

1h50min pour créer une App Front avec la Clean Archigonale ©

Tout le monde pense que la "Clean Architecture / Architecture Hexagonale" se fait uniquement côté Back.
Dans ce live coding, je vais vous montrer que cela fonctionne également côté Front.

Et oui, Redux n'est pas le Silver Bullet des applications front-end.

Et oui, Les Stores ne sont pas toujours obligatoires, mais ceci est une autre histoire !

On va coder ensemble une petit App Front pour voir comment mettre en place cela.

Take away :

Architecture

Stratégie de Tests



1h50min pour créer une **App Front** avec la
Clean Archigonale

6 Juin 2024





**Dev
Quest**

**1h50min pour créer une App Front avec la
Clean Archigonale ©**

6 Juin 2024





Guillaume Chauvet
7 ans d'xp

*“C’est en sciant que
Léonard Da vinci”*

Tech Lead

Développeur

Formateur

OCTO
Technology
Part of Accenture



Dorian Lamandé

15 ans d'xp

*“La confiance n'exclut
pas
le contrôle”*

Tech Lead

Développeur

Formateur

Leader
de tribu

OCTO
Technology
Part of Accenture



@zetidodo



dlamande



@instantcoding



@dorianlamande



@zetidodo

***Qui a assisté ou visionné le
talk de Dorian ?***



***Quelle est votre expérience
sur le sujet ?***



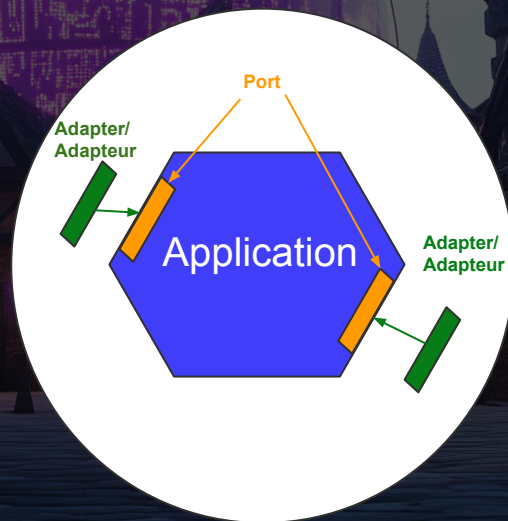
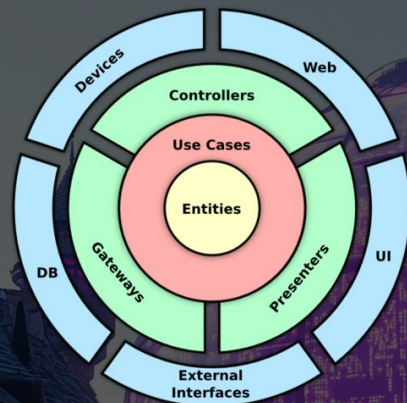


**Dev
Quest**

Un rappel rapide

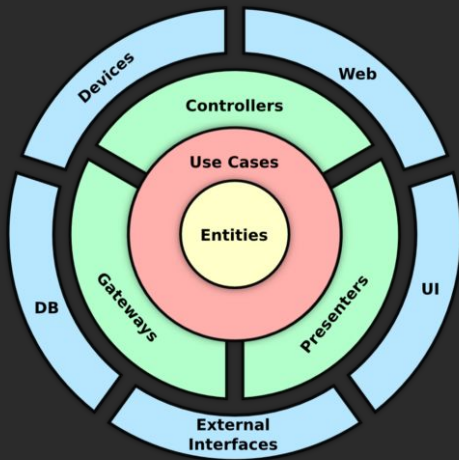


Isoler c'est gagné !

