

Comportement global

Lorsqu'une manche est en cours:

- ↳ L'interface ne doit pas permettre de détruire ou placer des tours (pas les sélectionner dans l'inventaire ?)
- ↳ Le timer de jeu est en marche et déclenche les ticks, jusqu'à la fin de la manche

Lorsqu'aucune manche n'est en cours:

- ↳ Le joueur peut acheter/supprimer des tours
- ↳ Le bouton ATTAQUE permet de lancer la prochaine manche

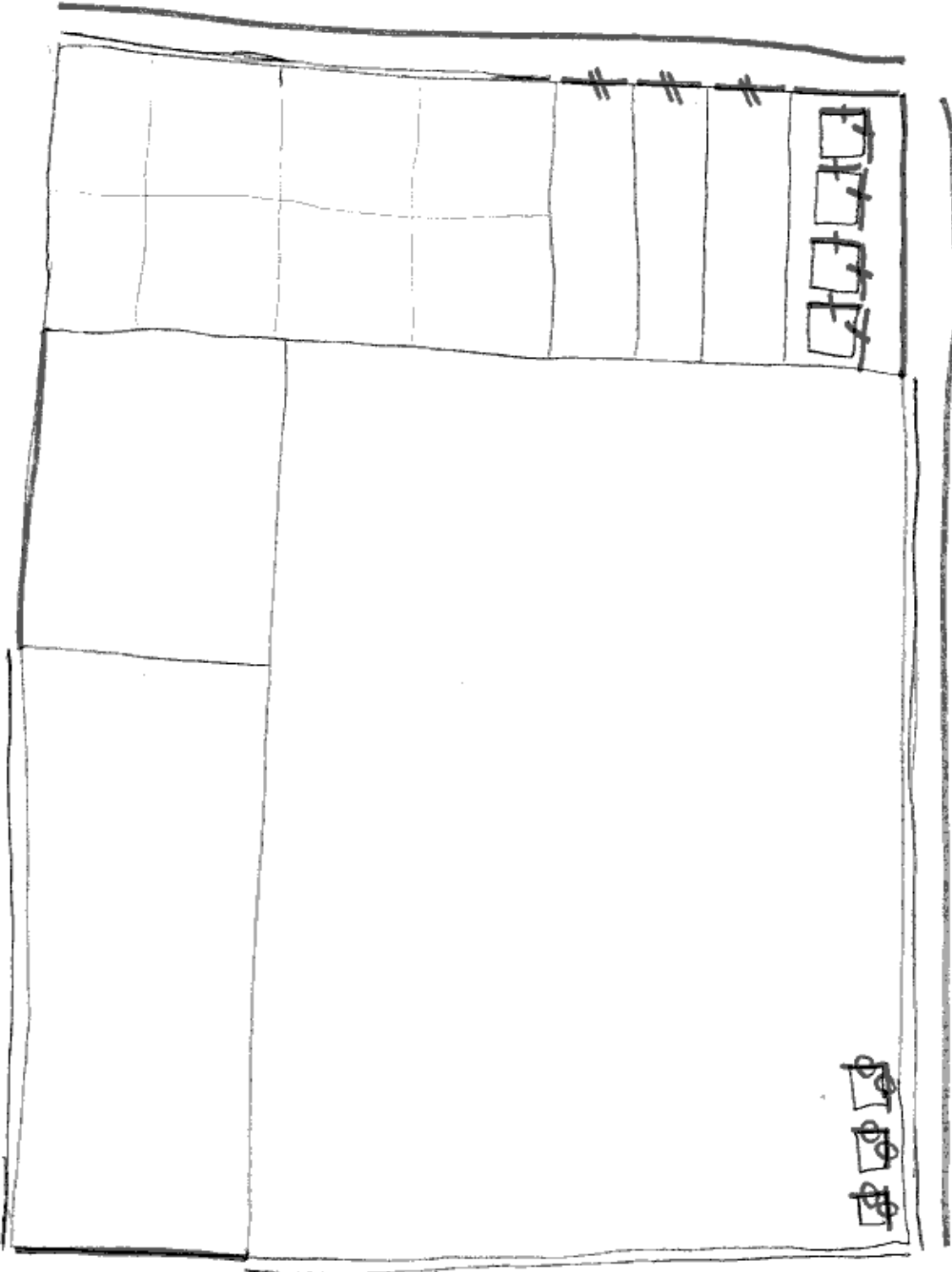
Lorsque la partie est finie:

