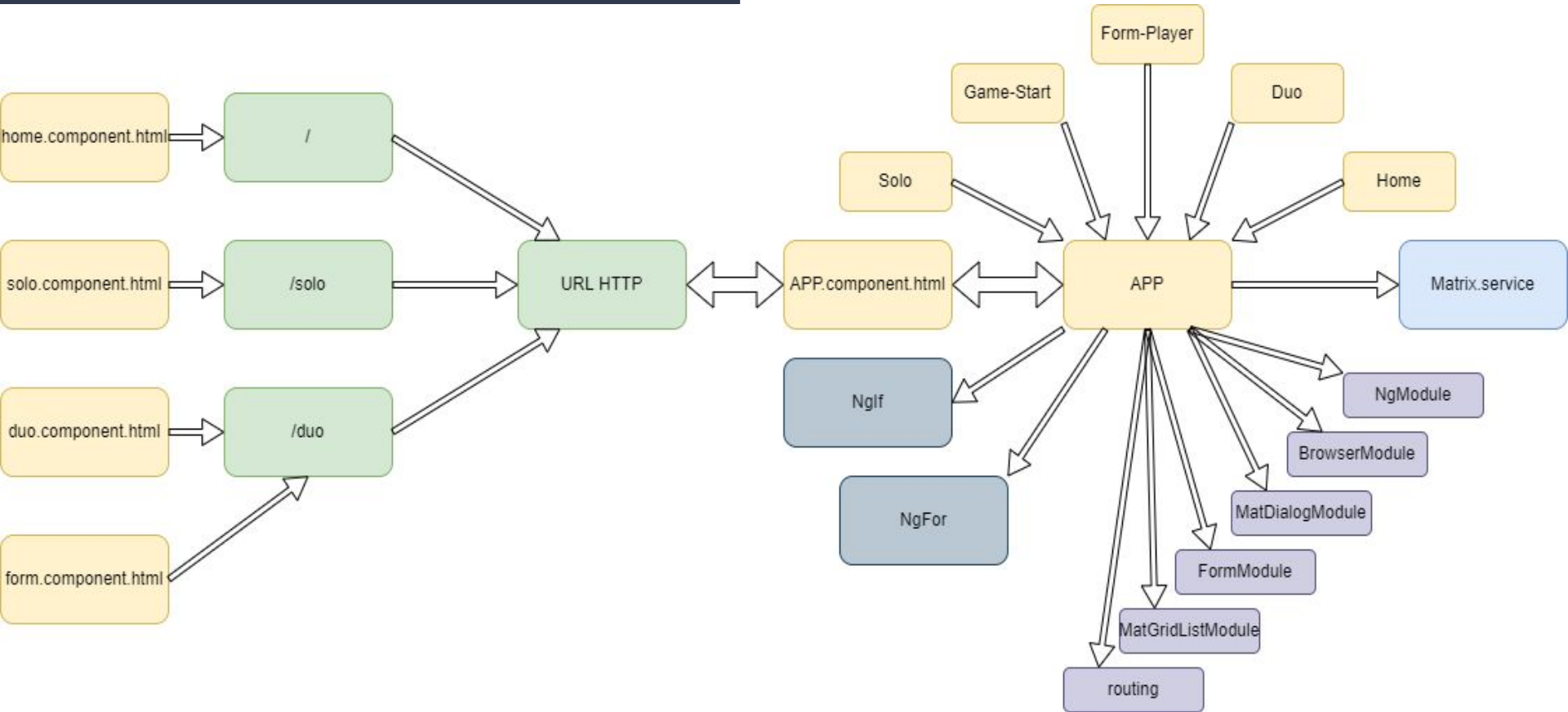
Projet Angular :

MineSweeper

Présentation de l'architecture



Présentation du Code

```
export class MatrixService {
  matrix: number[][] = this.createMatrix();
  coordMine!: number[][];
  constructor() {}

  getCoordMine() {···
  }

  getMatrix(){···
  }

  createMatrix() {···
  }

  checkNeighbors(i: number, j: number, numberChecked: number) {···
  }

  checkNeighborsToClose(i: number, j: number, numberChecked: number) {···
  }

  updateMatrix() {···
  }

  addClass(cardMore: HTMLElement | null, numberChecked: number, i: number, j: number) {···
  }
```

Service Matrix:

- Contient toute les fonctions pour interagir avec nos matrice.