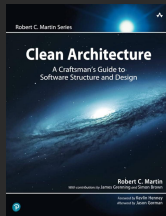


Dev
Quest

La Clean Architecture



Clean architecture par
Robert C. Martin en
2012



Structure

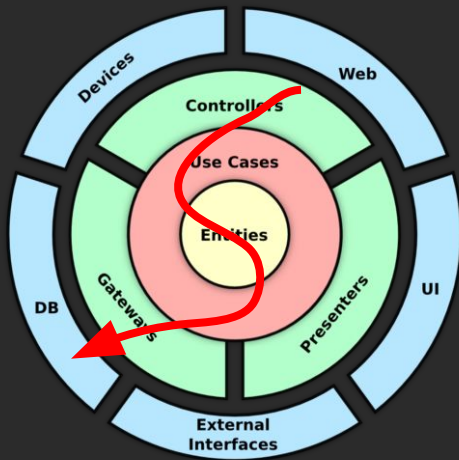
```
interface Presenter {  
    /** Règles de présentations **/  
}
```

```
class Usecase {  
    /** Orchestration des Règles de l'applications **/  
}
```

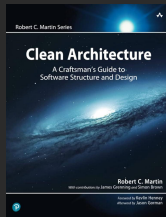
```
class Entity {  
    /** Règles métiers **/  
}
```

```
class PostGRESomeRepository implements  
SomeRepository {  
    /** Accès BDD **/  
}
```

La Clean Architecture



Clean architecture par
Robert C. Martin en
2012



Structure

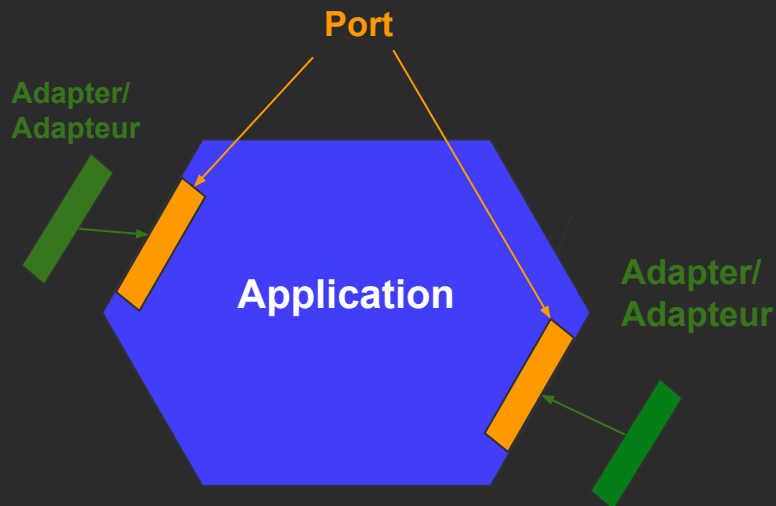
```
interface Presenter {  
    /** Règles de présentations **/  
}
```

```
class Usecase {  
    /** Orchestration des Règles de l'applications **/  
}
```

```
class Entity {  
    /** Règles métiers **/  
}
```

```
class PostGRESomeRepository implements  
SomeRepository {  
    /** Accès BDD **/  
}
```

L'architecture hexagonale

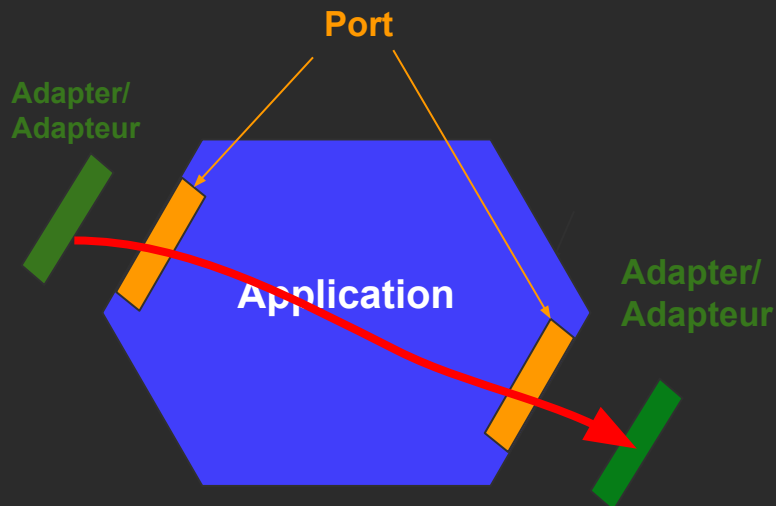


Architecture hexagonale par
Alistair Cockburn en 2005

Structure

```
interface SomePort {  
}  
  
class SomeAdapter implements SomePort {  
    /** Toute la partie technologique **/  
}  
  
interface Application {  
}  
  
class ApplicationImpl implements Application {  
    /** Règles de l'application & métier **/  
}
```


