



SIRIUS ENGINE

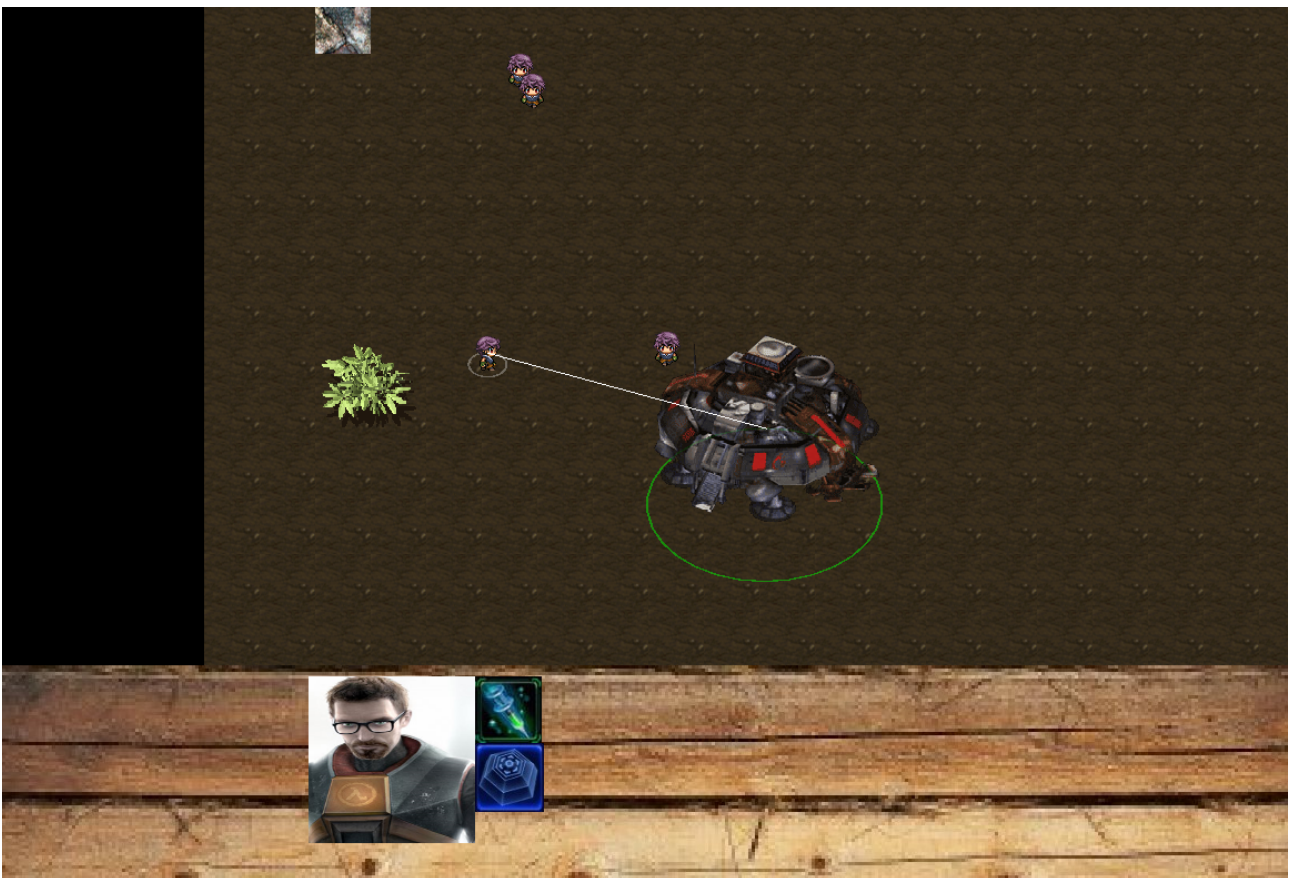
Create your own 2D Game...

