# Partie

↳ Lien entre l'affichage et les différents constituants d'une partie

↳ Gestion de l'argent (achat / vente d'une tour)

↳ Gestion du magasin (Tours et améliorations disponibles + prix)

(↳ Garde un compteur de ticks depuis le début du jeu)

↳ Gestion du mécanisme gagné / game over

## En particulier

↳ La partie est gagnée quand la dernière manche est gagnée

↳ La partie est perdue quand ~~la tour principale est détruite~~ la manche dernière est perdue

(~~↳ En cas de conflit la partie est perdue~~)

↳ 9 manche

# Carte

↳ Apparition des tours
↳ Disparition des tours
↳ Apparition des ennemis
↳ Disparition des ennemis
↳ Répartition des ticks

↳ Gestion des tours, de la tP, des ennemis

↳ Gestion des tuiles, dimensions de la grille (Stockage)

En particulier
↳ Osef des déplacement des monstres (pour l'instant)
↳ Par contre elle controle si une tour est déjà sur la case où une nouvelle tour est censée apparaître

↳ La tour principale fait aussi partie de la collection rassemblant les tours

# Tuile

↝ Gestion de l'accès aux tours et ennemis
(est-ce que des tours / monstres peuvent être sur cette case ?)

↝ Stockage de l'image de fond

En particulier

↳ Les éléments héritant de Tuile sont des case clues

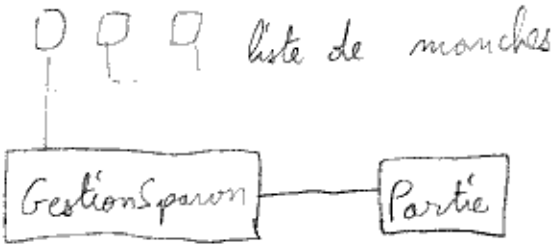
# GestionManches

↳ Gestion d'une liste de manches

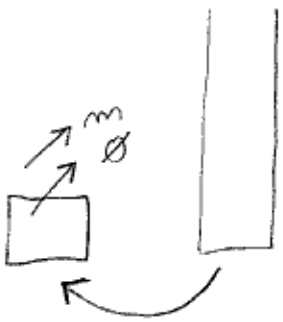
↳ Transmission des ticks à la manche en cours

↳ Répond si une manche est en cours ou pas

↳ Charge la manche suivante




Mécanisme :




chargeM vérifie si aucune manche n'est en cours, puis déplace (retire) la première manche de la liste si c'est le cas

En particulier :

↳ Chaque manche se débrouille pour faire apparaître les ennemis (ou d'autres choses), GestionSpawn ne s'en occupe pas (faire)

↳ Quand une manche est terminée elle renvoit une exception lors de l'appel suivant à tick

# Manche

↳ Reçoit des ticks, et fait se déclencher des actions (apparition d'ennemis, ou nouvelles tours, effets, ...)

(↳ possède un compteur de ticks depuis le début de la manche)

↳ possède une condition de fin (méthode renvoyant gagné, perdu ou rien)

---

En particulier :

↳ Une manche contient une méthode tick et fait ce qu'elle veut avec cette méthode

↳ Renvoie ~~une exception~~ lorsque la manche est terminée

# Magasin

↳ Gestion d'une liste d'achats disponibles

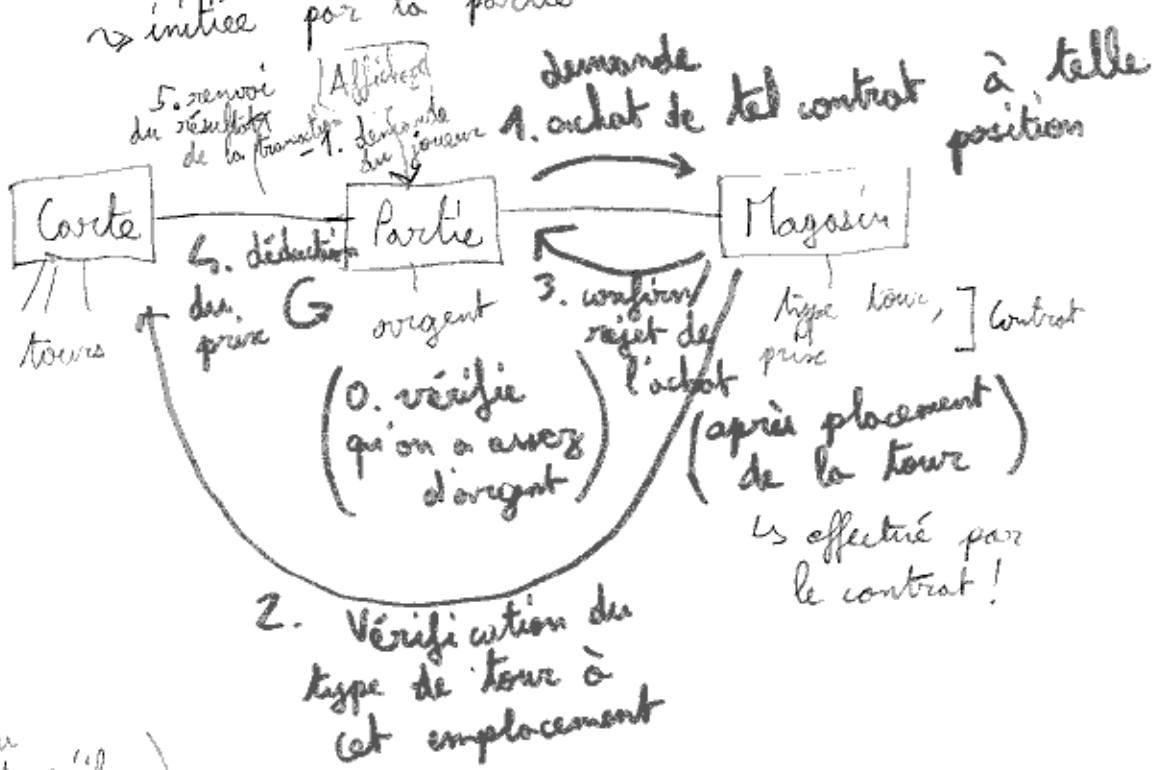
⇝ { prise
type de tour / amélioration
éventuellement une condition d'achat (ex nb de ticks) ⇝ on verra plus tard