L'architecture hexagonale

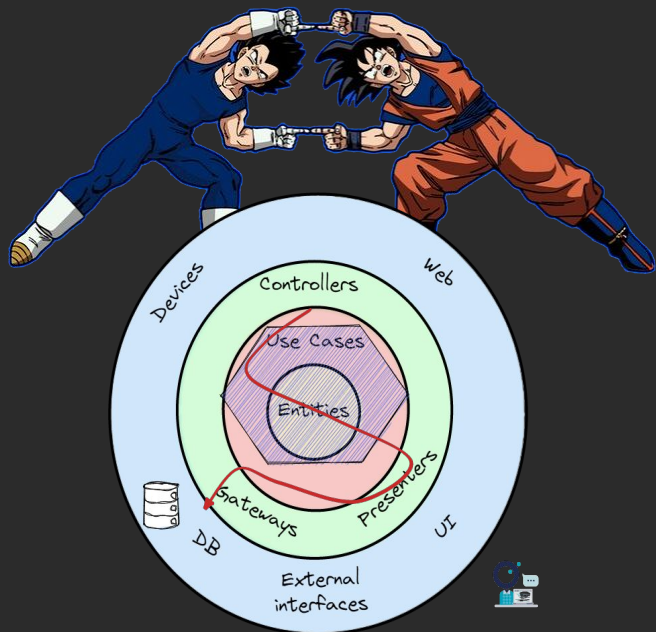


Architecture hexagonale par
Alistair Cockburn en 2005

Structure

```
interface SomePort {  
}  
  
class SomeAdapter implements SomePort {  
    /** Toute la partie technologique **/  
}  
  
interface Application {  
}  
  
class ApplicationImpl implements Application {  
    /** Règles de l'application & métier **/  
}
```

Le *meilleur* des 2 mondes



La Clean Archigonale par Christophe Breheret-Girardin en 2023

Structure

- > `onboarding`
- ▼ `profileUtilisateur`
 - > `adapters`
 - > `ports`
 - `chargerProfileUtilisateur.usecase.ts`
 - `mettreAJourProfileUtilisateurUsecase.ts`
- ▼ `quiz`
 - > `adapters`
 - > `ports`
 - `chargementQuiz.usecase.ts`
 - `envoyerDonneesQuizInteraction.usecase.ts`
 - `evaluerQuiz.usecase.ts`
- ▼ `recommandationsPersonnalisees`
 - ▼ `adapters`
 - `recommandationsPersonnalisees.presenter.impl.ts`
 - `recommandationsPersonnalisees.repository.axios.ts`
 - ▼ `ports`
 - `recommandationsPersonnalisees.presenter.ts`
 - `recommandationsPersonnalisees.repository.ts`
 - `recommandationPersonnaliseeAEteCliquee.usecase.ts`
 - `recupererRecommandationsPersonnalisees.usecase.ts`
 - `recupererRecommandationsPersonnaliseesUnivers.usecase.ts`
- > `score`
- > `services`
- > `thematiques`

“La valeur du front ***ce n’est pas
les frameworks/bibliothèques***
qu’on utilise”



“Dans le front, la valeur c’est :

✓ la partie du code où l’on trouve
les règles fonctionnelles

✓ la représentation ***graphique***
du métier

✓ assurer une ***expérience***
utilisateur de qualité”





**Dev
Quest**

Let's code !

Le DevQuest est en danger !

Une invasion de ***Vilains*** va avoir lieu !

On a besoin d'envoyer ***nos guerriers et guerrières*** avec ***nos meilleures armes***.

Aidez-nous !



