

# GAME DEVELOPMENT

## Esercizio 1

# SUPSI

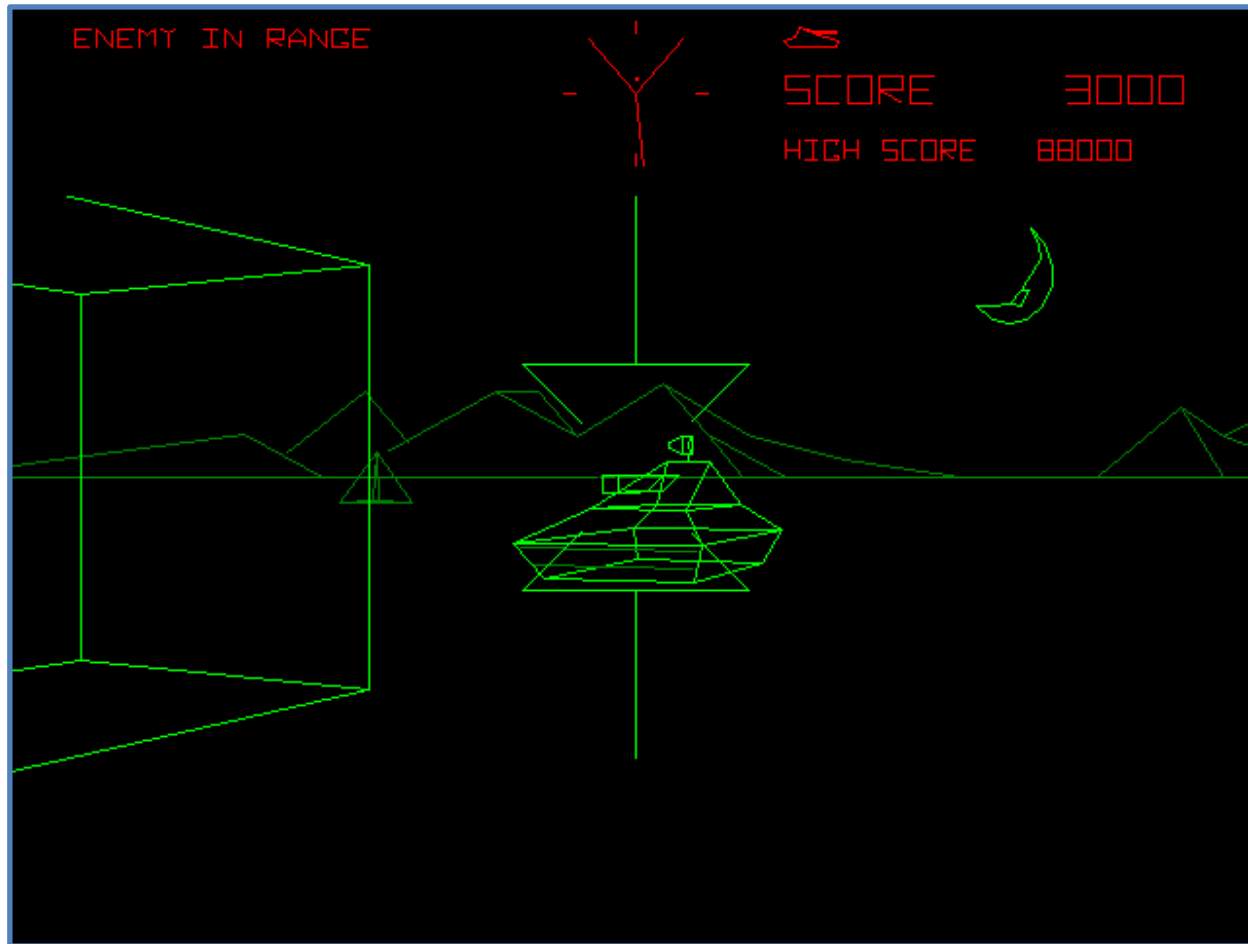
Contenuto realizzato da:  
Marino Alge

Scuola universitaria professionale della Svizzera italiana  
Dipartimento tecnologie innovative

