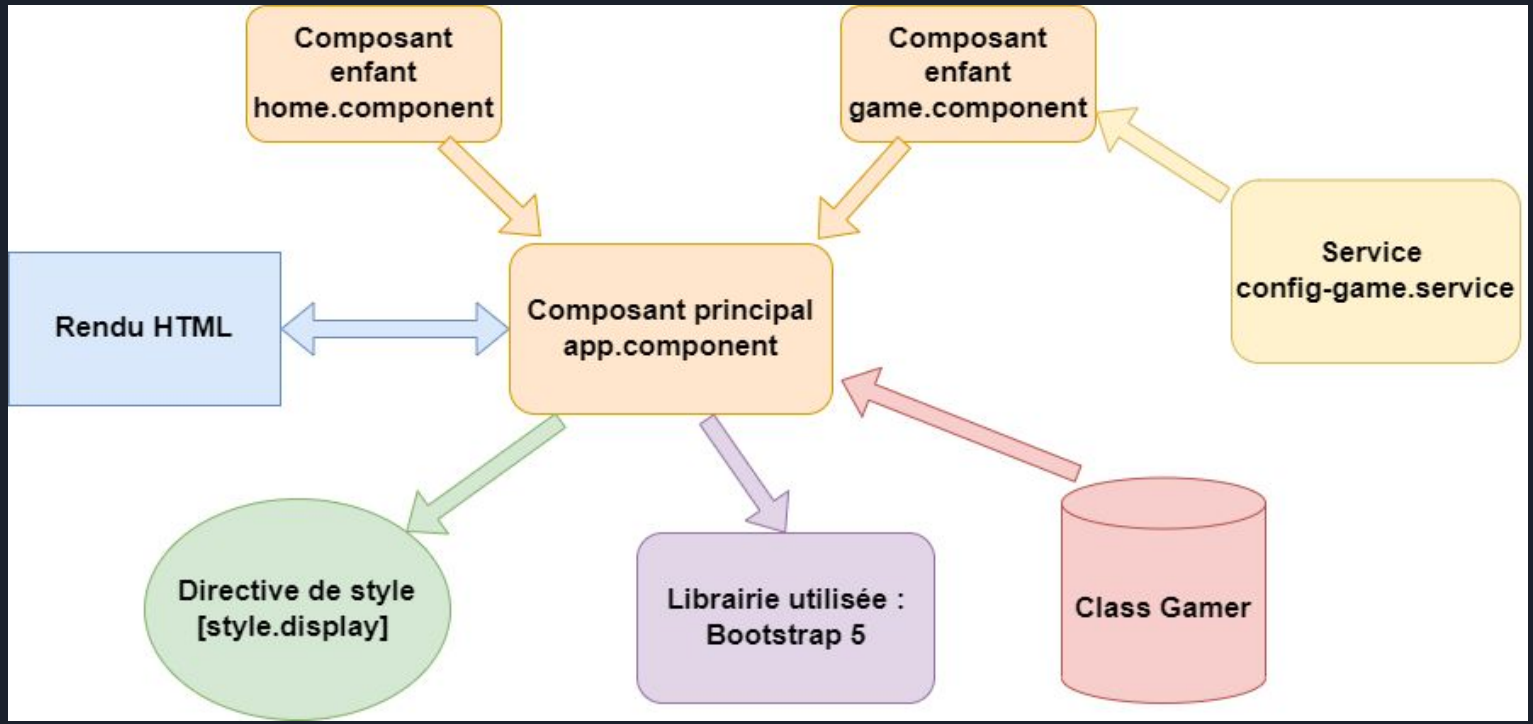
Développement front end : Angular

Jeux du morpion

Charlotte Vial
Nicolas Gilles

TABLE DES MATIÈRES

Présentation de l’architecture (Schéma)



Présentation du code & éléments Angular utilisés

Afin de profiter d'un design modern et simple à mettre en place, nous avons utilisé la librairie Bootstrap.