1. *Récupérer le starter-kit*
shorturl.at/ijQWB

2. *git checkout start*

3. *npm install*

4. *npm run dev*



“Le Header”

On veut afficher les **golds** et la **puissance d'attaque** du soldat

SHOP BATTLE

1000



0



LE SHOP



Épée en bois pas ouf

Prix : 100



ACHETER



Epée de boisaille

Prix : 300



ACHETER



Épée epic gneu taper

Prix : 1200



ACHETER

git checkout weapons

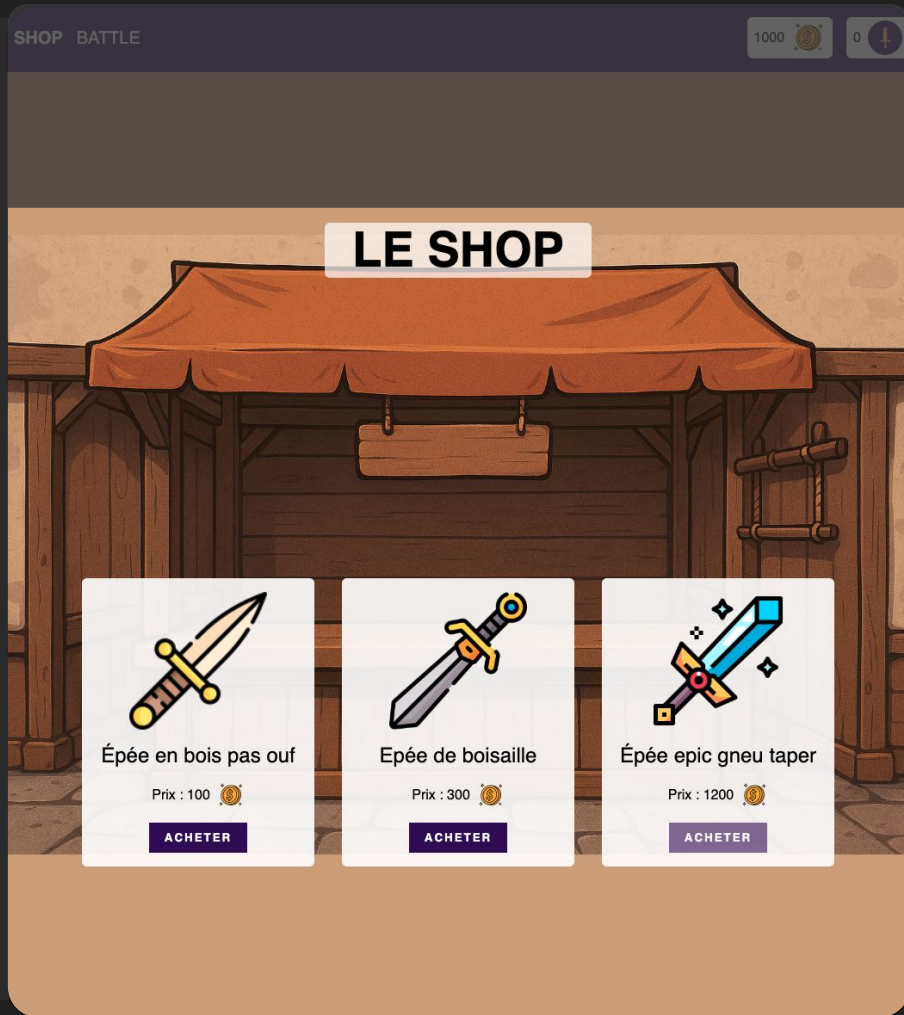


“Le Shop”

