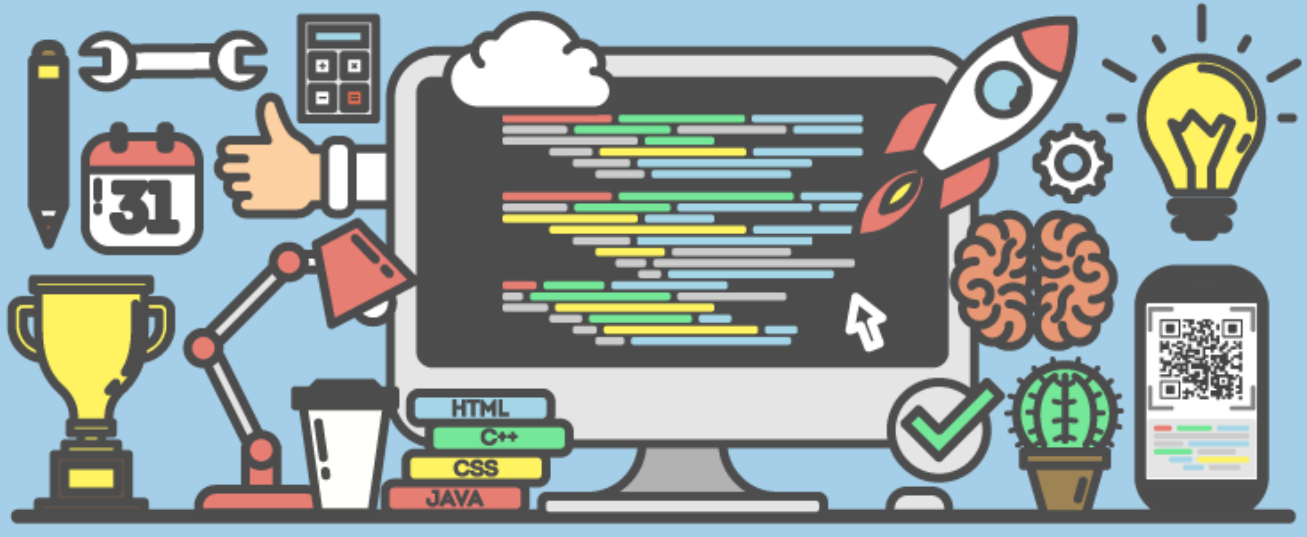
A TopDown Epic Adventure (with a Z)



Gaëtan Blaise-Cazalet - Head of programming section

Our hero is a sprite

The Sprite class

- In the previous games, we used always the same pattern for our game object classes :

We had some position and transformation variable : x, y, rotation, ox, oy...

We had some functions : Load, Update, Draw

- This is called boilerplate code. Code we always have to recreate.

- We would like to encapsulate this code into a class we will reuse through our next games. This class will be called the Sprite class.

The Sprite class

```
class Sprite
{
    public float X { get; set; }
    public float Y { get; set; }
    public float Ox { get; set; }
    public float Oy { get; set; }
    public float Rotation { get; set; }
    public bool Visible { get; set; }
    public Color Color { get; set; }

    Texture2D image;
    string path;

    public Vector2 Position
    {
        get
        {
            return new Vector2(X, Y);
        }
        set
        {
            X = value.X;
            Y = value.Y;
        }
    }

    public Rectangle Rect
    {
        get
        {
            return new Rectangle((int)(X + Ox), (int)(Y + Oy), image.Width, image.Height);
        }
    }

    public byte Opacity
    {
        get
        {
            return Color.A;
        }
        set
        {
            Color = new Color(Color.R, Color.G, Color.B, value);
        }
    }
...
```

The Sprite class

```
...
    public Sprite(int x, int y, string path)
    {
        X = x;
        Y = y;
        this.path = path;
        Color = Color.White;
    }

    public virtual void Load(ContentManager content)
    {
        image = content.Load<Texture2D>(path);
    }

    public void Draw(GameTime gameTime, SpriteBatch spriteBatch)
    {
        if (Visible)
        {
            Rectangle rect = new Rectangle(0, 0, image.Width, image.Height);
            spriteBatch.Draw(image, rect, null, Color, Rotation, new Vector2(Ox, Oy), SpriteEffects.None, 0);
        }
    }
}
```

The game objects classes

- From our sprite class, we can create different game object classes. They will inherit from Sprite.
- In our current game, we will want a Hero, a Projectile and an Enemy. They will all inherit from the Sprite class
- So create three Hero, Projectile and Enemy classes.

```
class Hero : Sprite
{
    public Hero(int x, int y, string path) : base(x, y, path)
    {

    }
}
```

```
class Projectile : Sprite
{
    public Projectile(int x, int y, string path) : base(x, y, path)
    {

    }
}
```

```
class Enemy : Sprite
{
    public Enemy(int x, int y, string path) : base(x, y, path)
    {

    }
}
```

Hero

Hero move

Make the Hero move up/down/left/right when pressing ZQSD.

We want a smooth move, like in the lander lesson.

Hero rotation

Rotate the hero the right direction. You have two choices :

- Use the Rotation member of the Sprite class
- Create 4 textures in the Hero and display one in function of the direction

Projectiles