform3D scaling = new Transform3D();  scaling.setScale(0.20);  Transform3D tfBug = new Transform3D();  tfBug.rotX(3 \* Math.PI / 2);  tfBug.mul(scaling);  tfBug.setTranslation(new Vector3d(0.3f, 0.1f, 0.1f));  TransformGroup tgBug = new TransformGroup(tfBug);  TransformGroup tgBody = new TransformGroup();  Shape3D blkwBody = (Shape3D) roachNamedObjects.get("blkw\_body");  Appearance app = new Appearance();  Color3f black = new Color3f(Color.black);  app.setMaterial(new Material(black, black, black, black, 0f));  blkwBody.setAppearance(app);  tgBody.addChild(blkwBody.cloneTree());  sceneGroup.addChild(tgBody.cloneTree());  BoundingSphere bs = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), Double.MAX\_VALUE);  Transform3D legRotAxis = new Transform3D();  legRotAxis.rotZ(Math.PI / 2);  Transform3D leg2RotAxis = new Transform3D();  int noRotHour = 1000;  int timeRotationHour = 300;  sceneGroup.addChild(getAnimatedLeg("leg1", noRotHour, timeRotationHour, legRotAxis, bs, (float) Math.PI / 8, 100));  sceneGroup.addChild(getAnimatedLeg("leg2", noRotHour, timeRotationHour, legRotAxis, bs, (float) Math.PI / 8, 200));  sceneGroup.addChild(getAnimatedLeg("leg3", noRotHour, timeRotationHour, legRotAxis, bs, (float) Math.PI / 8, 300));  sceneGroup.addChild(getAnimatedLeg("leg4", noRotHour, timeRotationHour, legRotAxis, bs, (float) Math.PI / 8, 300));  sceneGroup.addChild(getAnimatedLeg("leg5", noRotHour, timeRotationHour, leg2RotAxis, bs, -(float) Math.PI / 8, 100));  sceneGroup.addChild(getAnimatedLeg("leg6", noRotHour, timeRotationHour, leg2RotAxis, bs, -(float) Math.PI / 8, 200));  sceneGroup.addChild(getAnimatedLeg("leg7", noRotHour, timeRotationHour, leg2RotAxis, bs, -(float) Math.PI / 8, 300));  sceneGroup.addChild(getAnimatedLeg("leg8", noRotHour, timeRotationHour, leg2RotAxis, bs, -(float) Math.PI / 8, 200));  Transform3D tCrawl = new Transform3D();  tCrawl.rotY(Math.PI / 2);  long crawlTime = 20000;  Alpha crawlAlpha = new Alpha(1, Alpha.INCREASING\_ENABLE, 0, 0, crawlTime, 0, 0, 0, 0, 0);  float crawlDistance = 20.0f;  PositionInterpolator posICrawl = new PositionInterpolator(crawlAlpha, sceneGroup, tCrawl, -3.0f, crawlDistance);  posICrawl.setSchedulingBounds(bs);  sceneGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  sceneGroup.addChild(posICrawl);  tgBug.addChild(sceneGroup);  scene.addChild(tgBug);  addImageBackground("resources/web2.jpg", scene);  scene.compile();  su.addBranchGraph(scene); |

2.  
Створення анімації для лапки

|  |
| --- |
| Alpha legRotAlpha = new Alpha(noRotHour, Alpha.INCREASING\_ENABLE, l, 0, timeRotationHour, 0, 0, 0, 0, 0);  Shape3D leg = (Shape3D) roachNamedObjects.get(elementName);  TransformGroup tgLeg = new TransformGroup();  tgLeg.addChild(leg.cloneTree());  RotationInterpolator legRotation = new RotationInterpolator(legRotAlpha, tgLeg, legRotAxis, v, 0.0f);  legRotation.setSchedulingBounds(bs);  tgLeg.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  tgLeg.addChild(legRotation);  return tgLeg; |

**Результати**

****  
Рис. 1. Проміжний результат виконання програми

  
Рис. 2. Проміжний результат виконання програми

**Висновок**

Виконавши дану лабораторну роботу, ми навчилися анімувати складні об’єкти тривимірної сцени. Також ми дослідили структуру даних scene graph, яка призначена для розташування тривимірних об’єктів у деревовидній структурі, яке повністю визначає склад віртуального всесвіту, а також те, як всесвіт повинен бути відображений.