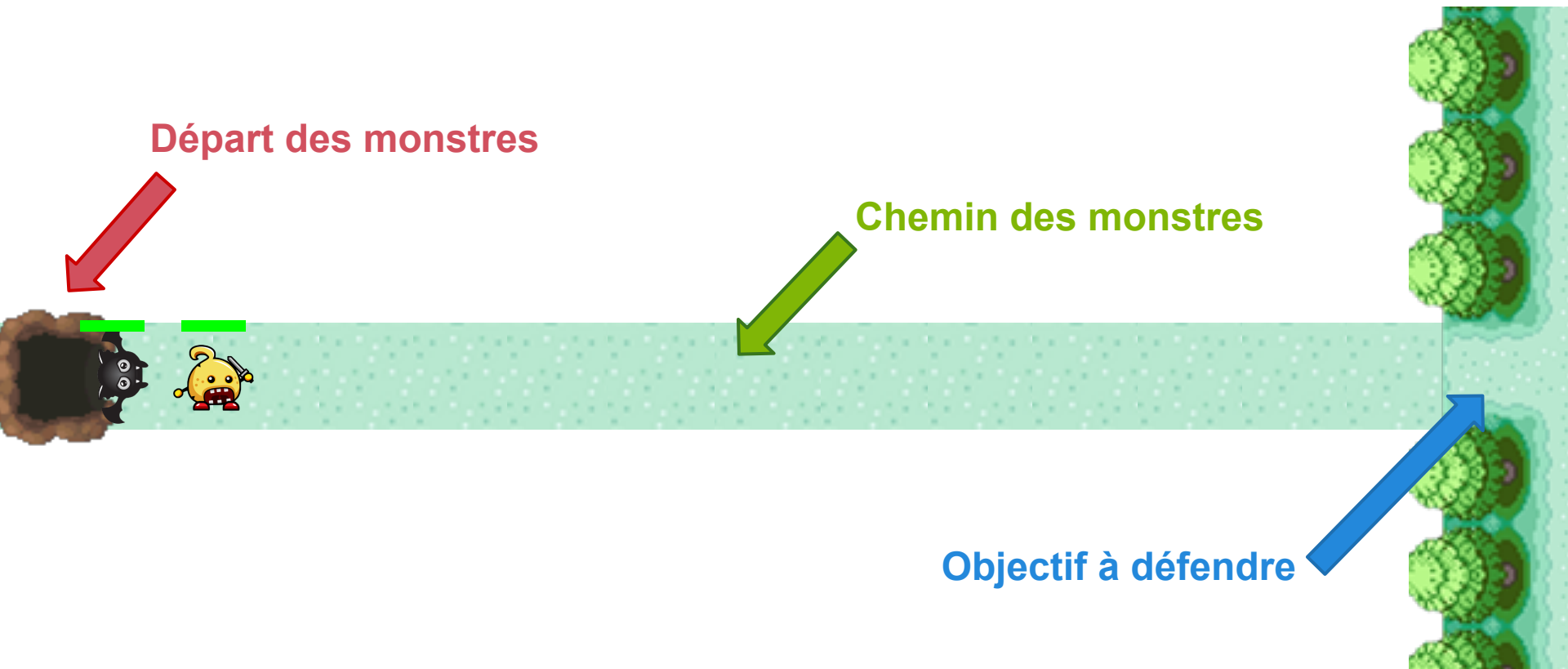


# Tower Défoncé