PROJET OPEN SOURCE

@ Magras Steve, 201

Présentation du projet

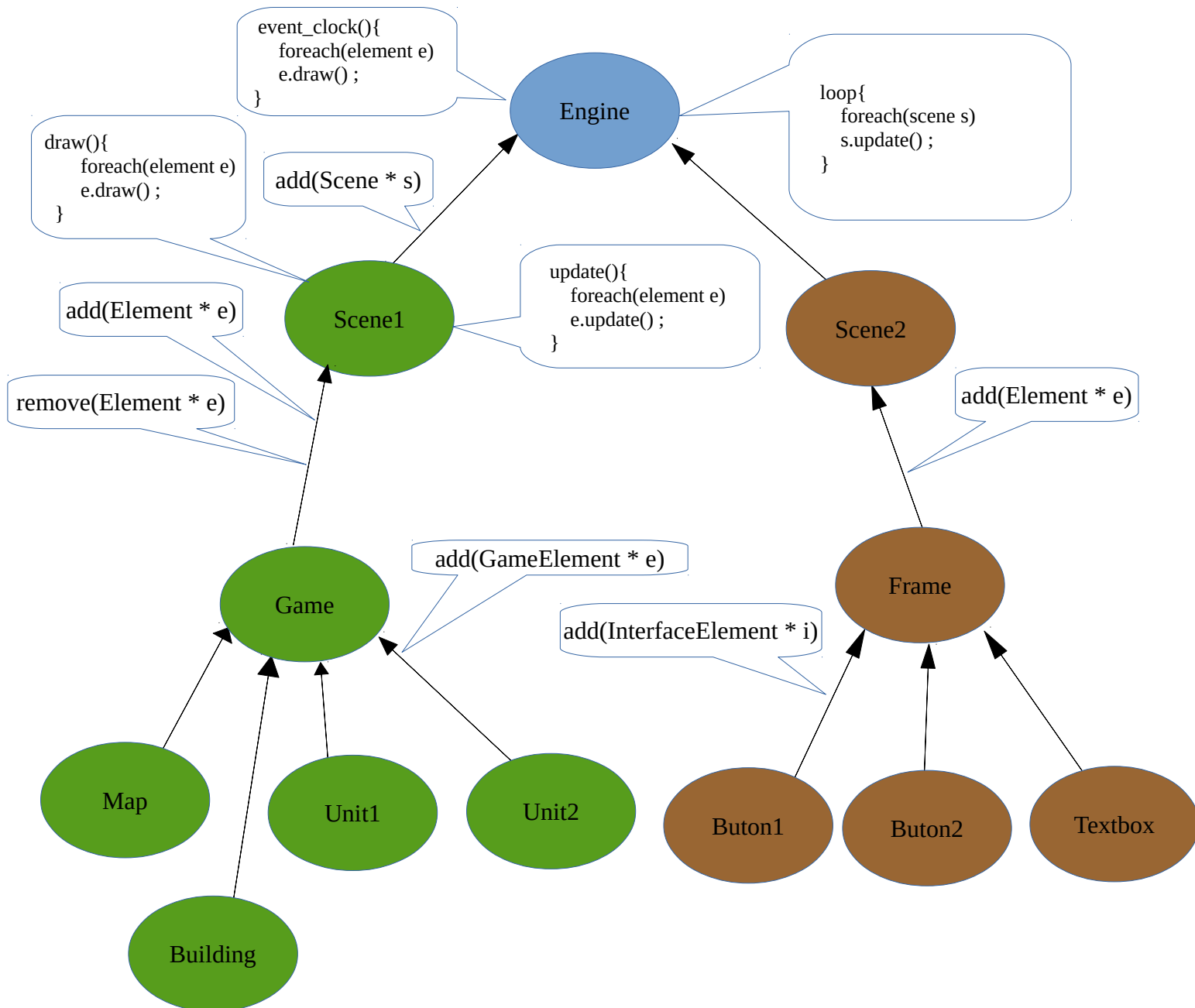
Ce projet a pour vocation de créer une librairie accessible à tous dans le but de faciliter la création de jeux 2D. La bibliothèque contiendra donc une encapsulation de plusieurs classes de haut niveau qui permettront de gérer tout ce qui est générique à ce genre de jeux. Parmi les différentes choses nous retrouverons la gestion de l'affichage de la carte, des unités, de leurs déplacements (collisions, pathfinding) de leurs interactions avec l'utilisateur et avec les autres entités, de la gestion des événements, de l'interface, du son, et de la couche réseaux, bref tous les services nécessaires au bon fonctionnement de ce genre de programme.



Technologies utilisées

Ce projet sera basé sur le langage c++ et de l'api SFML 2.1 pour tout ce qui est gestion des médias (image, son, réseaux). En ce qui concerne l'EDI, l'application Netbeans sera préconisée. Le versionning sera géré avec GitHub pour assurer une mise en commun des travaux de chacun d'une manière saine et organisée.

Fonctionnement de la bibliothèque



Comme le montre le schéma ci-dessus, la bibliothèque SiriusEngine utilise deux fonctions de manière systématique :

- La méthode add : qui permet d'ajouter un élément à la collection d'objets traités par l'élément qui l'appelle.
- La méthode remove : qui permet de supprimer un élément de la collection d'objets traités.

- La méthode draw : qui affiche a l'écran les objet de la collection. Elle est appeler par un timer définie par l'utilisateur.
- La méthode update : qui met a jours les données des objet de la collection a chaque tour de boucle.

Architecture

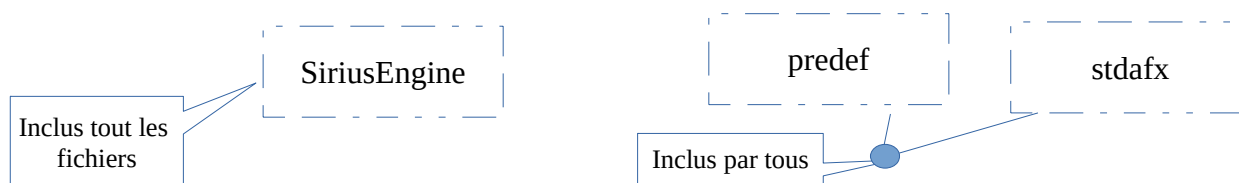
Vue générale

Lorsqu'un utilisateur de la librairie voudra utiliser le moteur de jeu, il inclura l'entête « SiriusEngine », toutes les fichiers seront donc inclus par son biais.

