Documentation Technique

Projet C# – Jeu *Doodle Jump*

Auteur : Etienne Caulier

Date : 28 octobre 2025

CFPT Informatique

Table des matières

1 Arborescence

2 Models

Listing 1 – code pour un element MonoGame

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DJGame.Interfaces;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Graphics;
using SharpDX.Direct2D1.Effects;

namespace DJGame.Models
{
    public abstract class UiElement : IMonogameElement
    {
        // Champs de la classe...
        protected Texture2D texture;
        protected Vector2 position;
        protected Vector2 velocity;
        private float scale;
        public bool flipped;
        private float rotation;
        private bool showHitbox;

        // Propri t s de la classe...
        public Vector2 Position { get => position; }
        protected Texture2D Texture { get => texture; }
        public Vector2 Velocity { get => velocity; }
        public float Scale { get => scale; }
        public bool Flipped { get => flipped; }
        public float Rotation { get => rotation; }
        public bool ShowHitbox { get => showHitbox; }


        // Constructeur de la classe...
        public UiElement(Vector2 position, Vector2 velocity, int sizePourcent = 100, bool
            flipped = false, float rotation = 0, bool showHitbox = false)
        {
            this.position = position;
            this.velocity = velocity;
            SetSize(sizePourcent);
            this.flipped = flipped;
            this.rotation = rotation;
            this.showHitbox = showHitbox;
            // this.showHitbox = true;
        }

        // M thodes de la classe...
        public abstract Rectangle Hitbox();

        public void Remove()
        {
            SetSize(0);
        }

        protected void SetSize(int pourcent)
        {
            scale = pourcent / 100f;
        }

        protected void SetRotation(int degrees)
        {
            rotation = MathHelper.ToRadians(degrees);
```

```
63          }
64
65          public abstract void LoadContent(ContentManager content);
66          public abstract void Update(GameTime gameTime);
67          public abstract void Draw(SpriteBatch spriteBatch, GameTime gameTime);
68      }
69  }
```

2.1 Agents

2.2 Controls

2.3 Game

2.4 Windows