**Marino Alge Eng. ETH**  
[marino.alge@supsi.ch](mailto:marino.alge@supsi.ch)

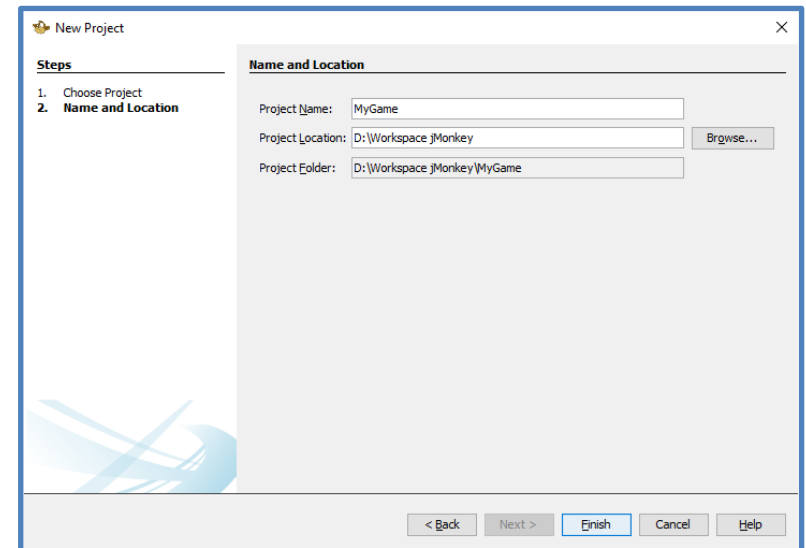
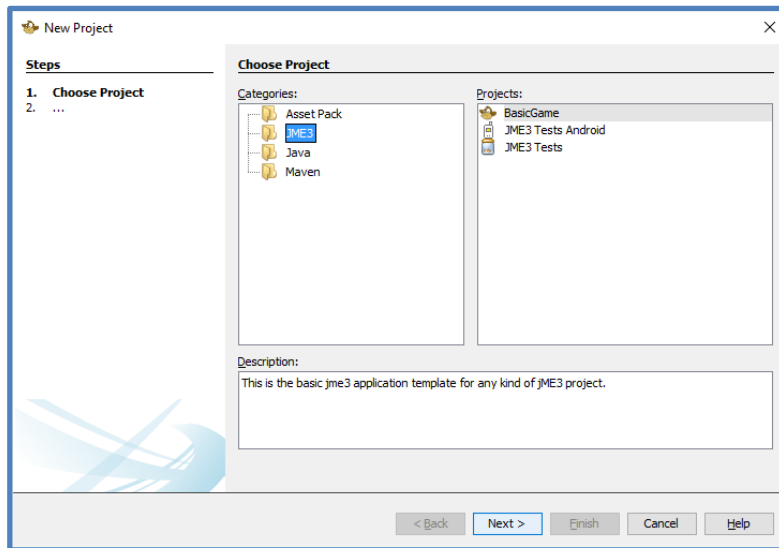
# BATTLEZONE

Il **progetto** sarà ispirato ad un vecchissimo gioco chiamato **battlezone**, un gioco d'azione dove carri armati futuristici combattono tra loro.



# NUOVO PROGETTO JMONKEY

Per cominciare creiamo un nuovo progetto jMonkey chiamato «**Esercizio1**»  
«File» → «new project» → «JM3 / BasicGame»



# CLASSE MAIN

Nella classe **Main** (application entry) troviamo già un po di codice d'esempio

```
public class Main extends SimpleApplication {
```

```
    public static void main(String[] arguments) {  
        Main main = new Main();  
        main.start();  
    }
```

Entry point

```
@Override
```

```
public void simpleInitApp() {  
    Box box = new Box(1, 1, 1);  
  
    Geometry geometry = new Geometry("Box", box);  
  
    Material material = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");  
    material.setColor("Color", ColorRGBA.Blue);  
    geometry.setMaterial(material);  
  
    rootNode.attachChild(geometry);  
}
```