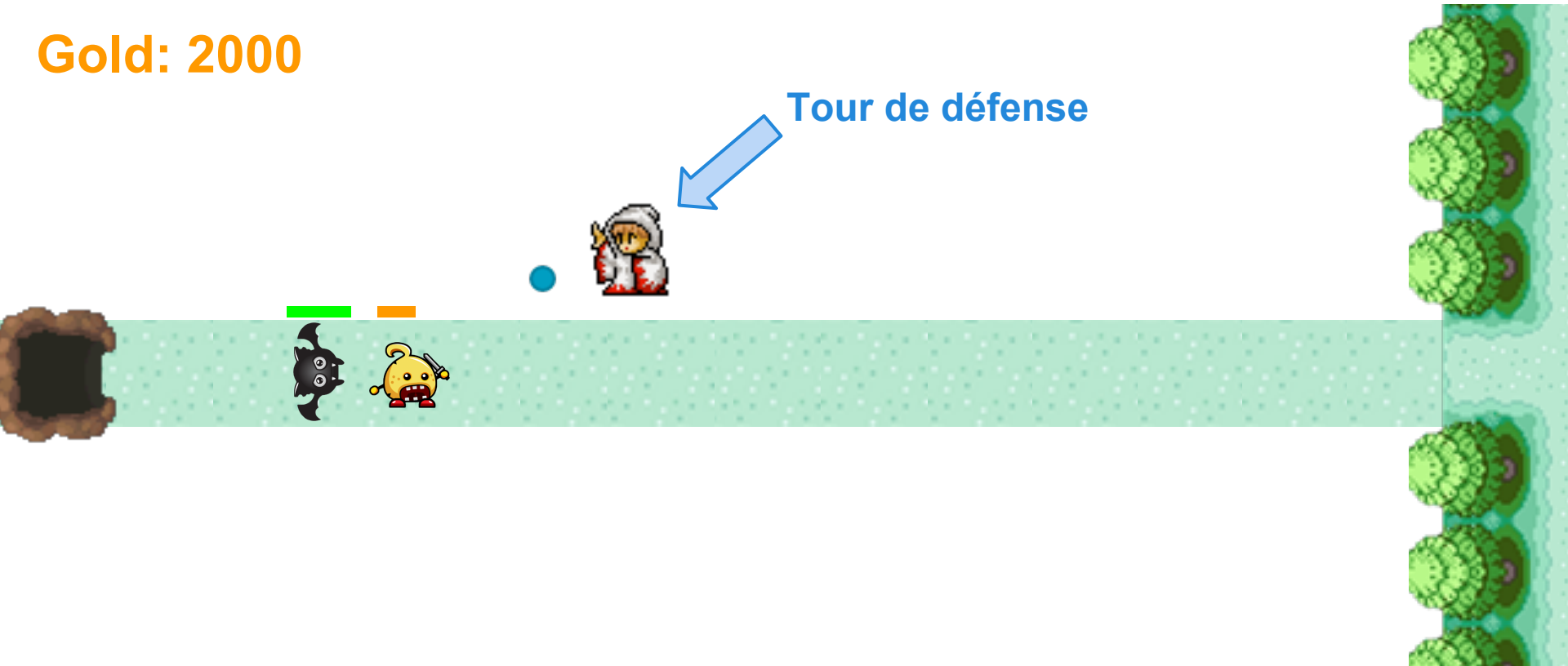
# Quelques explications !