Le component principal app.component va insérer dans son HTML les components enfants game.component et home.component.

```
<app-home [style.display]="formVisibility" (sendRequestToFather)="letsPlay($event)"></app-home>

<div [style.display]="gameVisibility" class="mt-5 d-flex justify-content-around">
    <div [style.display]="gameVisibility" class="tailleGamer">Joueur O : {{gamerO?.getPseudo()}}</div>
    <div [style.display]="gameVisibility" class="tailleGamer">Joueur X : {{gamerX?.getPseudo()}}</div>
</div>
<app-game [style.display]="gameVisibility"></app-game>
```

Au départ on initialise home.component qui contient le formulaire en visible et la grille du jeu en display none grâce à une directive de style.

Afin de transférer les noms des joueurs du formulaire de home.component à game.component pour afficher les joueurs on a utilisé un @Output dans home.component.ts afin de faire transférer les données de l'enfant au parent. Grâce à cela je peux récupérer le pseudos des joueurs.

Présentation du code & éléments Angular utilisés

La fonction liée à sendRequestToFather va écouter les évènements bindés sur le bouton de home.component.html. Si il y a un click on affichera la grille de jeu et on mettra le formulaire en display none.

```
letsPlay(name: any) {
  if(name[0] != '' && name[1] != '') {
    this.formVisibility = "none";
    this.gameVisibility = "block";
    this.gamerO = new Gamer(name[0]);
    this.gamerX = new Gamer(name[1]);
  }
  else {
    alert("Veuillez compléter les pseudos des jours.");
  }
```

On peut également voir ci-dessus que l'on initialise pour chaque joueur une instance de la class Gamer.

```
export class Gamer {
    pseudo: string;
    constructor(pseudo: string){
        this.pseudo = pseudo;
    }

    getPseudo(){
        return this.pseudo

    }
}
```

Présentation du code & éléments Angular utilisés

Sur le bouton du formulaire nous avons effectué un dataBinding afin de récupérer les valeurs des inputs.

```
<button (click)="onGame(inputO.value, inputX.value)"
```

```
onGame(el1: any, el2: any){
  this.gamerO = el1;
  this.gamerX = el2;
  this.sendRequestToFather.emit([this.gamerO, this.gamerX]);
  }
```

Pour jouer nous faisons appels à un dataBinding qui écoute le click sur chaque cellule de notre tableau. A chaque click, la fonction onGame() est appelée.

```
<td #td1 id="td1" (click)="playGame(td1)"></td>
```

```
event = 0
playGame(idCell: any){
  this.configGameService.play(this.event++, idCell);
  this.configGameService.matchNull();
  this.configGameService.didIWin();
}
```

Présentation du code & éléments Angular utilisés

Sur la capture ci-dessus, nous pouvons voir que la fonction onGame() fait appel à des méthodes provenants d'un service, c'est le service game-config.service contenant la logique de notre jeux. Ci-dessous la méthode play() de notre service.

```
play(event: any, cell: any) {
  if (event % 2){
    cell.innerHTML = "X"
  }else{
    cell.innerHTML = "O"
  }
}
```

Répartition du travail effectué

Pour la réalisation de ce projet, nous avons fait le choix original d'expérimenter le *pair programming* ou programmation en binôme. La programmation en binôme est une méthode agile de développement qui consiste à travailler ensemble sur le même poste. Ce projet Angular à était pour nous l'occasion de l'expérimenter. Ceci nous a permis de partager nos connaissances et de surmonter les obstacles plus rapidement. Cette expérience fut très enrichissante et nous recommencerons dans l'avenir si cette occasion se présente !

Note attribuée à notre projet

Nous nous attribuons la bonne note de 15.5/20 pour ce projet .

Effectivement, malgrés les difficultés rencontrées pendant l'apprentissage de ce framework, nous somme content du résultats obtenus. Nous avons répondu aux exigences principale du jeux, c'est à dire la gestion des différentes façon de gagner ainsi que la gestion des matchs nul. Comme demandé, le formulaire d'inscription des joueurs est présent. Cependant, limité par le temps, nous n'avons pas implémenter la fonctionnalité permettant de voir le nom du joueurs gagnant.

Merci !