Inizializzazione,  
chiamata prima  
che il gioco parta

```
@Override
```

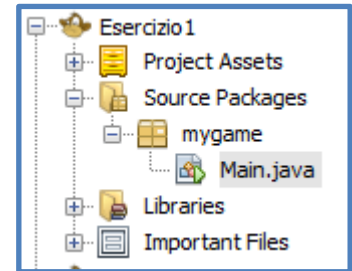
```
public void simpleUpdate(float tpf) {  
}
```

Update, chiamata  
ogni frame (grafico)

```
@Override
```

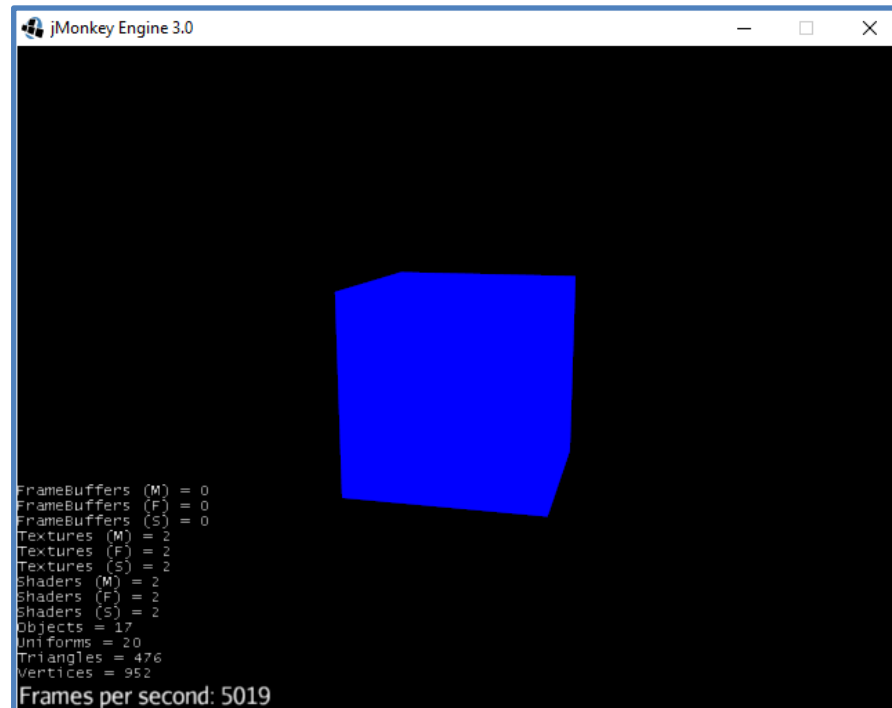
```
public void simpleRender(RenderManager renderManager) {  
}
```

```
}
```



# ESEMPIO

Se lanciamo l'esempio (ctrl+f5) vediamo un cubo blu sospeso per aria a cui possiamo lentamente navigare attorno con «wasd,qz + mouse»



**Tips:** Per velocizzare la camera si puo usare

```
public void simpleInitApp() {  
    flyCam.setMoveSpeed(16.0f);  
}
```

# CODICE D'ESEMPIO

Il cubo blu viene creato qui:

```
flyCam.setMoveSpeed(16.0f);  
Box box = new Box(1, 1, 1);  
Geometry geometry = new Geometry("Box", box);  
Material material = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");  
material.setColor("Color", ColorRGBA.Blue);  
geometry.setMaterial(material);  
rootNode.attachChild(geometry);
```

Crea un cubo  
(mesh) di 2x2x2

Crea un spatial  
Geometry di nome  
«box» e con il cubo  
come mesh

Crea un materiale  
di tipo «unshaded»

Aggiunge il spatial  
geometry alla root  
del scene-graph

Assegna il materiale  
alla geometria

Setta il colore del  
materiale a blu

# ESERCIZIO 1.1

## Esercizio 1.1

Creare una griglia 256x256 WU (world unit) come «pavimento» per il nostro gioco

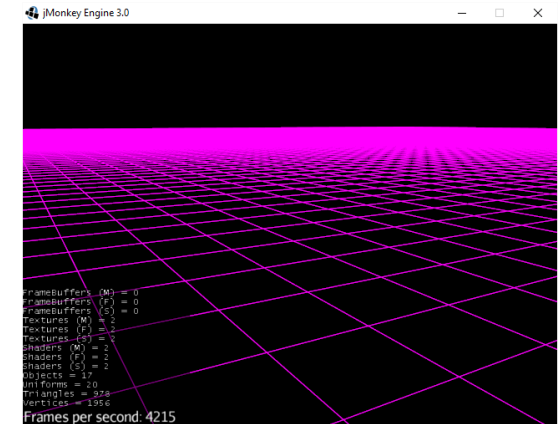
- Una griglia si crea utilizzando **Grid** (sottoclasse di **Mesh**)

```
Grid grid = new Grid(xLines, yLines, lineDistance);
```

