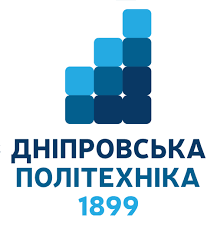
Міністерство освіти і науки України.

НТУ «Дніпровська політехніка».

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ.

Кафедра програмного забезпечення комп'ютерних систем. 

# З дисципліни: Розробка ігрових застосунків на основі jMonkeyEngine 3

**Виконав:**

Студент групи 122-18-1

Мануйленко Іван Миколайович

**Перевірив:**

Реута О.В.

**Дніпро**

**2021**

Лабораторна робота №1

**Створення найпростішої сцени**

Мета роботи – навчитися встановлювати середовище jMonkeyEngine (jME), ознайомитись з його інтерфейсом, ознайомитись із прикладами ігор, створити найпростішу сцену із шаблону і навчитися її модифікувати, розробити власну сцену згідно завдання.

**Варіант 20**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Компонент | Форма | Параметри форми | Параметри трансформації |
| 1 | body | Cylinder | R = 0.5 H = 1 | rotate(FastMath.DEG\_TO\_RAD\*90,0,0) |
| 2 | head | Sphere | R = 0.35 | move(0,0.8f,0) |
| 3 | left arm | Cylinder | R = 0.2 H = 0.7 | move(0.9f,0,0);  rotate(FastMath.DEG\_TO\_RAD\*90, 0, FastMath.DEG\_TO\_RAD\*45); |
| 4 | right arm | Cylinder | R = 0.2 H = 0.7 | move(-0.9f,0,0);  rotate(FastMath.DEG\_TO\_RAD\*90, 0, -FastMath.DEG\_TO\_RAD\*45); |
| 5 | left leg | Cylinder | R = 0.28 H = 0.8 | move(-0.35f,-1,0);  rotate(FastMath.DEG\_TO\_RAD\*90,0,0); |
| 6 | right leg | Cylinder | R = 0.28 H = 0.8 | move(0.35f,-1,0);  rotate(FastMath.DEG\_TO\_RAD\*90,0,0); |
| 7 | hat | Node |  | rotate(-FastMath.DEG\_TO\_RAD\*90,0,0); move(0,1.3f,0) |
| 8 | brim | Torus | RMIN=0.15 RMAX=0.4 |  |
| 9 | crown | Crown | R = 0.2 H = 0.5 | move(0,0,0.4f) |

**Опис завдання**: створіть власну сцену із об’єктом – ігровим персонажем. Персонаж складається з тіла, голови, рук і ніг, форма яких обирається з таблиці для свого номеру варіанта (в таблиці позначені форми: паралелепіпед (Box), циліндр (Cylinder) або сфера (Sphere)). Крім того, у персонажа має бути капелюшок, складний з напівсфера (Dome), площини (Quad) та/або тора (Torus). З яких елементів скласти капелюшок (можна взяти одну елементарну форму, чи більше), обираєте самостійно.

Посилання на віддалений репозиторій **Github** :

**Код:**

**toGetCharacter.java**

/\*\*

 \*

 \* @author Ivan

 \*/

class toGetCharacter {

  Material matForBodyElements ;

  Material mathForHatElements ;

  Vector3f originPos;

    public toGetCharacter(Vector3f originPos,Material mfbe,Material mfhe)

    {

        this.matForBodyElements = mfbe;

        this.mathForHatElements = mfhe;

        this.originPos = originPos;

    }

    public Node  getCharacter()

    {

    Node bodyparts = new Node("bodyparts");

    bodyparts.setLocalTranslation(this.originPos);

    Geometry body = new Geometry("body",new Cylinder(10,10,0.5f,1f,true));

     body.rotate(FastMath.DEG\_TO\_RAD\*90, 0, 0);

     Geometry head = new Geometry("head",new Sphere(10,10,0.35f));

     head.move(0,0.8f,0);

    Geometry rightArm = new Geometry("rightarm",new Cylinder(10,10,0.2f,0.7f,true));

    rightArm.move(0.9f,0,0);

    rightArm.rotate(FastMath.DEG\_TO\_RAD\*90, 0, FastMath.DEG\_TO\_RAD\*45);

    Geometry leftArm = new Geometry("leftarm",new Cylinder(10,10,0.2f,0.7f,true));

         leftArm.move(-0.9f,0,0);

         leftArm.rotate(FastMath.DEG\_TO\_RAD\*90, 0, -FastMath.DEG\_TO\_RAD\*45);

    Geometry leftLeg = new Geometry("leftleg",new Cylinder(10,10,0.28f,0.8f,true));

    leftLeg.move(-0.35f,-1,0);

       leftLeg.rotate(FastMath.DEG\_TO\_RAD\*90,0,0);

    Geometry rightLeg = new Geometry("rightleg",new Cylinder(10,10,0.28f,0.8f,true));

     rightLeg.move(0.35f,-1,0);

     rightLeg.rotate(FastMath.DEG\_TO\_RAD\*90,0,0);

    bodyparts.attachChild(body);

      bodyparts.attachChild(head);

      bodyparts.attachChild(leftArm);

        bodyparts.attachChild(rightArm);

          bodyparts.attachChild(leftLeg);

            bodyparts.attachChild(rightLeg);

