

I-SEB 18/19 S2 Deeltijd

Tips gebruik Game Engine

ICA

INFORMATICA
COMMUNICATIE
ACADEMIE

TIPS BIJ GEBRUIK GAME-ENGINE

COURSE naam SEB

Studiejaar 2018-2019

Opleiding HBO-ICT deeltijd

Inleiding

In hoofdlijnen wordt hieronder aangegeven hoe de Engine te gebruiken. Indien nodig kan de docent dit toelichten. Daarnaast zijn er diverse tips opgenomen voor het gebruik van de Engine. De Engine is niet 100% perfect, dus mocht je nog aanvullende tips hebben, laat het dan aan de docent weten.

Gebruik Engine in hoofdlijnen

Centraal staat je "wereld": de klasse die erft van GameEngine. Dit is het startpunt van je game (public static void main). Hierin worden zaken geïnitieerd, zoals:

- hoe ziet wereld eruit?
- welke objecten worden in het begin aangemaakt?
- laden van eerder opgeslagen gegevens
- etc.

Een klasse die erft van GameObject is een object in het spel. Deze kan bewegen of stilstaan. Deze kan zelfstandig bewegen (zoals vijand) of bestuurd worden (speler). Er zijn verschillende varianten:

- **GameObject** teken je zelf met Processingcode
- **SpriteObject** bevat een plaatje
- **AnimatedSpriteObject** bevat meerdere plaatjes die "vanzelf" veranderen, of bijv. doordat speler van richting verandert, plaatjes zitten als "frames" in 1 bestand

View en viewport:

In klasse die erft van GameEngine moet je een view aanmaken waarmee je bepaalt hoe je spel getoond wordt. Als je wereld niet in het venster past, maak je gebruik van een Viewport: voorbeeldcode in WaterWorld.

Tips

Hieronder vind je enkele tips die je kunnen helpen bij het maken van een game met OOPG. Mocht je zelf tips hebben die hieraan toegevoegd zouden kunnen worden, geef die dan door aan je docent.

Programma crasht bij implementeren ICollidableWithTiles

Als je nog geen tilemap hebt aangemaakt in je hoofdprogramma, en je hebt wel een klasse die de interface ICollidableWithTiles implementeert, crasht je programma. Los dit op door alvast een (kleine) tilemap te maken, of door de klassen tijdelijk deze interface niet te laten implementeren.

Sprite roteren

Als je een sprite wil laten roteren, bijvoorbeeld een kogel afhankelijk van de hoek waarin deze geschoten is, kun je dat doen door de draw-methode te overriden, en deze de onderstaande inhoud te geven (afhankelijk van je specifieke wensen, kun je andere waarden geven aan de rotatie en het draaipunt)

Let op: door rotatie zullen de hoogte en breedte van je object veranderen. Om deze goed te kunnen gebruiken, zul je de methoden getX(), getY(), getWidth() en getHeight() moeten gebruiken (en dus niet rechtstreeks de variabelen uitlezen).

```
private float rotatiehoek = 45;
```

```
@Override
```

```
public void draw(PGraphics g) {
```

```
    g.pushMatrix();
```

```
    g.translate(getCenterX(), getCenterY());
```

```

g.rotate(PApplet.radians(rotatiehoek));

g.image(getImage(), -width / 2, -height / 2);

g.popMatrix();
}

public static float getRotationInRadians(float rotationInDegrees) {
    float rotationInRadians = (float)
(PApplet.radians(rotationInDegrees) % Math.PI);

    rotationInRadians = (float) (((rotationInRadians > Math.PI *
0.5 && rotationInRadians < Math.PI * 1)

        || (rotationInRadians > Math.PI * 1.5
&& rotationInRadians < Math.PI * 2)) ? Math.PI - rotationInRadians
        : rotationInRadians);

    return rotationInRadians;
}

@Override

public float getWidth() {
    // met dank aan

    // http://stackoverflow.com/questions/10392658/calculate-the-bounding-boxes-x-y-height-and-width-of-a-rotated-element-via-jav

    float rotationInRadians = getRotationInRadians(rotatiehoek);
    return (float) (Math.sin(rotationInRadians) * height +
        Math.cos(rotationInRadians) * width);
}

@Override

public float getHeight() {

```

```

        float rotationInRadians = getRotationInRadians(rotatiehoek);

        return (float) (Math.sin(rotationInRadians) * width +
Math.cos(rotationInRadians) *height);

    }

    @Override

    public float getX() {

        return -(getWidth() / 2) + getCenterX();

    }

    @Override

    public float getY() {

        return -(getHeight() / 2) + getCenterY();

    }

```

Element op dashboard wordt niet getekend

Soms wordt een object op een dashboard niet goed getekend. Dit kun je oplossen door bij het toevoegen van het object aan het dashboard een hoog getal te zetten (bijv. 100) bij de layerposition. Dus bijvoorbeeld:

```
dashboard.addGameObject(dashboardText,x,y,100);
```

Teksten zijn niet zichtbaar

Als je een tekst toevoegt en deze is onzichtbaar, zou het kunnen dat deze zaken het probleem oplossen:

- Zie "Element op dashboard wordt niet getekend"
- Zorg dat de kleur van de tekst goed ingesteld is (met fill, en bij het gebruik van een alpha-waarde een waarde van (veel) meer dan 0).
- Houd er rekening mee dat in Processing teksten worden getekend vanaf linksonder, dus als je een tekst tekent op 0,0 (en bij gebruik van een dashboard bovenin het beeld is dat niet zo vreemd), is de tekst er wel, maar valt hij buiten beeld.

