

# Projet Bourbaki

## Principe du jeu

[https://fr.wikipedia.org/wiki/La\\_Pipopipette](https://fr.wikipedia.org/wiki/La_Pipopipette)

La Pipopipette ou « jeu des petits carrés » est un jeu de société au tour par tour.

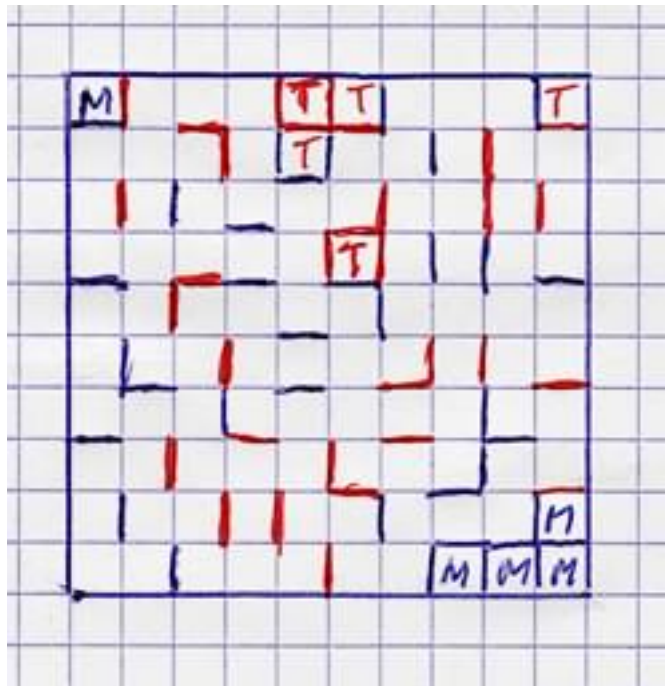


Figure 1:

Le jeu se joue généralement avec papier et crayon sur du papier quadrillé. À chaque tour, chaque joueur trace un petit trait suivant le quadrillage de la feuille. Le but du jeu est de former des carrés. Le gagnant est celui qui a fermé le plus de carrés.

## Fonctionnalités

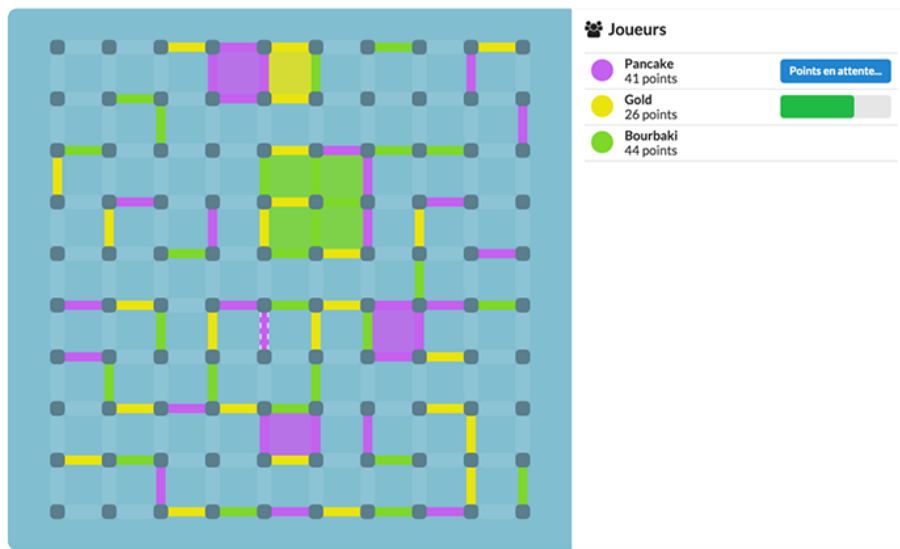
- Connexion / Inscription rapide
- Partie unique à laquelle des joueurs peuvent se connecter à tout moment
- Gestion des connexions / déconnexions en pleine partie
- Temps limité pour jouer
- IA en cas de déconnexion d'un joueur ou de temps écoulé

- Calcul et affichage des scores en temps réel
- Classement général avec cumul des points gagnés

*Le maximum du développement sera déporté sur le serveur, idéalement le client se tâchera de faire uniquement de l'affichage.*

Détail complet des fonctionnalités : <https://gitlab.info-ufr.univ-montp2.fr/HMIN302/go-bourbaki/blob/master/doc/features.md>

## Démonstration en ligne



La dernière version du jeu est en ligne publiquement sur <http://bourbaki.doelia.fr:2000>

## Installation et utilisation

Récupérer le projet dans le src/ du GOPATH :

```
cd $GOPATH/src
git clone https://gitlab.info-ufr.univ-montp2.fr/HMIN302/go-bourbaki.git
```

Dépendances :

```
go get github.com/boltdb/bolt/
go get github.com/googollee/go-socket.io
```

Compilation et installation :

```
cd $GOPATH/src/go-bourbaki/server
go install
```

Lancement du serveur :

```
server -port 2000
```

Ce qui ouvre un serveur web à l'adresse <http://localhost:2000>

## Technologies utilisées

- Serveur GO
  - Envoi des ressources via HTTP
  - Sauvegarde des données persistantes en [Bolt](#) (système clé/valeur)
- Client Web
  - Framework [Semantic-ui](#)
  - Librairie [jQuery](#)
  - Bibliothèque d'icônes [Font Awesome](#)
  - Pré-processeur [Less](#)
- Communication via WebSockets avec la librairie [Socket.io](#)
  - Implémentation GO : <https://github.com/googollee/go-socket.io>

## Documentation technique

- Documentation Go du projet générée : <http://bourbaki-doc.doelia.fr/pkg/go-bourbaki/>
- [Protocole](#) (liste des paquets)
- [Règles du jeu](#)
- [Ébauche diagramme UML](#)

## Développement

Pour modifier le css, installer le package [lessc](#) avec [npm](#) :

```
sudo npm install -g less
```

Plugin de compilation automatique avec Atom : <https://atom.io/packages/less-autocompile>.

## Difficultés rencontrées

- Problèmes avec l'usage de socket.io avec GO
  - La déclaration de paquets sans paramètre entraine la désactivation d'autres paquets (la déconnexion par exemple)
- Les nombreux événements modifiant le déroulement du jeu (déconnexion d'un joueur, déclenchement du mode pause, chronos...) génèrent souvent des conflits entre eux. Il est courant de tomber dans des appels récursifs ou dans un blocage total de la partie sans une étude rigoureuse
- Remarques sur GO :
- L'usage des majuscles/minuscules pour le concept de public/privé est pénible en cas de refactoring
- Les dépendances cycliques : pas de solution propre trouvée
  - Recherches : <https://groups.google.com/forum/#!topic/golang-nuts/Lj4RD3SLg6M>