

# Dactylo Game

---

## Single-player

### Mode normal

1. mots apparaissent à l'écran (donc GUI) et joueur est censé recopier au clavier sans faire d'erreur.
2. mots peuvent être aléatoires ou bien un texte cohérent (issu de la littérature, d'un discours, etc. . .)
3. pendant session, il y aura des indications graphiques qui indiquent la situation du joueur (prochain mots à taper; erreur signalées, etc . . .)
4. utilisation d'un pile pour gérer les mots à taper.
5. Session est delimitée par un temps ou par un nombre de mots à taper.
6. Affichage des statistiques

### Mode jeu

1. pour chaque mot validé, compter erreur (distance d'édition)
  1. une vie retirée pour chaque erreur
2. on monte d'un niveau tous les 100 mots tapés correctement
  1. la vitesse change après chaque niveau
3. les mots arrivent à une vitesse imposée par le jeu (?). Ainsi la file des prochains mots est alimentée par un nouveau mot tous les  $f(n)$  seconds.  $f()$  est une fonction décroissante du niveau courant(?).
4. le fait de valider un mot n'ajoute pas forcément un mot (contrairement au mode normal). On ne le fait que si la file n'était pas au moins à moitié remplie.
5. mots speciaux, en bleu, si tapé correctement dans le premier coup, joueur gagne  $n$  vies où  $n$  correspond à la taille du mot.

## Mode multiplayer

1. en réseau (2 joueurs ou plus)
2. Similaire au mode solo mais mots ne sont pas ajoutés à la file à cause du chronomètre mais plutôt par l'adversaire.
3. mots en rouge sont des pièges pour l'adversaire, si bien tapé, sera envoyé à la pile de toutes adversaire (sans la couleur).
  1. si file est remplie, la validation du mot en cours est forcée si non perte de vies.
4. Un joueur qui n'a plus de vies est éliminé, le dernier joueur gagne la partie.
5. Ici, dans multiplayer il n'y a pas de notion de niveau, cela est imposé par l'adversaire.

## Contraintes techniques

1. statistiques (voir pdf du projet)
2. réseau
  1. client serveur
  2. concurrence
3. serializable
4. tests unitaires

5. focus sur qualité du code et de l'architecture
6. gui simple et pratique
7. maven ou gradle
8. Connection TCP

## Criteres d'evaluation

1. fichier readme
2. diagrammes de classes
3. convention de codage
4. bonne architecture
5. javadoc
6. tests fournie
7. 3 modes de jeu