

```

Mode precis:
|>
Tour 2
<|
{-----} ^ ^ ^ ^ ^ ^ ^ ^ ^ [^^^^^^]
|A ---> | <--- B
|10      | 10
|Fant    | Fant
|         |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Base A   ----- < AGE OF WAR > ----- Base B
HP:100   ----- < AGE OF WAR > ----- HP:100
OR:14    ----- < AGE OF WAR > ----- OR:14
|||||||  ----- |||||||
Joueur A creat un soldat: (1.Fantassin 2.Archer 3.Catapulte 0.Pass  [ 8.Quitter ce jeu ] ) 2

```

Action 1 :

```
Mode precis: Tour 3
```

	>										<
{-----}	A --->	A --->									[^ ^ ^ ^ ^]
8	10									<--- B	
Arch	Fant									10	
										Fant	

Base A	< AGE OF WAR >									Base B	
HP:100	< AGE OF WAR >									HP:100	
OR:10	< AGE OF WAR >									OR:22	

Tour 2 est fini!

Tour 3 joueur A actionne!

Action1 est fini. Countinuer? [8 pour quitter] █

Action 2 :

```

Mode precis:                                Tour 3                                <|
      |>
      |
{-----} ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ {^^^^^^^}
|A --->| A --->|                                     <--- B
|8      | 10    |                                     10
|Arch   | Fant  |                                     Fant
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|Base A |-----< AGE OF WAR >-----|Base B
|HP:100 |-----< AGE OF WAR >-----|HP:100
|OR:10  |-----<           >-----|OR:22
||||||| |-----|-----|-----|-----|
Action2 est fini.  Countinuer? [ 8 pour quitter ] █

```

Action 3 :

```

Mode precis:                                Tour 3                                <|
      |>
{-----}          ^          ^          ^          ^          ^          ^          ^          ^          [AAAAAAAA]
|A --->|A --->|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|8     |10    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|Arch  |Fant  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----|-----|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|Base A|-----<----->----->----->----->----->----->----->----->----->----->----->|Base B
HP:100|-----< AGE OF WAR > ----->----->----->----->----->----->----->----->----->----->----->|HP:100
OR:10 |-----<----->----->----->----->----->----->----->----->----->----->----->|OR:22
|||||||-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Action3 est fini.   Continuer? [ 8 pour quitter ] █

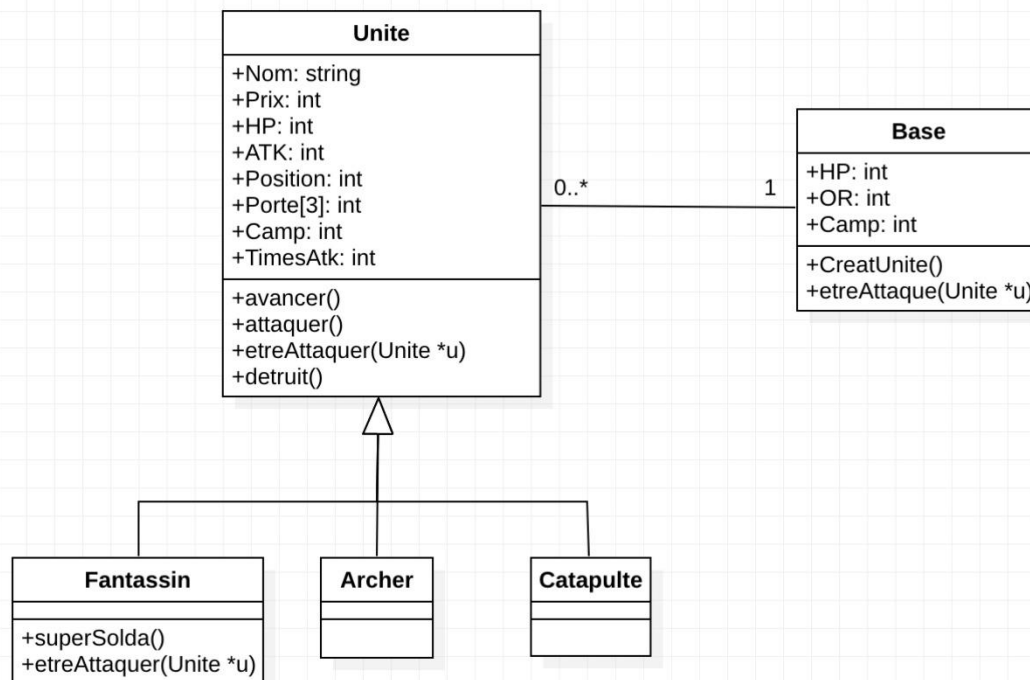
```

Après le joueur fait un choix pour le modèle d'affichage, le programme demandera de choisir encore le modèle de jouer (1p vs 2p ou 1p vs computer) par taper 0/1 plus un \Entre.