- La griglia dev'essere centrata in 0,0,0 usare **setLocalTranslation()** per centrarla

Tips: Creare una classe **GridFactory**

```
public class GridFactory {  
  
    private final AssetManager assetManager;  
  
    public GridFactory(AssetManager assetManager) {...}  
  
    public Node newGrid(String name) {...}  
  
}
```



## ESERCIZIO 1.2

Creare un **tank** utilizzando come geometria le **figure primitive box e cylinder** (utilizzare unità di misura ragionevoli  $1 \text{ WU} = 1 \text{ m}$ )

```
Box box = new Box(halfWidth, halfHeight, halfDepth);  
Cylinder cylinder = new Cylinder(axisSamples, radialSamples, radius, height, closed);
```

Il tank deve essere composto da **3 elementi**:

- **Chassis** (telaio)
- **Turret** (torretta)
- **Cannon** (cannone)



Creare un sotto-albero che rispecchia la **gerarchia cinematica**

Tips: Creare una classe **TankFactory**

```
public class TankFactory {  
    private final AssetManager assetManager;  
  
    public TankFactory(AssetManager assetManager) {...}  
  
    public Node newTank(String name) {...}  
}
```



## ESERCIZIO 1.3

**Animare** il tank nel metodo `simpleUpdate()` della classe `Main` nel seguente modo:

- Il telaio deve roteare di 1 rad/sec in senso anti-orario attorno l'asse y (il tank gira sul posto)
- La torretta deve roteare di 3 rad/sec in senso orario attorno l'asse y (la torretta gira in senso opposto al triplo della velocità)
- Il cannone deve cambiare l'elevazione fra -10 gradi (punta a terra) a 45 gradi (punta in alto) in moto sinusoidale.

Correggere l'origine (il 0,0,0) delle varie geometrie (chassis, turret, cannon) in modo che ruotino attorno al punto corretto.

**Tips:** `FastMath` è la variante veloce e dedicata di `Math` in jMonkey, usatela.

## ESERCIZIO 1.3 TIPS

Creare una classe **d'utilità** con metodi per semplificarsi la vita

```
public class Utils {  
  
    public static <T extends Spatial> T getChild(Node parent, String name, Class<T> spatialClass) {  
        T result = null;  
        Spatial spatial = parent.getChild(name);  
        if (spatialClass.isAssignableFrom(spatial.getClass())) {  
            result = (T) spatial;  
        }  
        return result;  
    }  
}
```

Getta il figlio della  
root di nome (sid)  
«tank»

Tempo in sec dal  
ultimo update

Buon **metodo per roteare**:

```
public void simpleUpdate(float tpf) {  
    Node tank = Utils.getChild(rootNode, "tank", Node.class);  
    if (tank == null) {  
        return;  
    }  
    angle += tpf * speed; // speed: RAD/sec  
    angle %= FastMath.TWO_PI; // [-2PI..2PI]  
    tank.setLocalRotation(new Quaternion(new float[] {0.0f, angle, 0.0f}));  
}
```

