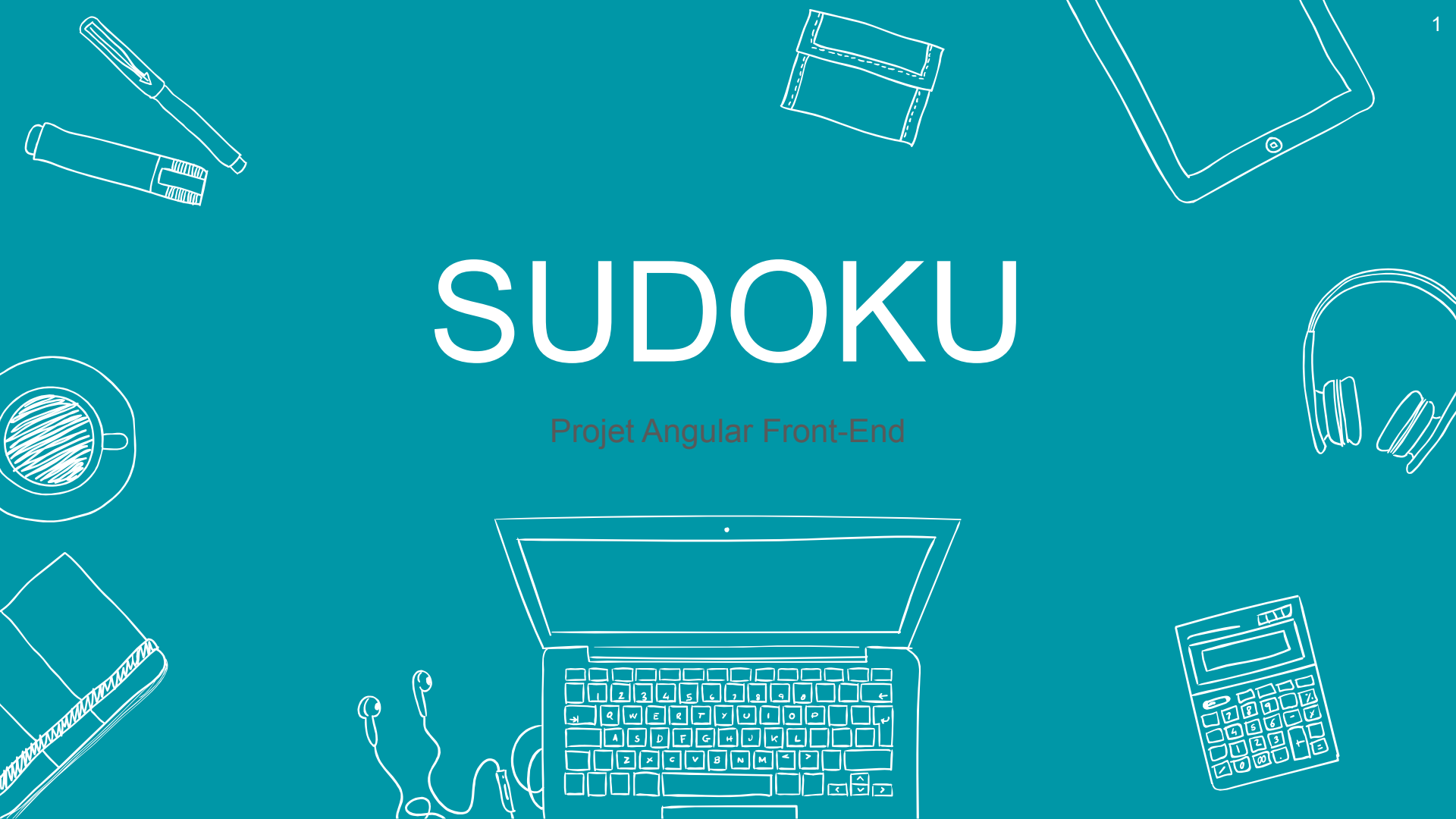


SUDOKU

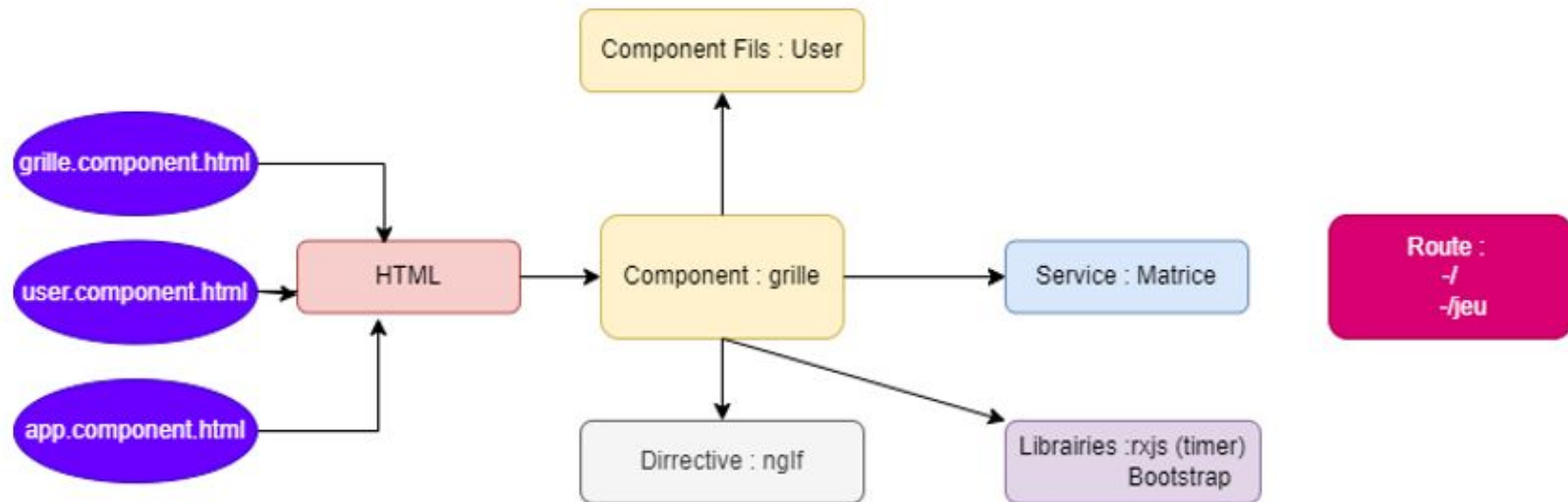
Projet Angular Front-End





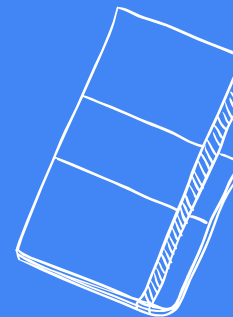
Architecture de notre jeu







La première page pour saisir les joueurs



Saisie des joueurs

L'HTML de la page qui contient un formulaire :

On récupère les variable des users avec la directive ngModel.

```
<form class="Saisie bg-light bg-gradient" #form="ngForm" (ngSubmit)="onSubmit(form)" novalidate>
  <label for="username1">Joueur 1 : </label><br>
  <input type="text" id="saisiUser1" name="username1" ngModel required #first="ngModel"><br><br>
  <label for="username2">Joueur 2 : </label><br>
  <input type="text" id="saisiUser2" name="username2" ngModel required #second="ngModel"><br><br>
  <button type="submit" class="btn btn-primary">Commencer une partie</button>
</form>
```