- ↳ GAME OVER / VICTOIRE s'affiche sur la fenêtre, (plus rien d'autre ne réagit)

Dimensions

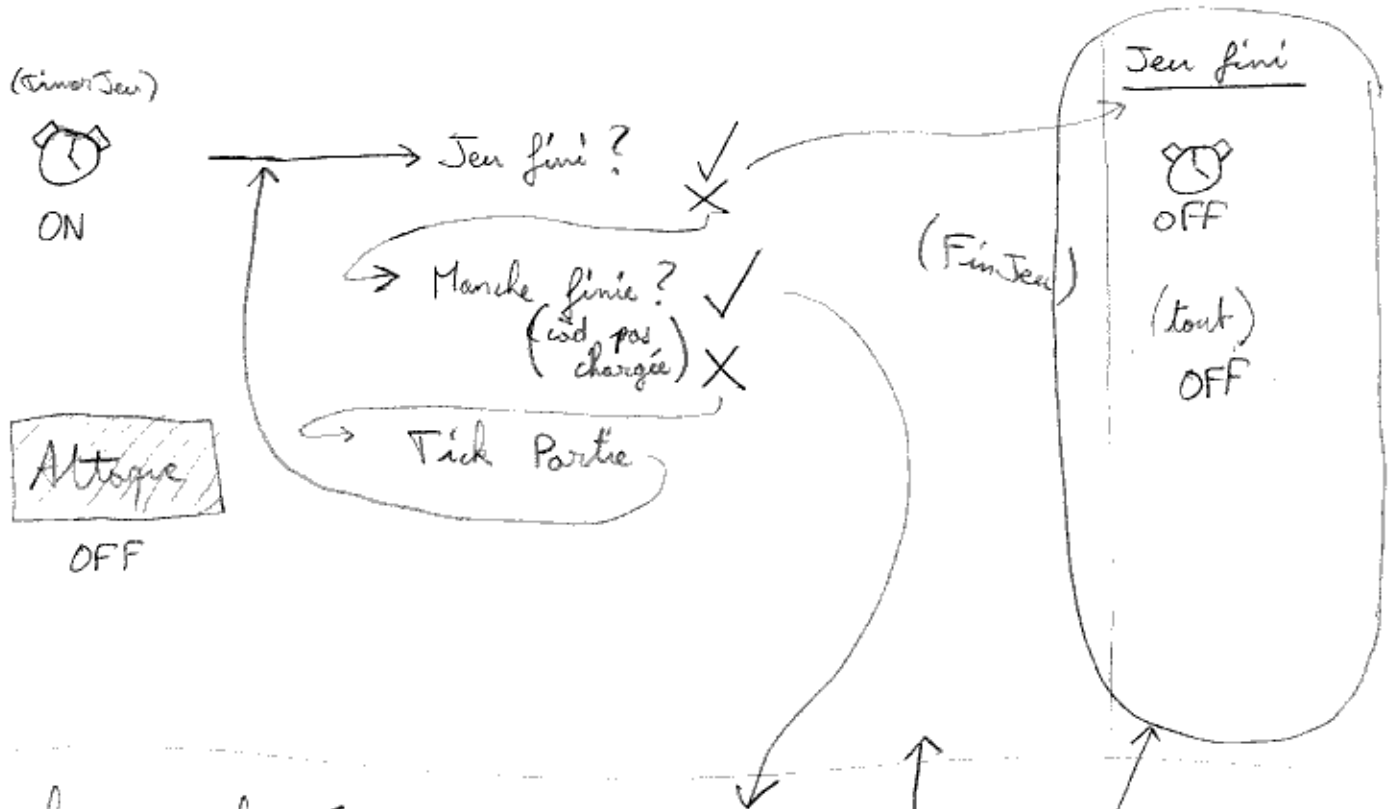
- ↳ Stockées dans l'objet DimJeu
- ↳ N'inclut pas le zoom ni la position sur la grille
- ↳ Seules les valeurs sont stockées, les composants se dimensionnent eux-mêmes