Incrementa l'angolo  
a velocità **speed**

Mantiene la  
precisione del float

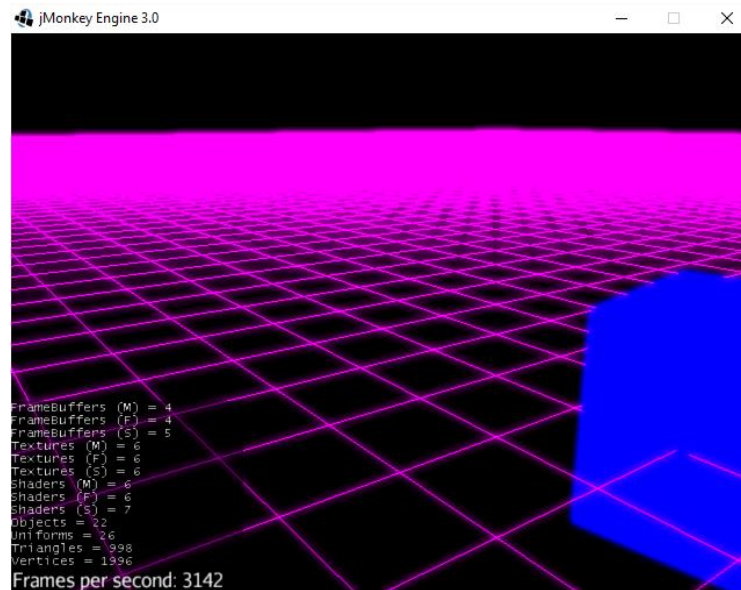
Angoli d'eulero

## ESERCIZIO 1.4 (OPZIONALE)

Aggiungere un filtro di **bloom** in post processing:

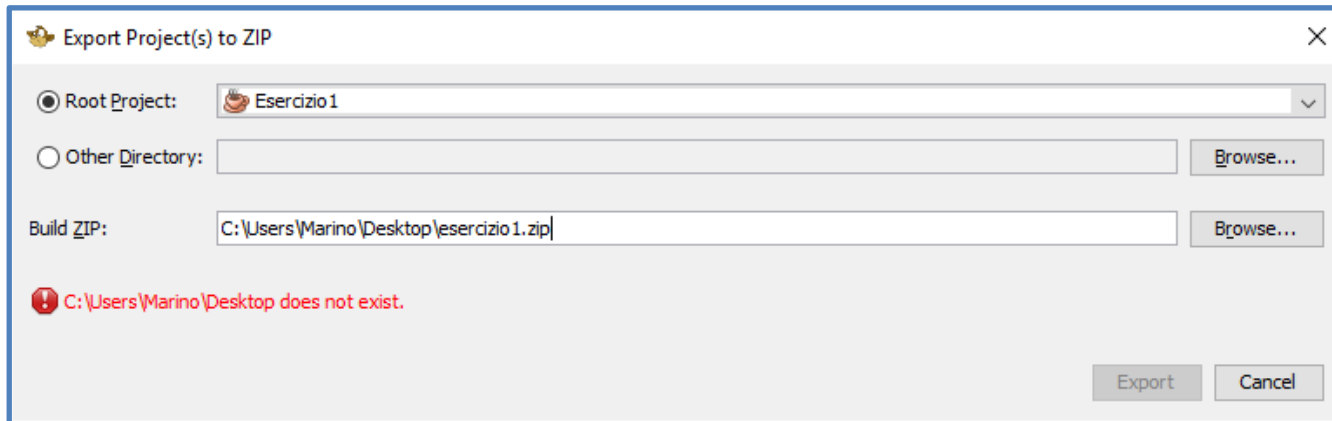
```
BloomFilter bloomFilter = new BloomFilter(BloomFilter.GlowMode.Scene);  
bloomFilter.setBlurScale(1.5f);  
bloomFilter.setExposurePower(5.0f);  
bloomFilter.setExposureCutOff(0.0f);  
bloomFilter.setBloomIntensity(2.0f);  
FilterPostProcessor filterPostProcessor = new FilterPostProcessor(assetManager);  
filterPostProcessor.addFilter(bloomFilter);  
viewport.addProcessor(filterPostProcessor);
```

Giocare con i **parametri**



# CONSEGNA

**Esportare** il progetto «Esercizio1» in  
«File» → «export project» → «to Zip...»



**Consegnare** lo zip sul sito **iCorsi**

# THE END

Esercizio 1

# SUPSI

Contenuto realizzato da:  
Marino Alge

Scuola universitaria professionale della Svizzera italiana  
Dipartimento tecnologie innovative

**Marino Alge Eng. ETH**  
[marino.alge@supsi.ch](mailto:marino.alge@supsi.ch)