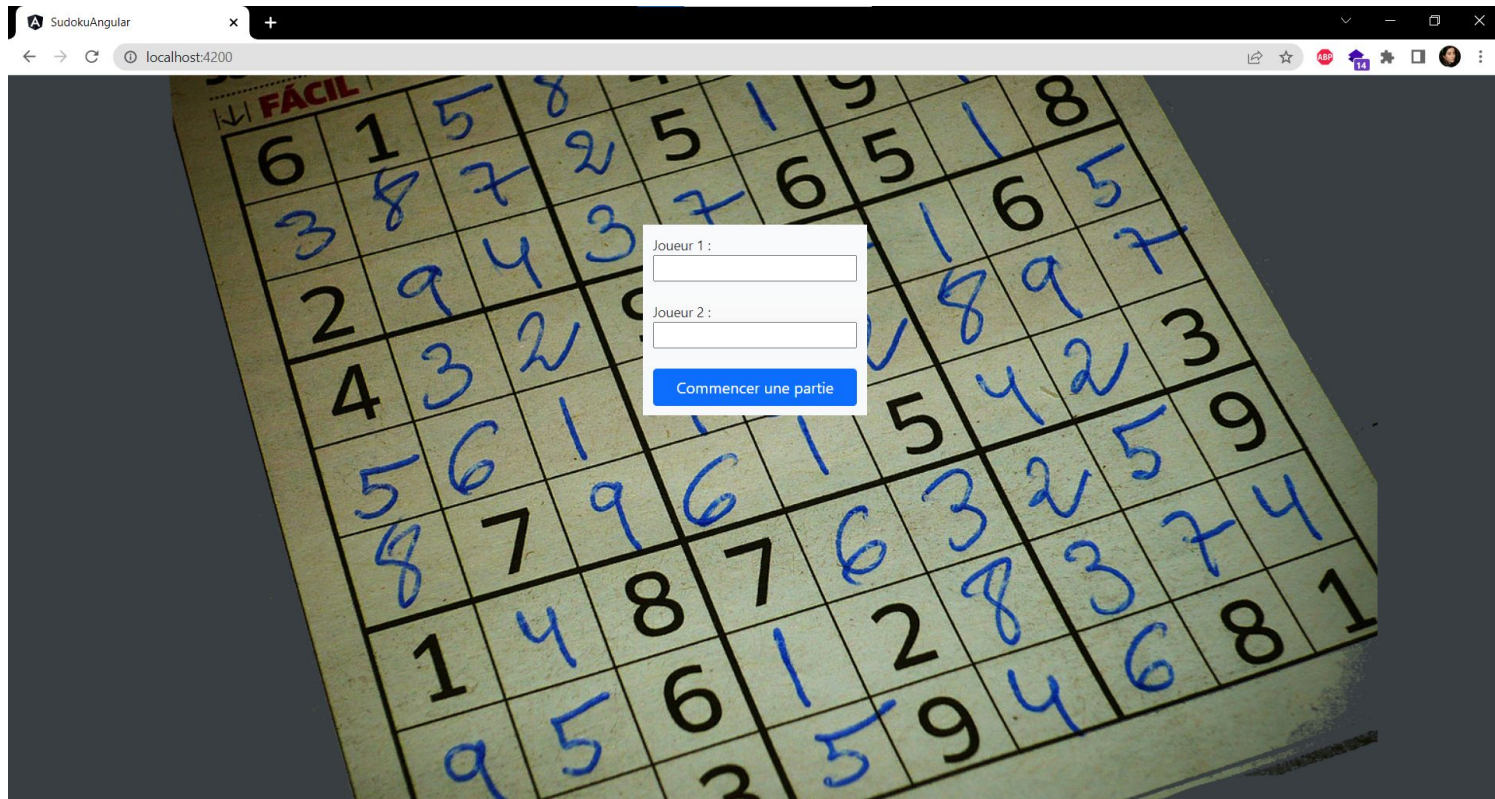
ilham, il y a 3 jours via PR #2 • recup valeur des joueurs (onsubmit) ...

Création d'un component user :

```
▼ user
  # user.component.css
  <> user.component.html
  TS user.component.spec.ts
  TS user.component.ts
```

Saisie des joueurs

Le résultat sur le navigateur <http://localhost:4200/>:



Le jeu



Jeu

En cliquant sur le bouton “Commencer la partie”, on se redirige directement dans la page du jeu du premier joueur :

Création de notre composant grille :

Dans l’html de ce composant grille.component.html, on affiche grille quand notre partie est en cours.



```
<div *ngIf="partieFinie; then finDePartie else lejeu"></div>
<ng-template #finDePartie>
  <div id="finGame"
    class="container bg-light bg-gradient h1 center position-absolute top-50 start-50 translate-middle mt-15">
    {{messageFin}}<br>
    <button (click)="recommencer()">Recommencer</button></div>
</ng-template>
<ng-template #lejeu>
  <div *ngIf="!partieEnCours; then formulaire else jeu"></div>
  <ng-template #formulaire>
    <app-user (sendRequestToFather)="sendData($event)"></app-user>
  </ng-template>
</ng-template>
```