On veut *afficher les armes* et
leur *prix*

*Si on n’a pas assez de gold :
on ne peut pas acheter
l’arme*

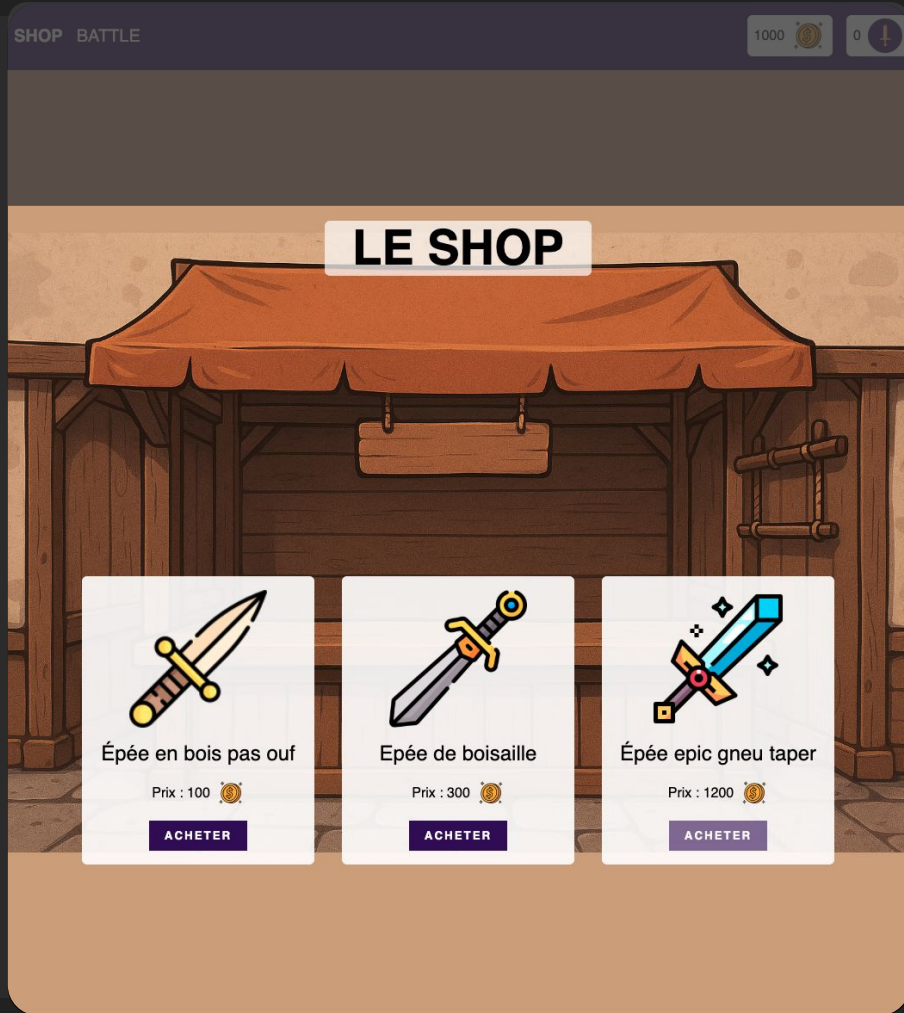
*Après l’achat, le “Header”
doit se mettre à jour*



“Le Shop”

