

spyrOpenGL

LACHAUD Samuel / PAZOLA Loïs – Info S5 TP4







SOMMAIRE

2/10

Afin de mener à bien notre projet, nous avons utilisé des ressources officielles de Spyro, le projet étant privé et non-commercial, cela ne devrait pas poser de problèmes. Nous avons donc utilisé les ressources suivantes :

* Fichiers textures et obj
  + Fichier texture provenant du jeu Spyro 2 sur PS1 (Officiel)
  + Fichier obj provenant du jeu Spyro 2 sur PS1 (Officiel)
  + [Lien vers le site 🡪](https://www.models-resource.com/playstation/spyro2riptosrage/model/26411/)
* Fichiers de musique
  + Dark Hollow (Spyro The Dragon Reignited - 1)
  + Enchanted Towers (Spyro The Dragon Reignited - 3)
  + Sgt. Byrd's Theme (Spyro The Dragon Reignited - 3)
  + [Lien vers le site 🡪](https://www.youtube.com/watch?v=L7Rqf4s8KQ0&list=PLo6SSHo84XM6OOrVDvXmms5jdiOeyC0wg)
* Fichier audio
  + « Salut moi c’est Spyro » d’une cinématique (Spyro The Dragon Reignited - 2)
  + [Lien vers le site 🡪](https://www.youtube.com/watch?v=ovA1417TuOM&t=75s)

Bien sur le fichier OBJ à uniquement été utilisé lors de l’étape de conception du projet comme modèle.

Le sujet étant de réaliser un dragon en 3D avec OpenGL, nous avons donc cherché les différents dragons iconiques des jeux vidéo. Parmi ceux qui sont ressortis de ce brainstorming, il y avait Spyro. Mais alors pourquoi choisir Spyro plutôt qu’un autre dragon ?

La trilogie des Spyro sur PS1 fut marquante pour beaucoup de joueurs car possédant un univers riche, de très beaux graphismes (pour l’époque, années 1998-2000) mais surtout un personnage « classe » et « charismatique ». Spyro avait du style et on pouvait facilement s’identifier à lui, ce qui en a fait une figure pour beaucoup comme a pu l’être Sonic pour d’autres.

Utilisation des ressources de Spyro

Pourquoi Spyro ?

Le projet SPYROpenGL est le projet demandé dans la matière Synthèse d’Image. Ce nom est la concaténation de SPYRO, notre dragon modélisé, et d’OpenGL, le langage (avec c++) pour modéliser ce personnage. Dans ce document, nous allons voir pourquoi nous avons choisis ce sujet, les étapes de conceptions de celui-ci ainsi que quelques informations supplémentaires.

1. Introduction

3/10

Une image contenant texte, tableau blanc

Description générée automatiquement

4/10

**Voici un exemple pour la réalisation d’un pied de Spyro :**

Afin de réaliser le pied de Spyro, je le découpe en trois parties :

* Les doigts de pieds unis en un **prisme triangulaire rectangle**
* Le centre du pied en un **cube**
* Le talon/attache de la jambe en un **prisme triangulaire équilatéral**

Armé de Blender avec notre fichier OBJ et d’un grand tableau blanc, on découpe Spyro en 11 parties : que nous allons ensuite découper en solides

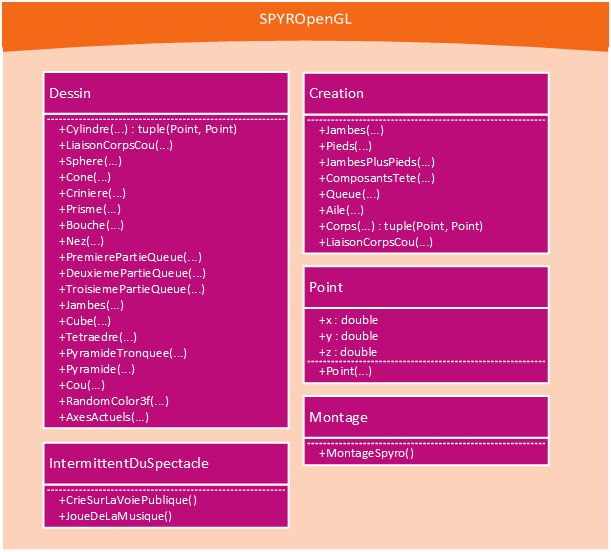
* Les ailes
* La bouche
* Les cornes
* Le corps
* Le cou
* La crinière
* Les jambes
* Le nez
* Les pieds
* La queue
* La tête

Sur chaque partie, on prend notre tableau blanc et on cherche les solides ou ensembles de solides basiques permettant de réaliser celle-ci.