Contrat

En particulier :

↳ Procédure d'achat d'une tour

⇝ initiée par la partie




(⚠ Penser au remplacement s'il s'agit de la tour principale)

Contrat

↳ Stocke les informations suivantes

- type tour à acheter
- type tour qui doit être remplacé
- prise

↳ peut effectuer des actions supplémentaires lors de l'achat

↳ c'est lui qui s'occupe de faire spawner l'endommageable

Endommageable

↳ position (ou pas)

↳ PV et PVMAX

↳ portée ↳ degats ↳ rayon (rayon d'effet d'une ACE)

↳ vitesse ↳ soin ↳ cooldown

(↳ stratégie ? ↳ action lors de la mort ?)

↳ effets ↳ type

---

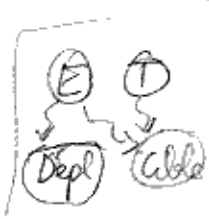
En particulier :

↳ Les stratégies des tours et ennemis (actions lors des ticks) ne sont pas définies dans des références.

Chaque type d'ennemi/tour possède une méthode tick, qui fait appel à des algorithmes codés séparément

ex Un ennemi voudra se déplacer et choisir sa cible ✓

Une tour ✗



Tour

↳ Gestion de ses PV (reçoit des attaques)

↳ Gestion de sa position (⇔ la case où elle est située)

↳ Gestion de ses attaques (quand elle attaque)

↳ f Endommageable

En particulier, ne retient pas les stratégies elle-même, les stratégies sont implémentées ailleurs

## Ennemi

↳ Gestion de ses PV (reçoit des attaques) ↓ et de sa mort

↳ Gestion de sa position

↳ Gestion de ses attaques (= de chaque fois qu'il attaque)

↳ Gestion (pas implémentation) de sa stratégie d'attaque

En particulier

↳ Ne gère <u>PAS</u> les stratégies d'attaque, de déplacement lui-même ! (appel à des algorithmes codés séparément)

**Effets** (Les ennemis et tours possèdent une liste d'effets qui s'appliquent sur eux)

↳ s'applique à un ennemi ou une tour

↳ Possède (ou pas) un cooldown (temps restant avant disparition)

↳ Modifie des caractéristiques des tours / ennemis

ex portée, pixDmg

ou effectue des actions supplémentaires à chaque tick

ex soigner, enlever des pv...

---

En particulier

↳ Un effet a quatre méthodes,

↗ begin (s'applique à l'application de l'effet)

↗ tick (action s'appliquant à chaque tick)

↗ end (s'applique au retrait de l'effet)

↗ mort (s'applique à la mort de l'entité qui a l'effet)

(↝ A prévoir : Gestion des retrait des effets lors de la mort / fin de la partie)

Voir : mécanisme d'application des effets

## Type Endommageable

↳ Gestion de l'image d'affichage sur la carte

↳ Gestion de l'image d'affichage dans le magasin

↳ Eventuellement, autres ressources concernant la classe associée (ex images supplémentaires)

↳ ~~Méthode permettant d'instancier la classe associée~~

Pas possible, car problème de type...

---

En particulier

↳ Les éléments héritant de cette classe sont des objets appelés Type__ où __ est le nom de la tour / de l'ennemi

Type Effet

↳ stocke l'image (logo) représentant l'effet

↳ peut contenir des | méthodes générales concernant
| autres ressources

un type d'effet

(↳ ~~méthode permettant de créer un nouvel effet ?~~)
a priori pas besoin

---

En particulier

↳ peut contenir des méthodes

begin_default
tick_default
end_default
mort_default

qui un effet pourra utiliser

↳ Mécanisme d'application

Void ↗