On veut afficher les armes et leur prix

*Si on n'a pas assez de gold :
on ne peut pas acheter
l'arme*



git checkout shop

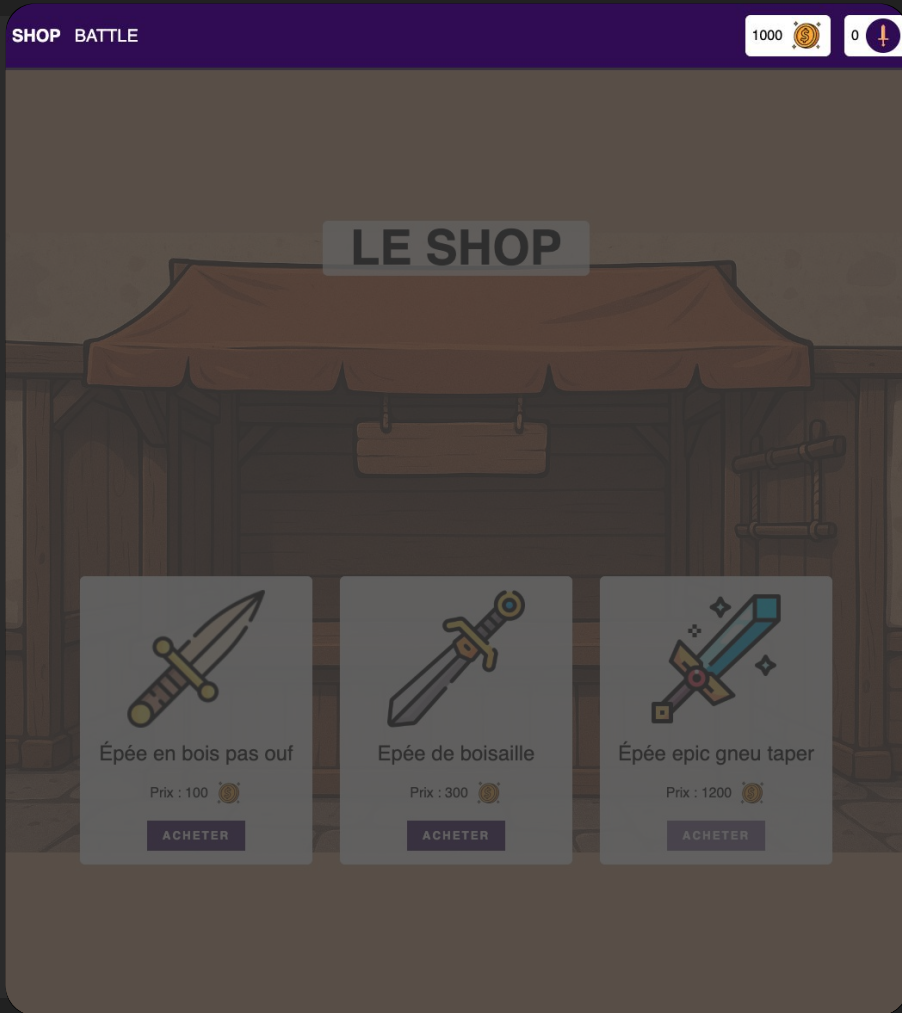


“Le Shop”