# Quelques explications !

Gold: 2000

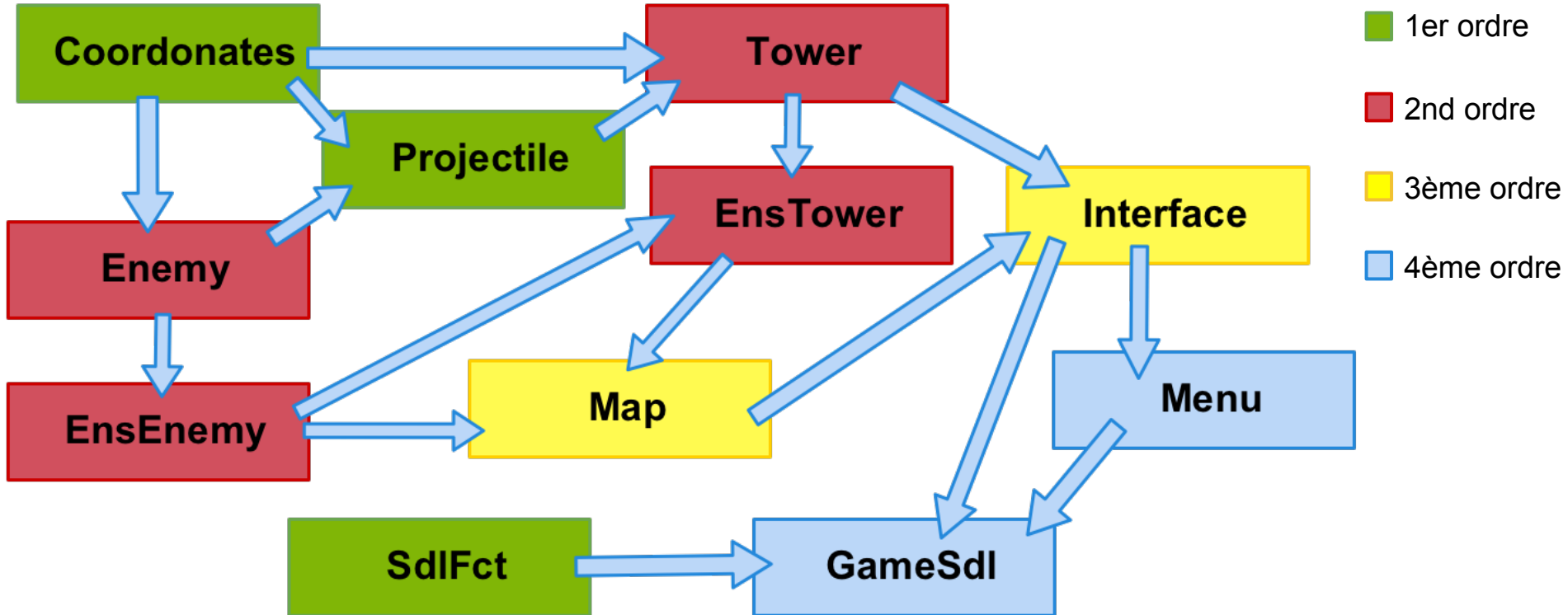
Tour de défense



# Répartition des taches

Guillaume	Tz'ayik	Serge
<ul style="list-style-type: none"><li>- Génération du terrain</li><li>- Définition du chemin (PathFinding)</li><li>- Gestion des Sprites</li><li>- SDL (Affichages / evenements)</li></ul>	<ul style="list-style-type: none"><li>- Système de coordonnées</li><li>- Ennemis</li><li>- Projectiles</li></ul>	<ul style="list-style-type: none"><li>- Tours</li><li>- Interaction Ennemis / Tours</li><li>- Interface</li></ul>

# Organisation des Modules



# Les Modules

# Coordonates

## Simplifie le positionnement des entités

Coord :


X  
Y


```
void multiplyCoord (Coord* v, double coeff);  
Coord* oppositeCoord(const Coord* v);  
double getCoordsDistance(Coord uno, Coord dos);  
void normaliseCoord (Coord* coo);  
Coord multipliedCoord(Coord* v, double coeff);  
int equalCoords(Coord* a, Coord* b);
```


## Opération sur les coordonnées

# Enemys

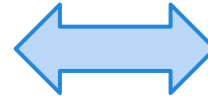
## 3 types différents:

 : Faible, vitesse moyenne, nombreux.

 : Non attaquable par certaines tours, Rapides.

 : Beaucoup de vie, lent, chaque 5 niveaux.

## EnsEnemy:



EnsEnemy contient tous les ennemis d'un niveau.  
C'est le lien entre les ennemis et la map.

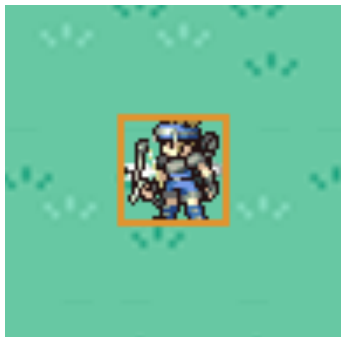


# Towers

Unités principales du jeu. Les tours permettent au joueur de défendre l'objectif en attaquant les ennemis.

Le module s'occupe des informations concernant les tours.

Il y en a 4 types différents de tours (Chevalier, Archer, Mage et Canon)



Ils sont gérés par EnsTower, qui s'occupe de faire le lien entre les tours, la carte et les ennemis. EnsTower a un tableau contenant chacune des tours présent sur la carte et pourra ainsi acheter leurs améliorations et traiter les interactions avec les ennemis et les informations concernant le joueur

# Towers

Une tour est définie par ses:

- identifiant
- niveau
- type
- son coût
- sa zone d'attaque
- sa vitesse d'attaque
- sa puissance d'attaque
- sa position