VARIABLES
METHODES
(dépendant des variables)

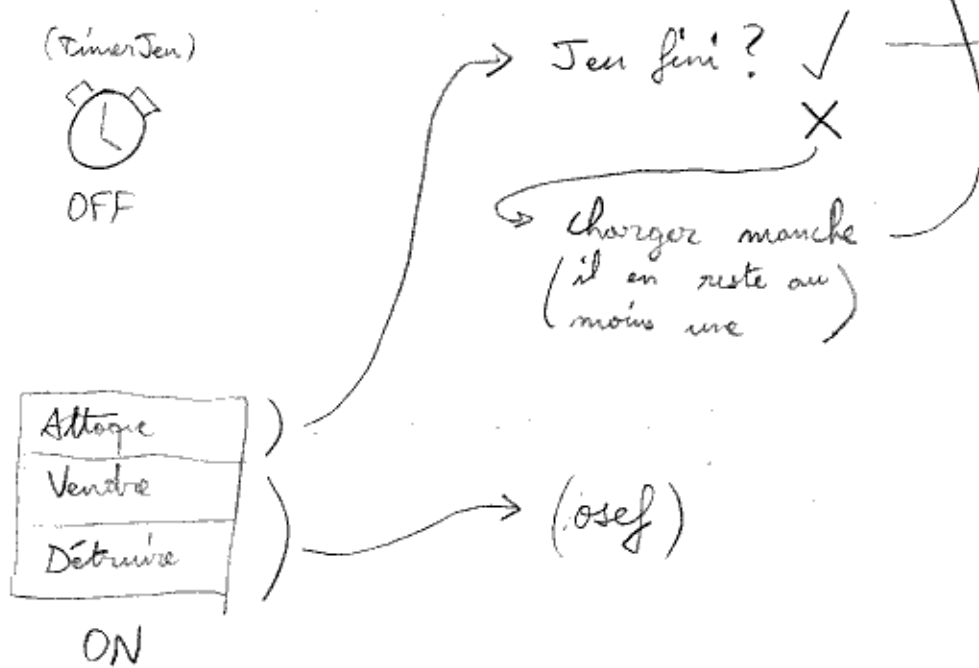


Comportement : Timer Jeu,
Attaque, fin du jeu

Manche chargée



Manche non chargée



Vérification de la fin de la partie / manche / chargement de la partie

↳ Tous les composants vérifient qu'une partie est bien chargée avant d'y accéder

(si une partie est chargée on suppose que toutes les références sont faites correctement)

↳ Timer Aff, avant d'actualiser la fenêtre, vérifie

si * la partie est chargée

CFinJeu

→ non : Tout bloquer, et afficher qu'aucune partie n'est chargée (CFenetreJeu)

→ oui :

* La partie est gagnée/perdue

→ oui : Idem

→ non :

* Vérifier si une manche est chargée

→ oui, et elle vient d'être chargée :

bloquer boutons ^{action} démarrer TimerJeu

→ oui, et elle vient de se finir :

décharger manche, stop TimerJeu,
débloquer boutons ^{action}

gère
par la
partie

Timers

↳ Possèdent une fréquence ^{→ stockée dans une variable (réglable)} et effectuent une action à intervalle de temps régulier (correspondant à la fréquence)

⚠ Doivent être arrêtés avant de quitter le programme!