On veut afficher les armes et leur prix

*Si on n’a pas assez de gold :
on ne peut pas acheter
l’arme*

*Après l’achat, le “Header”
doit se mettre à jour*



git checkout enemies

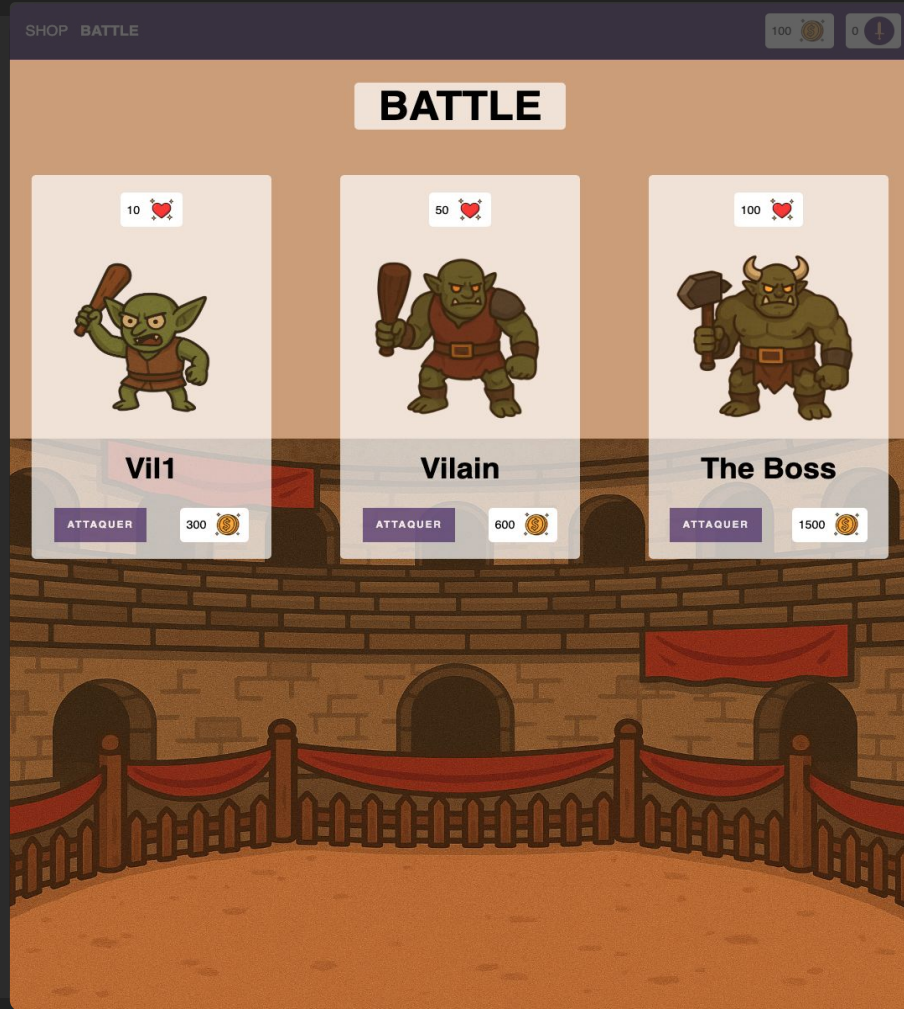


“Battle”

On veut afficher les “**Vilains**”
et leurs **points de vie**

Si on n’a pas assez de
puissance d’attaque : on ne
peut pas attaquer le vilain

Après avoir vaincu un
“Vilain” le “Header” se met à
jour

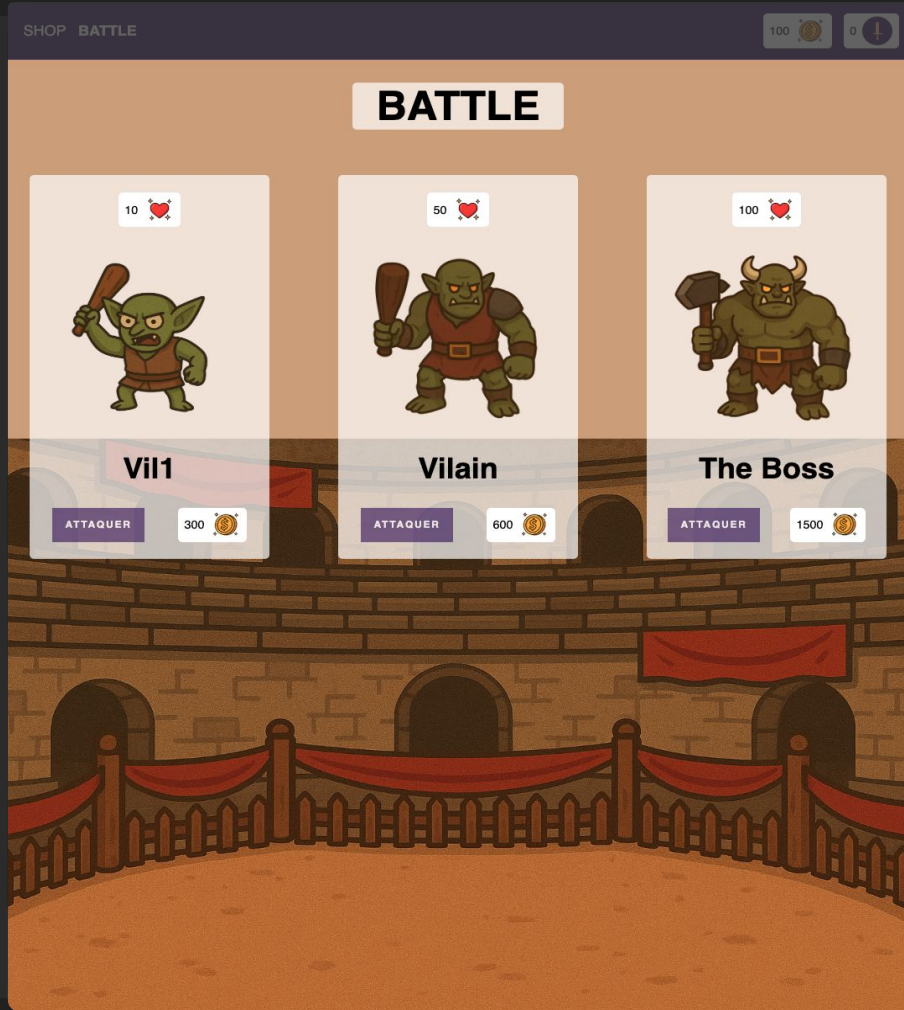


“Battle”