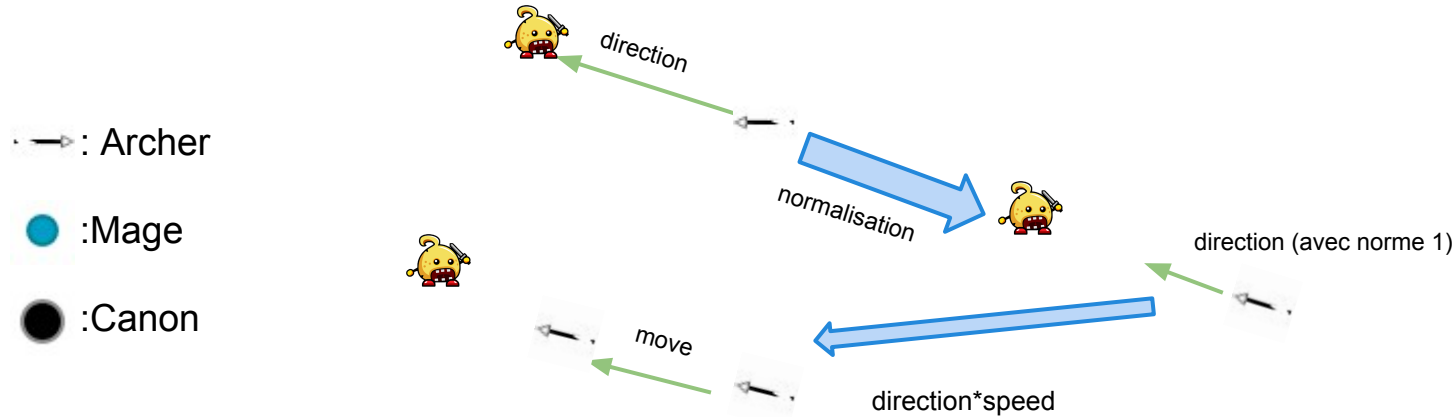
```
void setTowerArcher (Tower *t)
{
    setTowerId(t, -1);

    setTowerLvl(t,1);|
    setTowerType(t, 'A');

    setTowerCost(t,300);

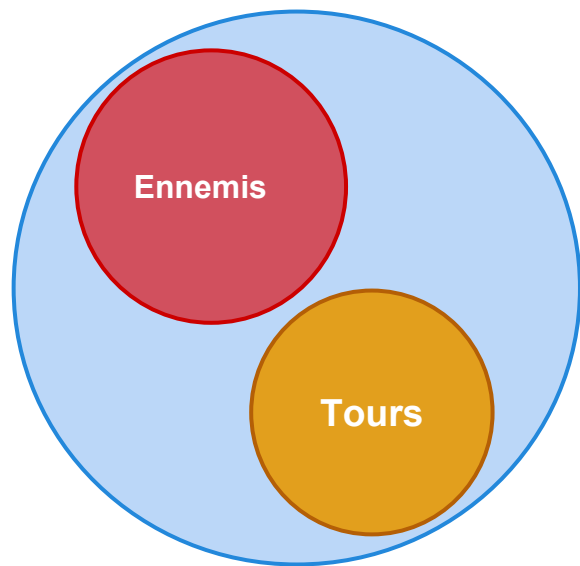
    setTowerRange(t ,2 , 3.6);
    setTowerDefaultTimer(t,20);
    setTowerDmg(t,80);
}
```

# Projectiles

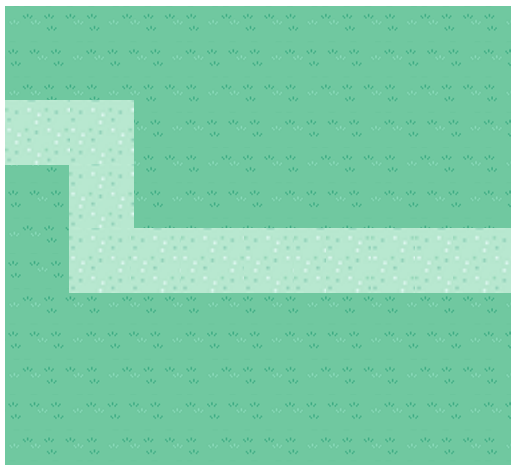


**Purement esthétiques. Un projectile par tour.**

# Map



**Sous Ensembles**



**Terrain**

**Vie:** 1000

**Or:** 1337

**Numéro de vague:**  
42

...

**Informations**

# Map - Initialisation

[illegible]

Cell1	Cell2	Cell3
Cell4	...	...
...	...	...

