

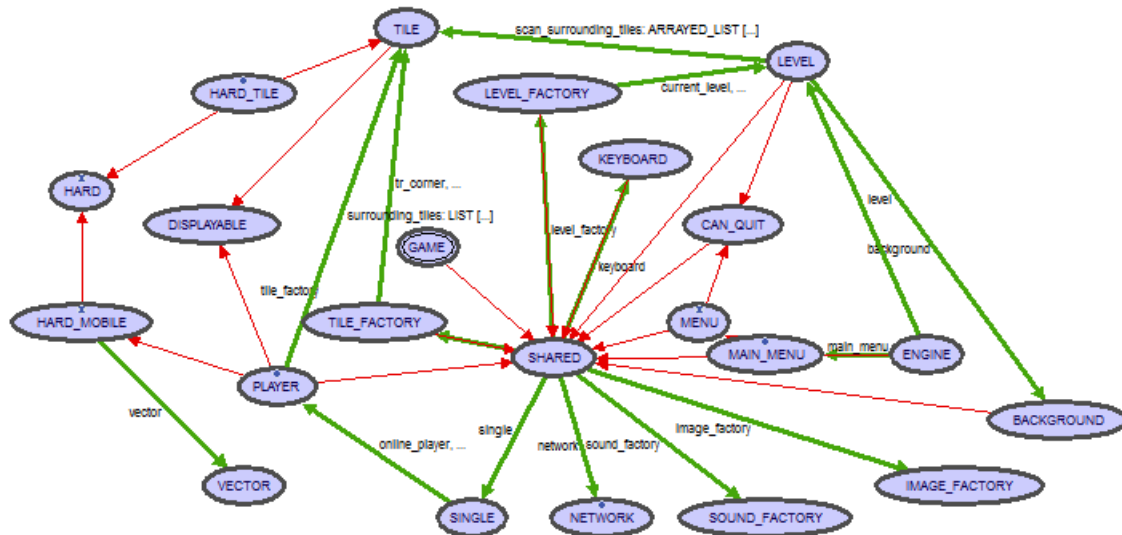
Utilisation

- Touche droit, gauche et barre d'espace
- Lancer le serveur et attendre la connexion du client. (L'application ne répond plus lors de l'attente)
- Connecter le client

Évolution

- Création du niveau avec fichier texte
- Création du joueur
- Déplacement
- Gestion de collision
- Physique
- Son (pas de musique)
- Réseau

Diagramme



Autocritique

J'aurai aimé avoir moins de cours afin de pouvoir mettre plus de temps sur celui-ci. Je ne suis pas satisfait du résultat , mais au moins j'ai été capable d'atteindre la plus part des critères. Je ne comprend pas comment utilisé les contrats et encore une fois je ne comprend rien au jeu de test. Je suis un peu perfectionniste, donc j'essaie que mes classe sois le mieux structuré possible, mais j'ai manqué de temps pour les refaire. Je trouve que ma classe LEVEL fait beaucoup trop de chose. J'ai apprécié la facilité du langage malgré le manque d'exemple sur le web.

Example

Constructeur

make

```
-- Initiaize Engine class
```

```
do
    create_surface.make_with_bit_per_pixel (1024, 768, 16, true)
    controller.create_screen_surface (surface.width, surface.height, 16, true, true, false, true, false)
end
```

Héritage simple

```
class
    ENGINE
inherit
    SHARED
```

Héritage multiple

```
class
    PLAYER
inherit
    ANIMATED
    DISPLAYABLE
    HARD_MOBILE
    SHARED
```

Polymorphisme

```
if attached {HARD_TILE} la_surrounding_tiles.item as la_tile then
    if la_tile.check_collision (current) then
        if y then
            la_tile.apply_collision_y (current)
        else
            la_tile.apply_collision_x (current)
        end
    end
end
end
```

Classe abstraite

```
deferred class
    MENU
```