Jouer avec computer? 0 vs Joueur / 1 vs Computer

Une fois le jeu démarre, le joueur peut quitter à chaque moment quand le jeu s'arrête par taper numéro 8 plus un \Entre. Mais **attention**, vous ne pouvez stocker le jeu qu'à la fin d'un tour, si vous quittez le jeu avant un tour est fini vous allez perdu toutes les données.

Diagramme de classe :



Le rôle de chaque classe :

Base : Sert à construire les deux bases A et B

Lire les choix qu'un joueur a fait et construire des unités de chaque base.

Un numéro random qui peut réaliser un AI joueur.

Fonction etreAttaque() pour diminuer le l'Hp de la base une fois si il est attaqué.

Unite : Construire les unités différentes

Fonction avancer() pour faire bouger les unités dans le champ (premièrement distingue si l'unité pouvoir avancer ou pas)

Fonction attaquer() pour marquer si une unité a faire l'action1 ou pas.

Fonction etreAttaque() pour diminuer le l'Hp de l'unité une fois si il est attaqué, au moment qui est tué, la fonction divise son prix par deux et le donne à la base d'ennemi.

Fonction detruit() pour détruire l'unité une fois son Hp est moins ou égale à 0.

Fantassin : Construire une unité comme Fantassin

Fonction Supersolda() change le nom de Fantassin 'Fant' par 'Super'

Fonction etreAttaque(Unite* u) pour diminuer le l'Hp de l'unité une fois si il est attaqué, au moment qui est mort la fonction divise son prix par deux et le donne à la base d'ennemi. Et vérifier si c'est un fantassin d'ennemi qui l'a tué, si oui, le fantassin d'ennemi devient un supersolda.

Archer : Construire un archer

Catapulte : Construire une catapulte

Problèmes rencontrés :

1. Au début de ce projet, on a eu un problème dans l'affichage du jeu. Comme on a utilisé les '|' pour diviser les 12 cases, on ne pouvait pas toujours afficher les '|' au même place dans chaque rang, les données sont différentes dans les cases donc combien d'espace qu'on devait laisser pour chacune ?

Façon de régler : Finalement on a remarqué qu'on peut utiliser '\t' comme Tab pour formater les cases, mais il faut faire attention à la longueur de donnée dans une case ne doit pas excéder 7 caractères. (un caractère est laissé pour '|'.)

2. Encore un problème avec l'affichage, on ne savait pas comment actualiser le terminal après une action ou un tour donc on a choisi d'afficher tous les résultats rang par rang et ça donnait énormément de chose sur l'écran.

Façon de régler : On a cherché sur Internet et a trouvé une fonction system("clear") qui s'occupe d'actualiser le terminal, on l'a placée avant chaque afficher().

3. On a utilisé un Switch dans la classe Base pour offrir 4 choix à joueur de créer ses unités différentes avec un std::cin>>c pour lire le clavier, mais si le joueur n'avait pas assez d'argent pour construire l'unité qu'il a choisi, le programme s'arrêtait, il ne pouvait plus continuer.

Façon de régler : Utiliser une phrase goto(Label) avec un label qui se place avant std::cin pour réaliser une récursion, une fois on tombe sur une unité qu'on ne peut pas construire, le programme va demander un autre choix.

4. Un autre problème avec le Switch : on a remarqué que si on a tapé un caractère qui n'était pas un entier, la programme allait ignorer le std::cin et tomber dans une boucle sans fin.

Façon de régler : C'était le std::cin qui pose le problème. Dans le cas qu'on a tombé sur un autre type de caractère on a ajouté un cin.clear() et un cin.ignore() pour vider le cin et enlever l'influence des caractère qu'on a tapé dans le dernière cin.

5. A cause de abus d'appel des fichier.h ou .cpp, au moment de compilation il existe souvent des erreur comme une redéfinition de quelque fonction ou classe (peut-être son fichier a été appelé plusieurs fois par autre fichier.)

Façon de régler : On a réorganisé les en-têtes des fichiers pour éviter au maximum ce type de problème, pour le fichier base.h on a ajouté #ifndef pour être sûr qui ne peut être appelé qu'une seul fois.