Exercice – Mappage de texture

A screenshot of a computer game

AI-generated content may be incorrect.

Résultat final de l’exercice !

## GameObjectImage – mode débogue

Faites en sorte que lorsque le mode débogue de l’engine est actif, une boite est dessinée pour la boite de collision à l’aide de :

DrawRectangleLinesEx(...)

Assurez-vous de dessiner après l’image pour le voir par-dessus. Assurez-vous d’utiliser cette méthode pour afficher le rectangle !

GetCollisionAtPosition()

**Trucs**

Aidez-vous de ce test pour savoir si le mode débogue est à actif ou non.

GetIEngine().GetDebugMode()

Vous devriez être en mesure d’activer et désactiver le mode débogue en appuyant sur la touche **p**.

## GameObjectImage – pivot au centre

Modifier Draw2D afin qu’elle utilise

DrawTexturePro(…)

et faites en sorte que le pivot soit au centre au lieu d’en haut à gauche.

## GameObjectImage – Réparer la collision

Vous remarquerez que l’image ne correspond plus à la boite de collision. Trouvez une solution pour réparer cela. Dans GameObject2D, rendez cette méthode virtuelle.

const Rectangle& GetCollisionAtPosition() const;

Modifier l’implémentation dans GameObjectImage pour qu’elle compense pour son pivot. Attention, utilisez la largeur et la hauteur de la boite de collision et non la texture.

## GameObjectImage – méthode changement du Draw2D

Dorénavant, dans la méthode Draw2D…

* La grosseur du quad est celle de la boite de collision multipliée par GetScale() .
* Ajustez le calcul du pivot en conséquence si applicable.

Autrement, dit, la boite de collisions sera la grosseur de notre objet par défaut et on peut faire une homothétie (*scaling*) de notre image autour !

## Player (CharacterControler) - Rotation

Faites en sorte que votre joueur fait face dans la direction dans laquelle vous appuyez. Inspirez-vous de ce code.

//Les images des personnages devront faire face à droite par défaut

constexpr Vector2 \_ImageFacingDirection = { 1.f,0.f };

//Angle entre la direction du contrôle et la direction de l'image.

const float angleRadian = Vector2Angle(\_ImageFacingDirection,controlDirection);

//Converti en degré et défini la rotation

SetRotation(RAD2DEG \* angleRadian);

## GameObjectTile - Nouvelle classe

Dans le fichier *GameObject.h*, ajouter

class GameObjectTile : public GameObjectSingleImage

{

public:

GameObjectTile(const std::string& InImageFileName, const Rectangle& InSceneRectangle);

virtual void Draw2D() override;

};

Pour le moment, copiez-coller la méthode Draw2D originale mais faites que la texture s’affiche en méthode tuile.