Timer Jeu

↳ Vérifie si la partie est terminée

↳ oui → stop timer, ~~afficher GAME OVER / VICTOIRE~~
↳ non

géré par CFinJeu

↳ Vérifie si une manche est chargée

↳ oui → tick Partie

↳ non → stop timer

Timer Aff

↳ Actualise l'écran à une fréquence donnée, peu importe qu'une manche soit chargée ou non, ou que la partie soit terminée

↳ Envoie un tick à la gestion des messages

(faire disparaître les messages vieux)

Comportement : Zone Grille

Roue : case désigne une case du quadrillage formé dans ZoneGrille, qui peut contenir une tuile ou pas (1/2)

↳ Affichage de la grille (tuiles + endommageables)

Interactions

↳ Drag'n Drop: L'affichage ^{de la grille} peut être déplacée en l'attrapant et relâchant ailleurs

↳ Zoom: * Les boutons BPlus, BMoins, BCentre permettent de zoomer, dézoomer (jusqu'à un certain point) et centrer la carte

* La molette permet de zoomer / dézoomer au dessus de la case pointée par la souris

↳ Passage de la souris:

- * La tuile (s'il y en a une) sous le curseur est distinguée des autres (affichée en surbrillance / encadrée)
- * Lorsque le curseur passe au dessus d'une tour / un endommageable, des informations sont affichées dans ZoneInfos

Sélectionneur

↳ Sélection:

* Lorsque un clic est effectué

→ au dessus de la tuile sélectionnée } Plus aucune
→ au dessus d'une case sans tuile } tuile sélectionnée

(→ au dessus d'une tuile pas sélectionnée) cette tuile est sélectionnée

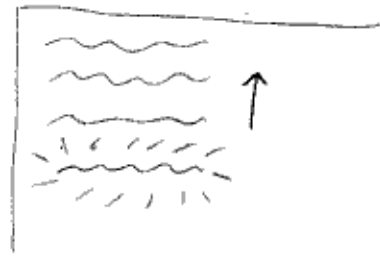
La tuile sélectionnée est distinguée des autres

Comportement : Zone Grille - affichage des messages

↳ Gère une file de messages, qui s'affichent en haut à gauche.

↳ Une méthode permet d'ajouter un nouveau message

↳ Les messages apparaissent en dessous des précédents et remontent au fur et à mesure que les précédents disparaissent



↳ Stocke une ^{valeur de} durée de vie des messages, fixée (en s).

↳ Gère une file de cooldown, correspondant aux temps d'affichage en tick restant des différents messages d'affichage

↳ A chaque tick, chaque cooldown de la file décroît, et les éléments dont le cooldown vaut 0 sont défilés.

Comportement : CTmgJeu

↳ Gère des collections d'images et les rend accessible aux autres composants

↳ En particulier, gère deux dictionnaires d'images dont les clés sont un couple

type d'effet
type d'endommageable

 + "dimensions

et les valeurs sont les images correspondantes (à la bonne dimension)

↳ Les images sont accessibles à partir d'une méthode prenant en paramètre la clé

↳ si une instance dimensionnée a déjà été créée, (et est dans le dictionnaire), la renvoyer

↳ sinon, redimensionner l'image correspondante

↳ Met à disposition une méthode pour charger une image à partir de son nom (dans le dossier src/main/resources/images)

↳ ou d'un type d'image +

Sélectionneur

- ↳ Possède une grille de cases (plan) dont certaines d'entre elles situées sur un rectangle peuvent être parcourues et sélectionnées (voir comportement : ZoneGrille)
- ↳ Possède les bornes du rectangle
- ↳ L'orientation, les bornes sont les mêmes que pour la carte
- ↳ Possède un offset en x et en y , qui permettent de décaler l'affichage du rectangle
- ↳ Réagit aux événements des clics