Découpage de Spyro

Le fichier OBJ présentant Spyro avec des formes trop complexes pour nous, on a donc divisé Spyro en plusieurs parties et avec ce fichier, on a décidé des différents solides à utiliser pour monter notre propre Spyro. Puis on a ensuite divisé la création en OpenGL de ce personnage en différentes classes afin de commencer sur des bases solides.

1. Conception



Afin de ne pas commencer la programmation à l’aveugle, nous effectuons un diagramme de classes, avec les différentes méthodes nécessaires à la bonne réalisation du projet.

Chacune des classes de ce diagramme est une classe statique (sauf la classe Point). Les classes sont expliquées à la suite du diagramme UML.

5/10

Diagramme de classe



6/10

Répartition des tâches

Afin de ne pas effectuer les mêmes tâches dans le projet, nous avons divisés les taches les plus faciles et gardé en commun les taches les plus complexes. Voici donc la répartition des tâches :

Samuel LACHAUD :

* Jambes, Pieds, Ailes, Tête, Cornes, Crinière.

Loïs PAZOLA :

* Queue, Corps, Cou, Bouche, Nez.

Taches Communes :

* Application des texture, Animation, mise en place des sons, Lumières

1. Classe Point :

Cette classe est une classe très sommaire car permettant de créer des points à l’aide des coordonnées X, Y et Z. Elle est utilisée lorsqu’il y a besoin de retourner un point ou de passer un point en paramètre au sein d’une méthode.

1. Classe Dessin :

Cette classe est la classe de plus bas niveau. En effet celle-ci gère la création de figures géométriques en 3D telles que la sphère ou le cube. C’est une classe statique, elle est donc appelable directement depuis n’importe où sans avoir à l’instancier.

1. Classe Création :

Cette classe est la classe de création des différentes parties du corps de Spyro. En effet elle utilise les méthodes statiques de la classe Dessin afin de monter à l’aide de translations et rotations les parties du corps. Cette classe est également statique.

1. Classe Montage :

Cette classe est la classe du plus haut niveau. En effet elle assemble à l’aide de translations et rotations toutes les méthodes statiques de la classe Création. Cette classe est également statique.

1. Classe IntermittentDuSpectacle :

L’intermittent du spectacle permet de jouer de la musique en arrière-plan, parmi 3 musiques différentes aléatoirement, et ce sans jamais jouer la même d'affilé (quel talent !). Il crie également « Salut, Moi c’est Spyro » lors de l’appuis de la touche ESPACE.



1. Réalisation

7/10



6 / 9



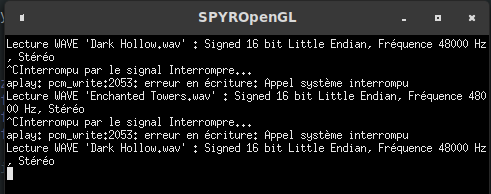


8/10





9/10



En bonus, dans SPYROpenGL, de la musique est jouée en arrière-plan, celle-ci est jouée dans un thread appelant des commandes Linux. Les trois musiques sont les suivantes :

Dark Hollow, Enchanted Towers, Sgt. Byrd's Theme

Elles sont donc jouées en arrière-plan. Il y’a quelques subtilités, quand une musique s’arrête, une autre prend le relais, mais jamais la même d’affilé, évitant donc la redondance pour l’utilisateur. On peut également passer à la musique suivante en appuyant sur ‘CTRL+C’ sur la deuxième fenêtre ouverte lors de l’exécution nous avons donc l’affichage suivant permettant de savoir les informations de la musique jouée actuellement :

Changement de musique de fond

De nombreuses touches du clavier permettent d’utiliser facilement SPYROpenGL, voici la liste des commandes permettant la navigation ou l’utilisation de fonctionnalités :

* 'z’ 🡪 Permet de zoomer sur Spyro
* ‘Z’ 🡪 Permet de dézoomer sur Spyro
* ‘i’ et ‘k’ 🡪 Permettent de déplacer Spyro sur l’axe Y
* ‘j’ et ‘l’ 🡪 Permettent de déplacer Spyro sur l’axe X
* ‘J’ et ‘L’ 🡪 Permettent de déplacer Spyro sur l’axe Z
* ‘q’ 🡪 Permet de quitter l’application
* ‘ESPACE’ 🡪 Permet de faire bouger la bouche de Spyro avec le son de sa voix
* Les touches fléchées 🡪 Permettent de rotationner la caméra autour de Spyro

Utilisation du clavier

Maintenant que nous avons vu la réalisation du projet et sa finition, nous allons voir les fonctionnalités associées, plus particulièrement les touches claviers utilisées et la gestion de la musique, ceci nous permettant une utilisation plus personnalisée du projet SPYROpenGL.

1. Utilisation du projet

10/10