Jeu

Le code HTML de la grille est le suivant :

```
<ng-template #jeu>
  <p></p>
  <div id="lesJoueurs"
    class="container bg-light bg-gradient h1 center position-absolute top-20 start-50 translate-middle mt-5">
    {{joueur1.getNomJoueur()}} VS {{joueur2.getNomJoueur()}}</div>
  <div id="grille"
    class="container bg-light bg-gradient border border-2 border-light position-absolute top-50 start-50 translate-middle">
    <div>C'est le tour de <div *ngIf="!partie1Jouee; then player1 else player2"></div>
      <ng-template #player1><span>{{joueur1.getNomJoueur()}}</span></ng-template>
      <ng-template #player2><span>{{joueur2.getNomJoueur()}}</span></ng-template>
    </div>
    <div>Votre temps : {{minutes}}min et {{temps%60}}s.</div>
    <div class="row center" *ngFor="let obj of matrice; let x = index;">
      <div class="col-sm-auto p-0" *ngFor="let keyValues of obj; let y = index;">
        <input class="default" id="{{ 'case' + x + '-' + y }}" [(ngModel)]="this.matriceService.grille[x][y]"
          (change)="verifNombre(x,y,this.matriceService.grille[x][y])">
      </div>
    </div>
    <div class="row center"><button (click)="verifSolution()"> Vérifier</button></div>
  </div>
</ng-template>
```

On a utiliser la directive ngIf.

Jeu

Génération de la grille :

On a créé une grille en utilisant le service Matrice et on l'a rempli avec une matrice.

```
createMatrice() {  
  this.matrice = [];  
  for (let x = 0; x < 9; x++) {  
    this.matrice[x] = new Array(9);  
  }  
  return this.matrice;  
}
```

```
genererGrille() {  
  this.grille = []  
  for (let x = 0; x < 9; x++) {  
    this.grille[x] = new Array(9);  
    for (let y = 0; y < 9; y++) {  
      if (Math.random() < 0.45) {  
        console.log("yolo");  
        this.grille[x][y] = this.solution[x][y];  
      }  
    }  
  }  
  return this.grille;  
}
```

Jeu

Génération de la grille :

Quand on vérifie notre grille, on passe à la deuxième partie du deuxième joueur et donc on régénère une grille tout en enregistrant le score du premier joueur

```
restartGame() {  
    this.matriceService.grille = this.matriceService.genererGrille();  
    this.partie1Jouee = true;  
    this.temps = 0  
}
```

Jeu

SudokuAngular x +

localhost:4200

Emeline VS Ilham

C'est le tour de Emeline
Votre temps : 0min et 3s.

3	4		1	7			2
5	8		4	2		9	7
			5	6			
8			3	4	2		5
			1				
7			8				
1	8	2					
	5	3	4		7		
			6	8			

Vérifier

Résultat du jeu

Comparaison des scores et affichage du gagnant :

Après que les deux joueurs ont fait leur partie, on compare leurs deux scores en prendre en compte les erreurs qu'ils ont fait (une erreur ajoute sur le temps 60s) et on affiche le joueur qui a moins de timer+erreur.

```
verifSolution() {  
  for (let x = 0; x < 9; x++) {  
    for (let y = 0; y < 9; y++) {  
      if (this.matriceService.grille[x][y] != this.matriceService.solution[x][y]) {  
        this.var = false;  
      }  
    }  
  }  
  
  if (this.var) {  
    console.log("fini !");  
    //if joueur 1 -> Restart game  
    if (this.partie1Jouee == false) {  
      this.joueur1.setScoreJoueur(this.temps + this.erreurs[0] * 60);  
      this.restartGame();  
    }  
    //if joueur 2 -> Fin game  
    else {  
      this.joueur2.setScoreJoueur(this.temps + this.erreurs[1] * 60);  
      this.endGame();  
    }  
  }  
}
```

Résultat du jeu

Notre Timer :

```
startTimer() {  
  this.interval = setInterval(() => {  
    this.temps++;  
    this.minutes = Math.trunc(this.temps / 60);  
  }, 1000)  
}
```

