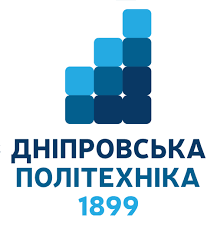
Міністерство освіти і науки України.

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 Павло 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 21**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Компонент | Форма | Параметри форми | Параметри трансформації |
| 1 | body | Sphere | R = 0.7 |  |
| 2 | head | Box | H=0.5;W=0.5;L=0.5; | move(0,1.5f,0); |
| 3 | left arm | Box | H=0.5;W=0.5;L=0.5; | move(-1f,1f,0); |
| 4 | right arm | Box | H=0.5;W=0.5;L=0.5; | move(1f,1f,0); |
| 5 | left leg | Box | H=0.5;W=0.5;L=0.5; | move(-0.7f,-1.2f,0); |
| 6 | right leg | Box | H=0.5;W=0.5;L=0.5; | move(0.7f,-1.2f,0); |
| 7 | hat | Node |  | rotate(-FastMath.PI / 2, 0, 0)  move(0, 2.2f, 0) |
| 8 | brim | Torus | RMIN=0.1 RMAX=0.5 |  |
| 9 | crown | Crown | R=0.4 H=0.6 | move(0, 0, 0.3f) |

**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** :

<https://github.com/Dozez13/JmonkeyEngine>

**Код**

**Person class**

/\*\*

 \*

 \* @author Pavlo

 \*/

public class Person {

     private final Material body;

     private final Material hat;

    public Person(Material body,Material hat){

      this.body = body;

      this.hat = hat;

    }

    public Node personToNode(){

        Geometry bodys = new Geometry("body",new Sphere(100,100,0.8f));

        Geometry head = new Geometry("head",new Box(0.5f,0.5f,0.5f));

        Geometry leftHand = new Geometry("leftHand",new Box(0.5f,0.5f,0.5f));

        Geometry rightHand = new Geometry("rightHand",new Box(0.5f,0.5f,0.5f));

        Geometry leftLeg = new Geometry("leftLeg",new Box(0.5f,0.6f,0.5f));

        Geometry rightLeg = new Geometry("rightLeg",new Box(0.5f,0.6f,0.5f));

        Geometry brim = new Geometry("crown",new Torus(20,20,0.1f,0.5f));

        Geometry crown = new Geometry("brim",new Cylinder(20,20,0.4f,0.6f,true));

        Node hatElements = new Node("hat");

        hatElements.attachChild(crown);

        hatElements.attachChild(brim);

        crown.move(0,0,0.3f);

        hatElements.rotate(-FastMath.PI/2,0,0);

        hatElements.move(0,2.5f,0);

        Node person = new Node("Body");

        person.attachChild(leftLeg);

        person.attachChild(rightLeg);

        person.attachChild(leftHand);

        person.attachChild(rightHand);

        person.attachChild(bodys);

        person.attachChild(head);

        leftLeg.move(-0.7f,-1.2f,0);

        rightLeg.move(0.7f,-1.2f,0);

        leftHand.move(-1f,0.6f,0);

        rightHand.move(1f,0.6f,0);

        head.move(0,1.8f,0);

        List<Spatial> spatials = person.getChildren();

        for(Spatial sp:spatials){

            if(sp instanceof Geometry){

                ((Geometry)sp).setMaterial(this.body);

            }

        }

          spatials = hatElements.getChildren();

        for(Spatial sp:spatials){

            if(sp instanceof Geometry){

                ((Geometry)sp).setMaterial(this.hat);

            }

        }

        Node all = new Node("all");

        all.attachChild(person);

        all.attachChild(hatElements);

        return all;

    }

}

**Main class**

public class Main extends SimpleApplication {

    public static void main(String[] args) {

        Main app = new Main();

        app.start();

    }

    @Override

    public void simpleInitApp() {

        Material body = new Material(assetManager,"Common/MatDefs/Misc/Unshaded.j3md");

        body.setColor("Color",ColorRGBA.Yellow);

        Material hat = new Material(assetManager,"Common/MatDefs/Misc/Unshaded.j3md");

        hat.setColor("Color",ColorRGBA.Green);

       Person person = new Person(body,hat);

        rootNode.attachChild(person.personToNode());

    }

}

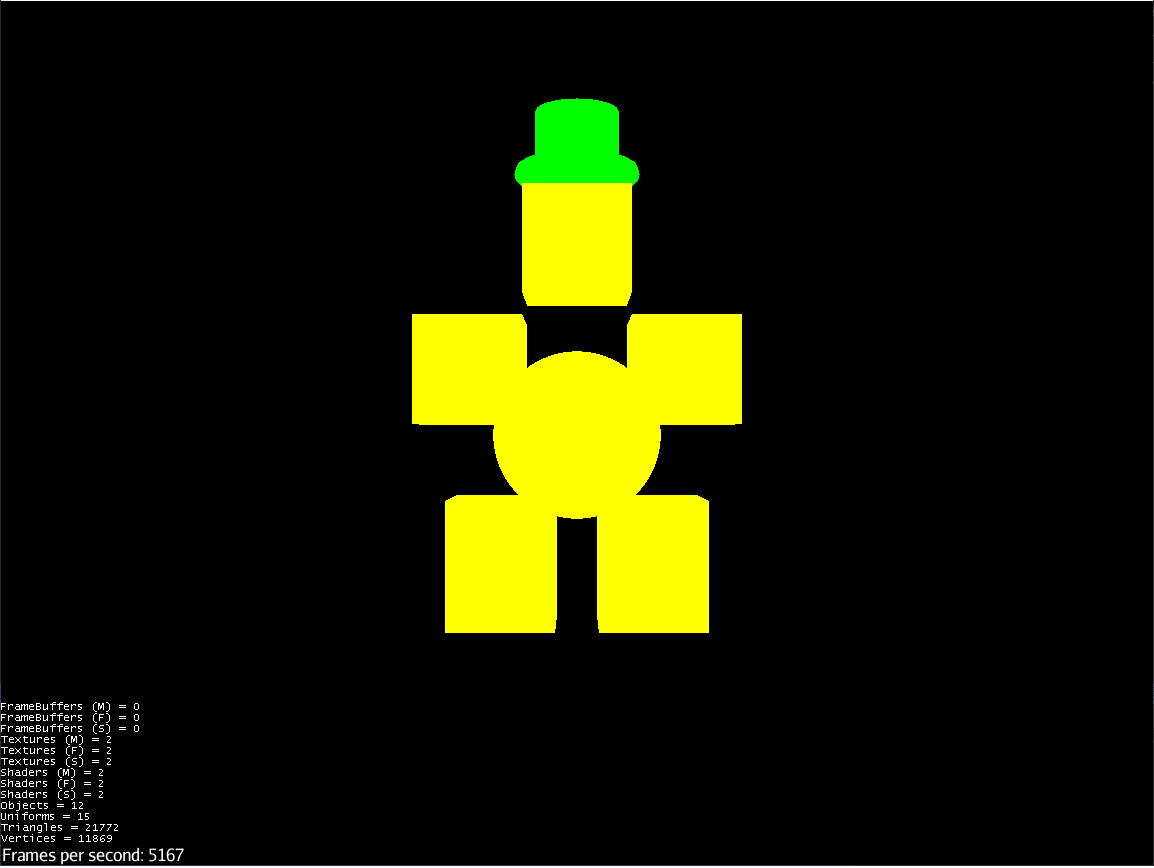


Рисунок 1 Вигляд персонажу

Висновок: я дізнався про основні фігури, які використовуються у JME, а також зрозумів, що являє собою сцена у движку і її основні елементи.