Meerdere toetsen tegelijk kunnen indrukken

Bij gebruik van de methode `keyPressed` kan maar één toets tegelijkertijd worden afgehandeld. Wil je meerdere toetsen tegelijk willen kunnen indrukken (bijvoorbeeld springen en in een bepaalde richting bewegen tegelijk), dan kun je daarvoor de volgende oplossing implementeren:

- Maak een klasse (bijvoorbeeld met de naam `Toets`) met daarin twee variabelen:
 - Een `char` voor de ingedrukte toets
 - Een boolean die aangeeft of de toets is ingedrukt, met daarbij een getter en een setter
- Maak een arraylist aan waarin je voor elke toets die je wil ondersteunen een object van het type `Toets` zet, plaats deze in de klasse die de toetsafhandeling regelt (mogelijk is dat de speler)
- Zorg dat de `keyPressed`-methode van die klasse door de arraylist loopt en het object behorende bij de ingedrukte toets op `true` zet (via de setter).
- Doe hetzelfde bij de `keyReleased`-methode, maar zet daar de waarde op `false` (via de setter).
- In de `update`-methode van de klasse kun je nu door de arraylist lopen en nagaan welke toets ingedrukt was. Let op: als een toets ingedrukt blijft, wordt dit in elke update geconstateerd, dus mogelijk zul je iets moeten inbouwen dat voorkomt dat de actie heel vaak herhaald wordt.

Meerdere soorten tiles toevoegen

1. Alle tiles moeten vierkant zijn en ze moeten allemaal even groot zijn (zo niet, dan zal de engine ze schalen).
2. Maak een klasse aan die erft van `Tile` en die dezelfde inhoud heeft als `BoardTile` in `WaterWorld` (bijv. `wallTile`)
3. Maak in je hoofdklasse (die erft van `GameEngine`) op de volgende manier een nieuwe tiletype aan:
`Sprite wallSprite = new Sprite("src/main/java/nl/han/ica/mijnGame/media/wall.jpg");`
1.
`TileType wallTileType = new TileType<>(WallTile.class, wallSprite);`
2. Zet alle `TileTypes` in een `TileType`-array (in `WaterWorld` zie je nu een array met slechts 1 element: `boardTileType`, als we daar de `wallTileType` aan zouden toevoegen, zou dat er zo uitzien: `TileType[] tileTypes = { boardTileType, wallTileType };`)
3. Je tilemap is een tweedimensionale array waarin je vastlegt hoe de wereld eruit ziet. Een -1 betekent dat er geen tile moet zijn op die positie, een 0 geeft aan dat er een tile moet staan van het type dat op positie 0 van de `tileTypes`-array staat (in het voorbeeld hierboven dus een `boardTileType`), et cetera.

Tiles waar je niet doorheen kunt

Voeg code zoals deze toe aan de methode `tileCollisionOccurred` in het object dat niet door de tile heen mag:

```
if (ct.theTile instanceof WoodenWall ||
    ct.theTile instanceof BrickWall)
{
    if (ct.collisionSide == ct.TOP)
    {
        vector = tilemap.getTilePixelLocation(ct.theTile);
        setY(vector.y - height);
        allowedToJump = true;
    }
    if (ct.collisionSide == ct.LEFT)
    {
        vector = tilemap.getTilePixelLocation(ct.theTile);
        setX(vector.x - width);
    }
    if (ct.collisionSide == ct.RIGHT)
    {
        vector = tilemap.getTilePixelLocation(ct.theTile);
        setX(vector.x + tilemap.getTileSize());
    }
}
```

Hulp bij het tekenen van je tilemap

Aangezien alle tiles in je game even groot moeten zijn, kan het bij een grote wereld veel werk zijn om handmatig een tilemap te maken. Gelukkig ben je niet de eerste met dit probleem en heeft iemand er al een handige tool voor gemaakt: op www.mapeditor.org kun je het programma Tiled downloaden. Hiermee kun je een nieuwe tilemap maken, zelf tiles

toevoegen (bij “Tilesets”, rechtsonder in het scherm), een kaart tekenen en deze exporteren. Als je een export maakt naar een CSV-bestand, hoef je alleen nog maar wat accolades toe te voegen en kun je het resultaat zo kopiëren naar je game.