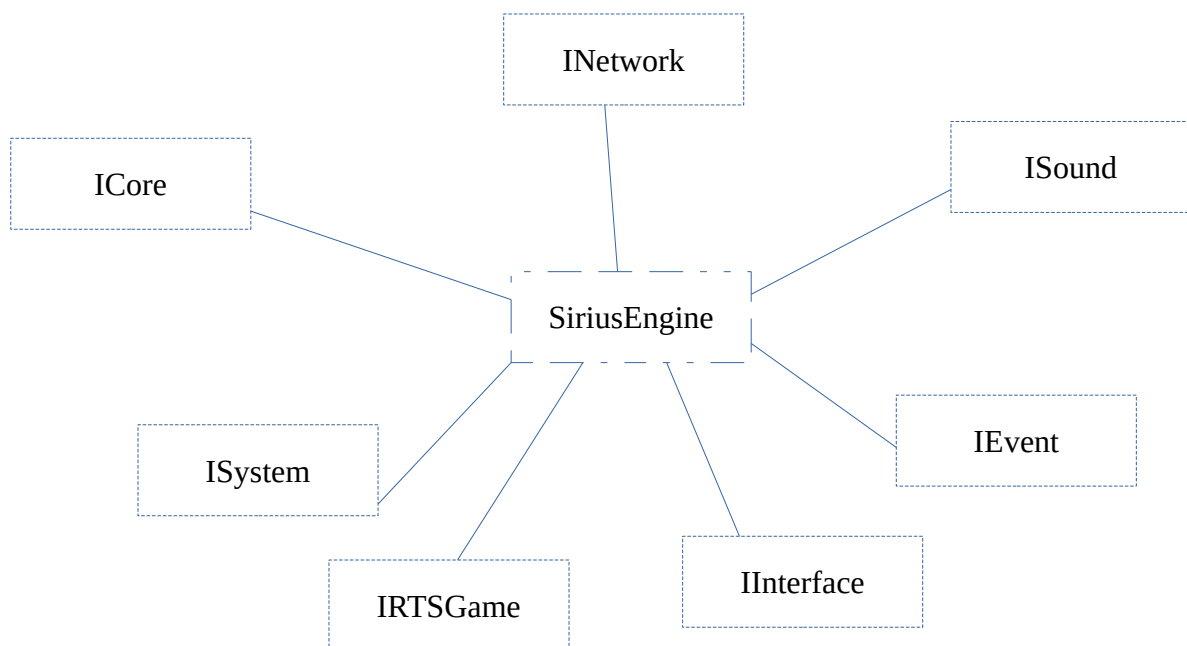
predef.hpp : Il permet d'inclure toutes les pré-définitions des classes.

stdafx.hpp : Il permet d'inclure toutes les classes contenues par le package « System », en effet ce package regroupe les classes basiques qui sont utilisées très souvent (ex : Point, Rectangle).

Chaque classe du projet inclura donc ces deux fichiers pour permettre un accès total à chacune des fonctionnalités totales du projet.



Il est à noter que chaque package possède une classe qui inclut les autres, celle-ci commence par un I majuscule et est suivie par le nom de celui-ci.



Package System

Cet ensemble de classes n'as pour vocation que de fournir un ensemble de structures générique permettant a la bibliothèque et a ses utilisateurs un développement plus rapide et plus sain.

System

Vector
<template>

Rectangle
<template>

Point
<template>

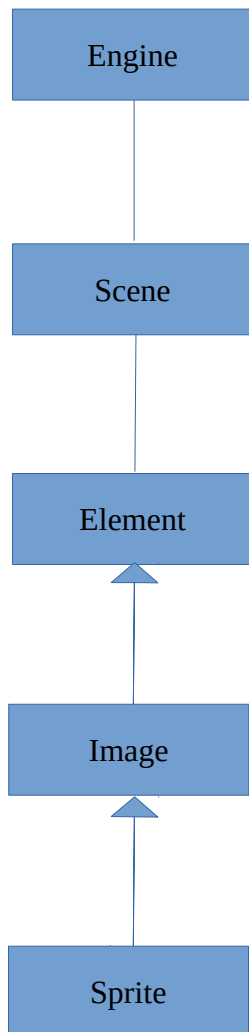
Color

Circle
<template>

Package Core

Cet ensemble regroupe toutes les classes de basique de la bibliothèque. L'Engine est composée de scène qui sont elle même composée d'éléments.

Core



Package RTSGame