Présentation du Code

Game-start.component.html:

- Contient tout le HTML d'un partie mais surtout la grille du dénimeur crée avec ngFor.

```
<div class="container d-flex w-100 flex-column justify-content-center align-items-center minesweeper">
  <div class="row center" *ngFor="let obj of matrix; let i = index;">
    <div class="col-sm-auto" *ngFor="let keyValues of obj; let j = index;">
      <div class="default" id="{{'card' + i + '-' + j}}" (click)="check(keyValues,i,j)"
        (contextmenu)="flag(keyValues, i, j)"></div>
    </div>
  </div>
</div>
```

Présentation du Code

form–player.component.html :

- Contient le HTML du formulaire utilisé pour récupérer le nom des joueurs.
- On utilise ngForm pour récupérer les valeurs des inputs.

```
<form class="form"  #form="ngForm" (ngSubmit)="onSubmit(form)" novalidate>
    <br>
    <label for="Player1">Name player 1 :</label>
    <br><br>
    <input name="Player1" id="Player1" class="form-control" ngModel required #first="ngModel">
    <br><br>
    <label for="Player2">Name player 2 :</label>
    <br><br>
    <input name="Player2" id="Player2" class="form-control" ngModel required #second="ngModel">
    <br><br>
    <button type="submit" class="btn-primary one block"><b>Play !</b></button>
</form>
```

Présentation du Code

app.routing.ts:

- Les routes de notre application.
- On utilise RouterModule.

app.component.html:

- On utilise router-outlet pour refaire la redirection des pages.

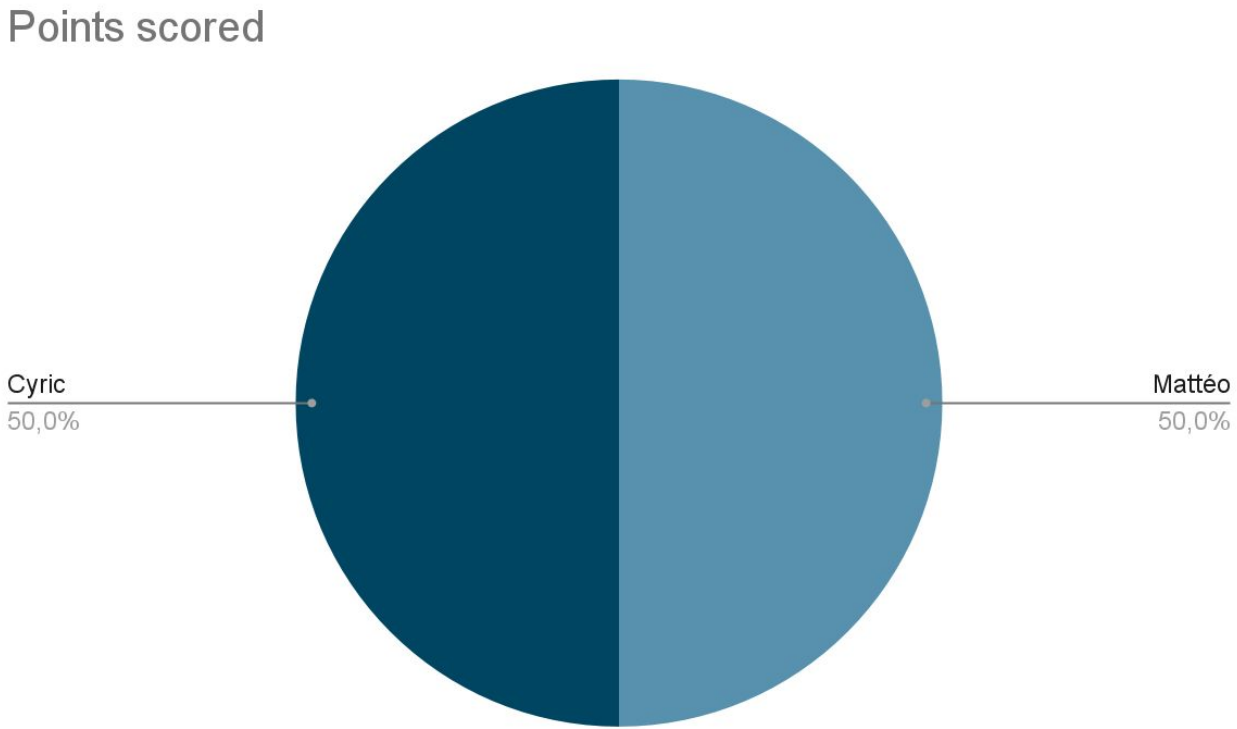
```
const APP_ROUTING: Routes = [
    { path: "", component: HomeComponent },
    { path: "solo", component: SoloComponent },
    { path: "duo", component: DuoComponent },
];

export const routing = RouterModule.forRoot(APP_ROUTING);
```

```
<body>
  <router-outlet></router-outlet>

</body>
```

Pourcentage de répartition du travail



Note du Projet

18