Constructible: true

**Cellule du chemin:**  
**false**

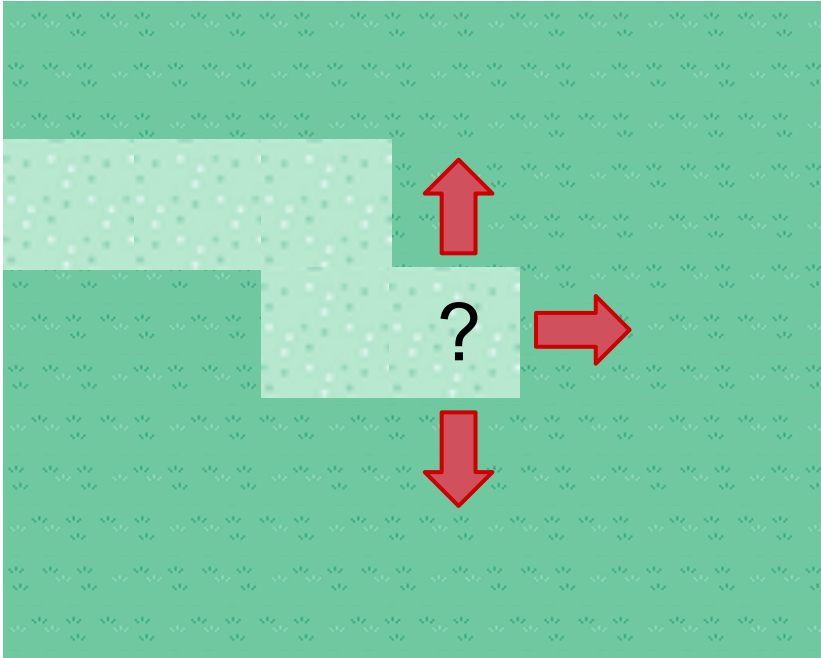
**Debut / fin de chemin:**  
**false**

**Position:** x / y

## Définition d'un niveau

## Cellule

# Map - Initialisation



1 / 1	1 / 2	1 / 3	2 / 3	2 / 4
-------	-------	-------	-------	-------

**Path Finding**

# Interface



Curseur

```
+void moveCursorLeft(Cursor *c) { ... }  
-void moveCursorRight(Cursor *c)  
  {  
    if (c->position.x < c->max.x)  
    {  
      c->position.x++;  
    }  
  }  
+void moveCursorUp(Cursor *c) { ... }  
+void moveCursorDown(Cursor *c) { ... }  
+int isCursorOnTower(Cursor *c, Map *map) { ... }
```

Interaction Joueur / Tour via le curseur

# Interface

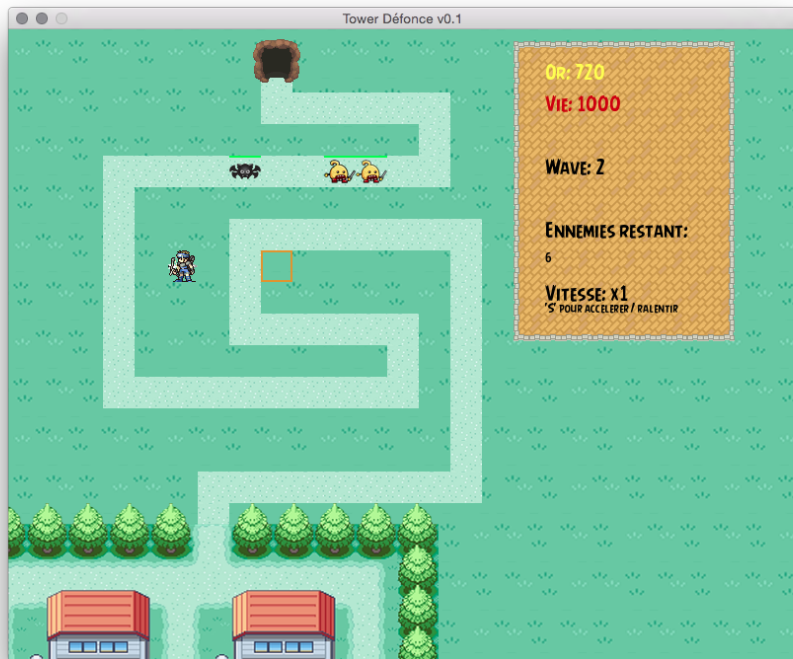


```
void setInterfaceTower(Interface *i, Tower *t);  
  
Tower *getInterfaceTower(const Interface *i);  
  
int buyInterfaceTower(unsigned char type, Coord *c, Map *map);  
  
int sellInterfaceTower(Map *map, Coord *c, Tower *t);  
  
void placeInterfaceTower(Tower *t);  
  
int isCursorOnTower(Cursor *c, Map *map);  
  
Tower *getTowerUnderCursor(Cursor *c, Map *map);
```

**Interaction Joueur / Tour via le curseur**



# Interface



# sdlFct

```
void sdlApplySurface(SDL_Surface* source, SDL_Surface* destination, doubl  
void sdlApplyText(SDL_Surface *destination, double x, double y, char text  
SDL_Surface *sdlLoadImage(const char* filename);  
SDL_Surface *textToSurface(char txt[255], SDL_Color color, TTF_Font *poli
```

**Fonction d'affichage**

**Tableau de surface**

**Nombre d'image: 7**

**Sprite actuel: 3**

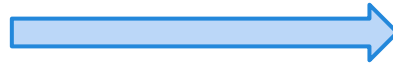
**Timer: 15**

**Sprites**

# sdlFct - Sprites



**Sprite actuel**



**Timer + 1**



**Sprite actuel**

# Menu + GameSdl

## Boucle principale du Jeu

### Boucle du Menu

Évènements

Affichages

Évènements

Mise à jour des éléments

Affichages



Menu

# Merci de votre attention !

