



### 1. AWELE

**Lien** : <https://github.com/laminebarghouda/Awele>

**Explication** : L'awalé ou awélé est un jeu de société combinatoire abstrait créé en Afrique et dans notre contexte, le jeu est purement programmé en Java.

**Langage de Programmation** : Java

**Contribution** : Dans ce projet, on a optimisé l'affichage console de la partie du jeu entre les deux joueurs :

- Affichage des résultats de chaque coup
  1. Tour Machine (IA)
  2. Tour Joueur (Utilisateur)
- Affichage des résultats final du jeu et du gagnant
- Optimisation du code Java et nettoyage

**Exemple du code ajouté :**

```
// Determiner le gagnat de la partie
private static void resultatFinal(Joueur j1, Joueur j2) {
    System.out.println("~~~~~ Scores Finales ~~~~~");
    System.out.println(j1.getNom() + " : " + j1.getScore());
    System.out.println(j1.getNom() + " : " + j2.getScore());
    if (j1.getScore() > j2.getScore())
        System.out.println("IA l'emporte");
    else if (j1.getScore() < j2.getScore())
        System.out.println("Joueur l'emporte");
    else
        System.out.println("Egalité");
    System.out.println("~~~~~");
}
```



## 2. DETECTEUR DE LANGUE

**Lien** : <https://github.com/laminebarghouda/Detecteur-de-Langue>

**Explication** : Ce projet sert à détecter la langue d'un texte (saisie dans un fichier) grâce à une interface graphique.

**Langage de Programmation** : Java

**Contribution** : Dans ce projet, on a ajouté une nouvelle fonctionnalité et c'est la reconnaissance et détection de la langue suédoise.

- Ajout du texte référence de la langue suédoise
- Etendre l'utilisation de l'algorithme de détection qui prendra en charge de la langue suédoise maintenant.
- Modification du code nécessaire pour le test et l'affichage pour cette nouvelle fonctionnalité.

**Exemple du code ajouté :**

```
// Initialisation ArrayList suedois = new ArrayList<>()
// Preparer le texte de comparaison
suedois = compterOccurences(new
File(getClass().getResource("/LanguesDeTest/sv.txt").toURI()));
scores.add(new Double((float) calculPS(texte, suedois) / (texte.size() *
suedois.size())));
// Faire la correspondance du texte à étudier aux langues connus
Double lng = Collections.max(scores);
.....
case 4:
this.langue = "Suedois";
```



### 3. TCP/IP SERVER CLIENT

**Lien** : [https://github.com/laminebarghouda/TCP\\_IP\\_SERVER\\_CLIENT](https://github.com/laminebarghouda/TCP_IP_SERVER_CLIENT)

**Explication** : Ce projet assure une communication entre un serveur et un client grâce au protocole réseau TCP/IP.

**Langage de Programmation** : C++

**Contribution** : Dans ce projet, on a :

- Optimisation de taille des messages échangés entre le client et le serveur.
- Optimisation de gestion des erreurs de création ou la gestion des sockets

**Exemple du code ajouté :**

**/\* Exemple Coté Client \*/**

**// Create a socket**

```
SOCKET listening = socket(AF_INET, SOCK_STREAM, 0);
if (listening == INVALID_SOCKET || listening == ERROR_INVALID_ACCESS)
{
    cerr << "Can't create a socket! Quitting" << endl;
    return ;
}
```

**// Taille du message en buffer**

```
char buf[8192];
```

**// Wait for client to send data**

```
int bytesReceived = recv(clientSocket, buf, 8192, 0);
.....
```

**/\* Exemple Coté Client \*/**

**// Wait for response//**

```
ZeroMemory(buf, 8192);
```

```
int bytesReceived = recv(sock, buf, 8192, 0);
```

```
.....
```

DJAMAA SEFOUANE

<https://github.com/sefouanedjamaa>



### 4. NOTES APP FRONTEND

**Lien** : <https://github.com/guedriOussema/NotesAppFrontend>

**Explication** : Ce projet Permet d'ajouter et supprimer dynamiquement des notes, remarques, à faire etc..

**Langage de Programmation** : TypeScript, Angular Framework

**Contribution** : Dans ce projet, on a :

- Optimisation du design du site web (SCSS)
- Optimisation et Nettoyage du code HTML et ts

**Exemple du code ajouté :**

```
// src/app/pages/main-layout/main-layout.component.scss
.top-bar
{ height: 65px;
background: linear-gradient(to right, $light-red, $blue);
}
h1{
color: white;
font-size: 24px; }
// src/app/app.component.scss
Body{ font-family: 'Roboto';
}
// src/app/app-routing.module.ts
{path: '', component: MainLayoutComponent,
children:[
{path:'', component: NotesListComponent},
{path:'new', component: NoteDetailsComponent},
{path:'id', component: NoteDetailsComponent}
]
}}
```

DJAMAA SEFOUANE

<https://github.com/sefouanedjamaa>



### 5. VIDLY NODE API

**Lien** : <https://github.com/guedriOussema/vidly-node-api>

**Explication** : Ce projet est une movies API

**Langage de Programmation** : JavaScript, Node Js, Express Js

**Contribution** : Dans ce projet, on a :

- Centralisation de routes de l'API
- Optimisation de L'index code de l'application

**Exemple du code ajouté :**

```
// vidly-node-api/routes/index.js
const express = require('express');
const genres = require('./genres');
const customers = require('./customers');
const movies = require('./movies');
const rentals = require('./rentals');
const users = require('./users');
const auth = require('./auth');
const router = express.Router();

router.use('/api/genres', genres);
router.use('/api/customers', customers);
router.use('/api/movies', movies);
router.use('/api/rentals', rentals);
router.use('/api/users', users);
router.use('/api/auth', auth);

module.exports = router;
```

```
// vidly-node-api/index.js
const routes = require('./routes/v1');
app.use('/', routes);
```

DJAMAA SEFOUANE

<https://github.com/sefouanedjamaa>