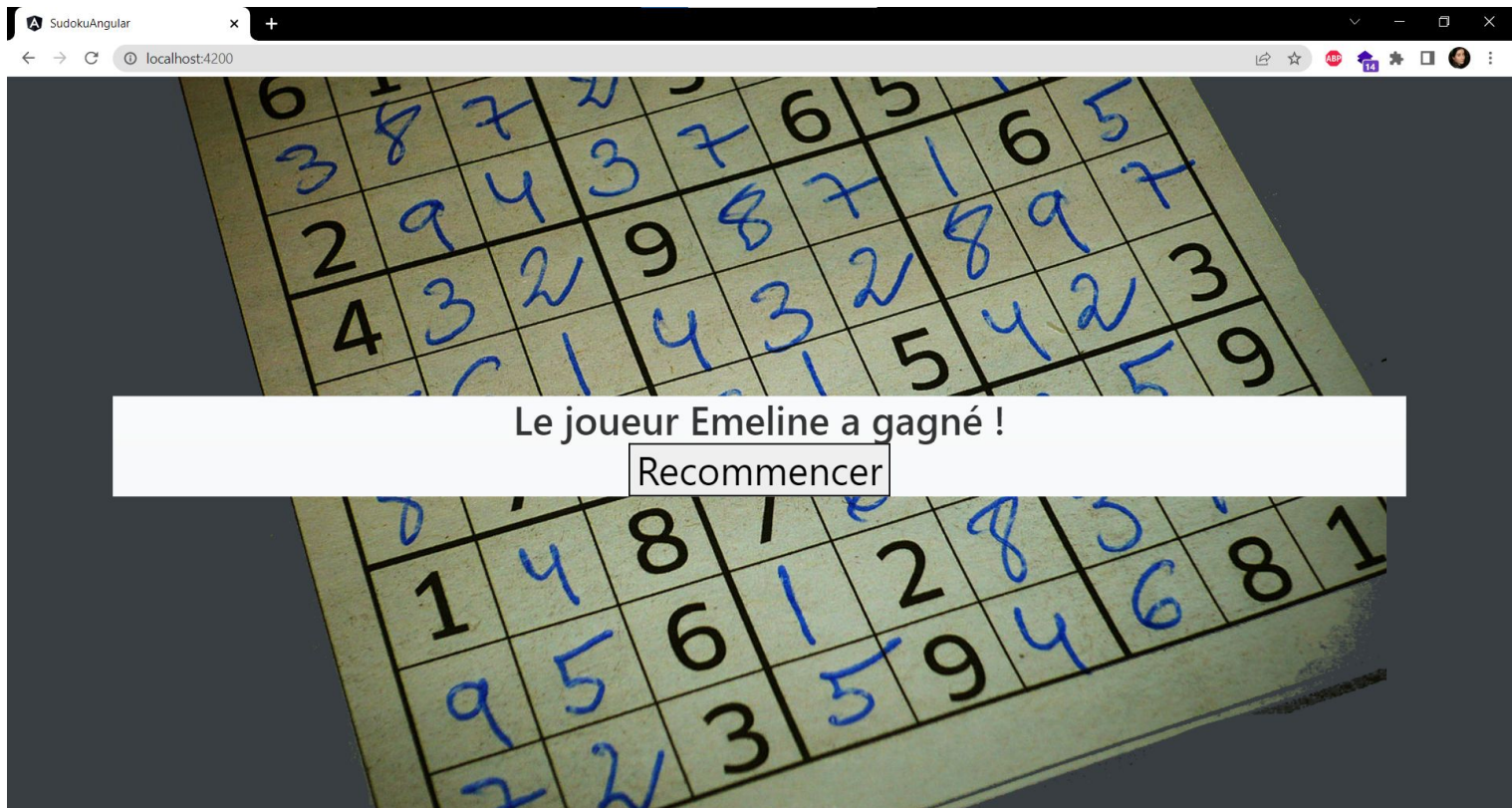
Affichage avec des minutes :

```
<div>Votre temps : {{minutes}}min et {{temps%60}}s.</div>
```

Votre temps : 1min et 12s.

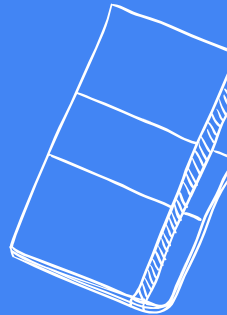
Résultat du jeu

Affichage du gagnant :





Notation et pourcentage de travail



Notation

Répartition de travail :

Emeline SANCHEZ : code 60% documentation : 50%

Ilham ARBOUCH : code 40 % documentation : 50%

Note attribuée à notre projet : 18

-Nouveaux langage et donc notre projet était très bien réussi malgré le très peu de temps qu'on a eu pour l'apprendre.

-Tous les éléments obligatoires pour réaliser ce projet ont étaient respectés (services, directive, component...).

-On a pris en compte l'affichage et on s'est mis à la place de l'utilisateur pour mieux réussir notre sudoku.