On veut afficher les “Vilains”
et leurs points de vie

Si on n’a pas assez de
puissance d’attaque : on ne
peut pas attaquer le vilain

Après avoir vaincu un
“Vilain” le “Header” se met à
jour

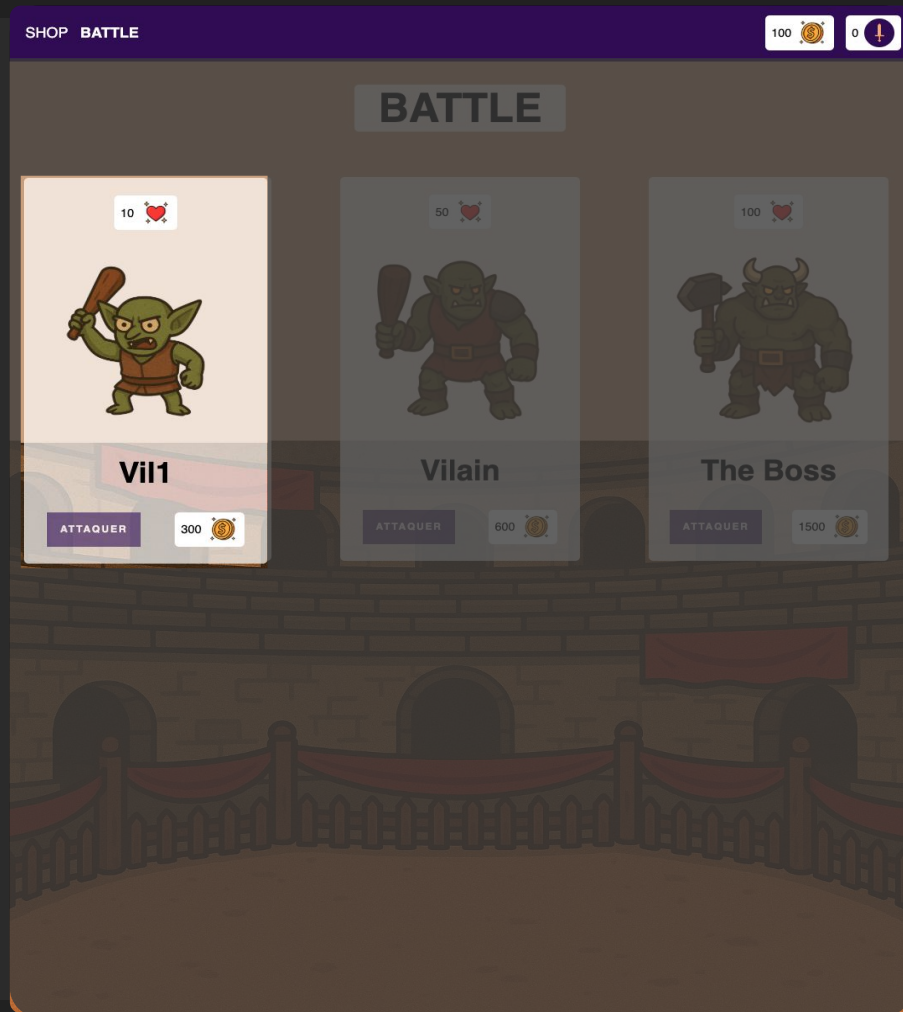


“Battle”

On veut afficher les “Vilains”
et leurs points de vie

Si on n’a pas assez de
puissance d’attaque : on ne
peut pas attaquer le vilain

Après avoir vaincu un
“Vilain” le “Header” se met à
jour



Ce ***n'est pas adapté***
à tous les projets !





DevQuest



C'est tout pour nous !

Feedback

