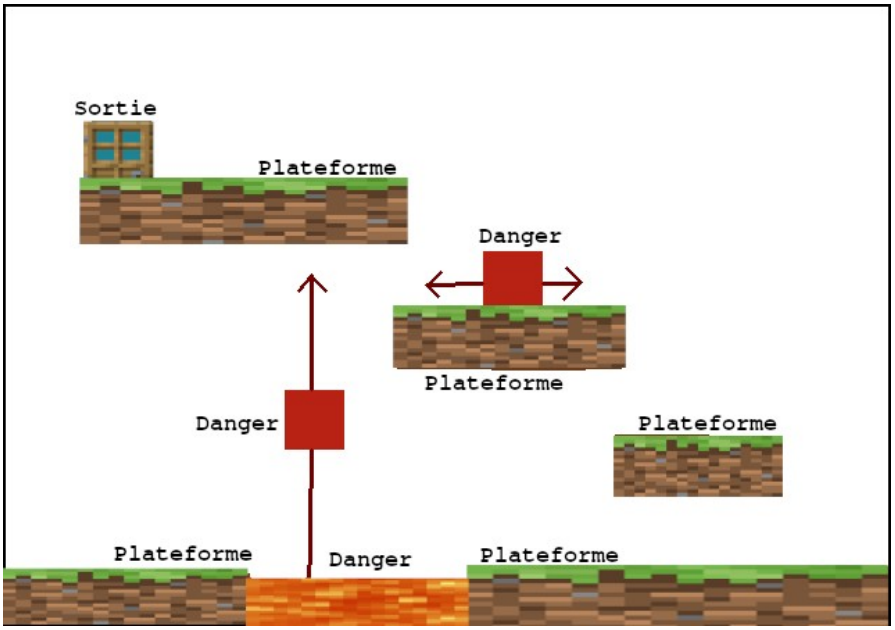
Projet Plateformer

Le but du projet est de développer un jeu de plateforme, ou au moins un niveau d'un jeu de plateforme.

Le principe de base doit donc être de déplacer un personnage, en sautant de plateforme en plateforme, afin d'atteindre un endroit identité comme la sortie. Pour rendre le jeu intéressant, le chemin jusqu'à la sorti est parsemé de dangers mortels.



Modélisation

Element

La classe **Element** doit servir à représenter les différentes composantes du jeu : le personnage, chaque plateforme, la sortie, les pièges/monstres. La classe **Element** est destinée à servir de classe mère pour les classes qui vont modéliser ces différents éléments.

La classe **Element** possède un attribut sprite de type sf ::Sprite et un attribut texture de type sf::Texture, afin de pouvoir à afficher **l'Element** et lui faire subir des transformations, en particulier des déplacements avec les méthodes Sprite ::move() et/ou Sprite::setPosition().

Personnage

Le **Personnage** est contrôlé au clavier, il doit pouvoir aller à gauche, à droite et sauter.

Il doit pouvoir gérer les collisions avec les autres **Elements** du jeu, pour se maintenir sur les plateforme, perdre la partie en cas de collision avec un piège/monstre, gagner en cas de collision avec la sortie. La méthode Rect::intersects() sera très utile pour cela.

Le **Personnage** doit être soumis à la gravité, de manière plus ou moins réaliste. En première approche, on peut estimer qu'il cherche à se déplacer automatiquement d'un pixel vers le bas à chaque frame
De manière plus réaliste, il faudrait utiliser un vecteur vitesse vertical, une constante de gravité universelle, et un valeur de vitesse terminale.

Plateforme

Ce sont des **Elément** immobiles. Leur principale utilité est de servir de chemin au **Personnage**.
Faire une classe différente de **Element** pourrait permettre de faire évoluer le jeu, par exemple en donnant des propriétés à certaines plateforme (par exemple elles pourraient être friables, mobiles, etc.).
Il serait sans doute pratique de pouvoir définir facilement la largeur d'une **Plateforme**.

Sortie

La **Sortie** est un **Element** immobile. Si le **Personnage** entre en contact avec la **Sortie**, il gagne.

Danger

Les **Dangers** sont des **Elements** dont le contact est mortel pour le Personnage. Que ce soit des pointes acérées immobiles, des montres qui patrouillent, des projections de lave, le principe est le même, ce sont des éléments qui peuvent avoir des mouvement oscillants.
Afin de gérer des mouvement oscillants de gauche à droite, un **Danger** possède une vitesse horizontale vx (par exemple 1 pour aller à droite, -1 pour aller à gauche), et des coordonnées xmin et xmax. L'attribut vx change de signe quand le **Danger** sort de l'intervale [xmin ; xmax].
De même, afin de gérer des mouvements oscillants de haut en bas, il possède des attributs vy, ymin et ymax

Jeu

Cette classe doit réunir toutes les objets du jeu, regroupée en vector<> quand ils appartiennent à la même classe, comme les **Plateformes**.
Le constructeur de la classe définit la configuration du niveau.

Apparence

Vous pouvez trouver sur OpenGameArt.org des tilesets et des spritesheets libres de droit

Rendu du projet

Les attributs atomiques (int, double, etc.) de vos classes doivent être encapsulées.
Votre code doit respecter des conventions de nommages cohérentes et être commenté.
Le projet doit être divisé en fichiers .h et .cpp et doit pouvoir être reconstruit grâce à un makefile.
Le projet doit compiler dans les salles de TP, c'est à dire sur Ubuntu avec la SFML 2.5.1
Le projet doit être remis à la dernière séance et envoyé par mail à l'adresse prissette@univ-tln.fr