           List<Spatial> bparts =  bodyparts.getChildren();

           for(Spatial part : bparts)

           {

                 if(part instanceof Geometry)

                 {

                     ((Geometry)part).setMaterial(this.matForBodyElements);

                 }

           }

    Node hat = new Node("head");

    Geometry headGear = new Geometry("headgear",new Torus(10,10,0.15f,0.4f));

    Geometry headCylinder = new Geometry("headcylinder",new Cylinder(10,10,0.2f,0.5f,true));

    hat.attachChild(headGear);

    hat.attachChild(headCylinder);

    hat.rotate(-FastMath.DEG\_TO\_RAD\*90,0,0);

    hat.move(0,1.3f,0);

    hat.setMaterial(this.mathForHatElements);

    headCylinder.move(0,0,0.4f);

    bodyparts.attachChild(hat);

    return bodyparts;

    }

}

**Main.java**

public class Task4 extends SimpleApplication {

    public static void main(String[] args) {

        Task4 app = new Task4();

        app.start();

    }

    @Override

    public void simpleInitApp() {

            Material matForBody = new Material(assetManager,

                "Common/MatDefs/Misc/Unshaded.j3md");

              matForBody.setColor("Color", ColorRGBA.Red);

               Material matForHat = new Material(assetManager,

                "Common/MatDefs/Misc/Unshaded.j3md");

              matForHat.setColor("Color", ColorRGBA.Blue);

       toGetCharacter tgc = new toGetCharacter(Vector3f.ZERO,matForBody,matForHat);

       rootNode.attachChild(tgc.getCharacter());

    }

    @Override

    public void simpleUpdate(float tpf) {

        //TODO: add update code

    }

    @Override

    public void simpleRender(RenderManager rm) {

        //TODO: add render code

    }

}

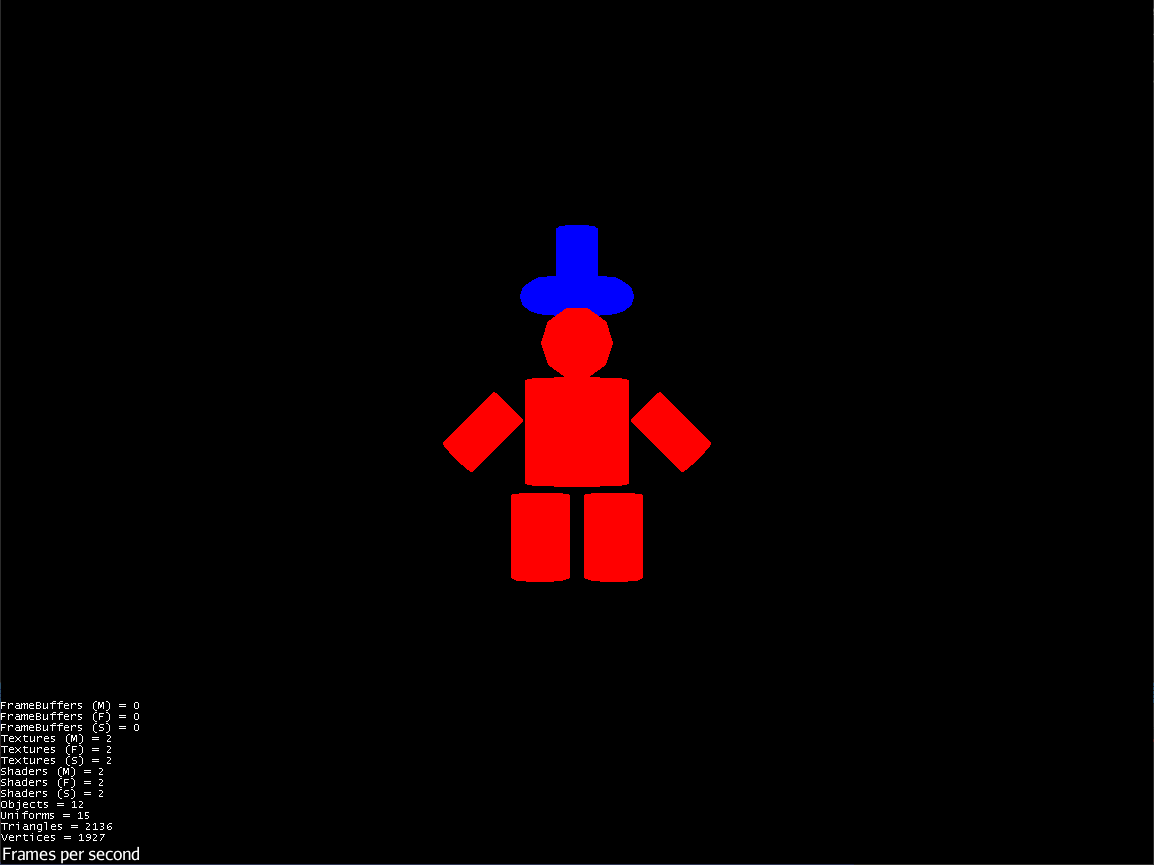


Рисунок 1 Вигляд персонажу

Висновок: я дізнався про основні можливості візуалізації деяких елементів фігур, для створення більш компонентних об’єктів , про можливості вкладення елементів один до одного, для подальшого компонування, та про базові можливості конфігурації матеріалів.