

# Projet PC2R 2013-2014

## iSketch

version du 01/12/13

Ce projet consiste à implanter une variante simplifiée du jeu *iSketch*.

## 1 Présentation du jeu iSketch

Ce jeu se déroule de la façon suivante : chaque joueur à son tour cherche à faire deviner un mot caché aux autres joueurs en le dessinant, cela en un temps limité.

Chaque joueur a deux rôles dans la partie :

- dessinateur : il doit par ses dessins faire deviner par le maximum de joueurs le mot caché qu'il est seul à connaître, cela en un temps limité ;
- chercheur : il faut trouver dans le temps imparti le mot caché que le dessinateur essaie de faire deviner sachant que les propositions des autres joueurs sont visibles à tous (à l'exception du mot caché trouvé).

Quand un chercheur trouve le mot caché, les autres joueurs ont un temps maximum pour trouver (en général 30 secondes).

Bien sûr pour le dessin, il est interdit d'écrire des lettres ou de faire des rébus, seul le dessin doit permettre de trouver le mot caché.

Les scores des joueurs sont calculés de la manière suivante :

- le premier qui trouve le mot caché gagne 10 points, le suivant 9 points, cela ainsi de suite jusqu'au minimum de 5 points ;
- le dessinateur gagne 10 points si au moins un joueur trouve le mot puis 1 point supplémentaire par autre joueur qui le trouve jusqu'au score maximum de 15 points.

Un tour se termine quand :

- le dessinateur décide de passer son tour ;
- le dessinateur arrête le tour en cours ;
- le temps limité pour trouver est fini et personne n'a trouvé le mot caché ;
- le temps limité après découverte du mot est fini ;
- tous les chercheurs ont trouvé le mot caché ;
- si plusieurs joueurs (au moins 3) décident que le dessinateur transgresse les règles (lettres, rébus, ...).

## 2 Travail à réaliser

1. Programmer un serveur et un client dans deux langages différents, communiquant par le protocole décrit dans ce document. Le serveur gèrera la connexion des joueurs, le démarrage de la partie, les connexions possibles des joueurs en cours de partie, le choix du dessinateur (chacun son tour), le score à chaque tour, la transmission du dessin de chaque dessinateur, la diffusion des propositions de chaque joueur chercheur à l'ensemble des joueurs (sauf le mot caché trouvé). Le client gèrera la demande de connexion au serveur, l'attente du début de partie, pour le rôle de dessinateur l'envoi de la description du dessin au fur et à mesure de sa construction et pour le rôle de chercheur l'affichage du dessin transmis par le serveur, l'affichage des propositions des joueurs chercheurs (sauf le mot caché trouvé), l'envoi de propositions, l'indication qu'une proposition est la bonne.
2. Rédiger un petit rapport de 4 à 8 pages, avec une limite de deux captures d'écran et des marges et tailles de fonte raisonnables.

## 3 Protocole

Le serveur écoute sur le port 2013 en TCP. Les deux parties parlent un protocole texte qui prend la forme d'une succession de commande. Chaque commande est composée d'une suite de chaînes terminées par des caractères '/'. La dernière chaîne est suivie d'un retour chariot Unix \n. La première chaîne est le nom de la commande, les autres ses arguments. Chaque nom de commande n'a qu'un seul rôle dans le protocole (le même nom de commande ne peut signifier deux choses différentes à deux moments différents). La suite '\\/' dans un argument est utilisé pour échapper le caractère '/', et '\\\\' le caractère '\\'.

**Conseil :** tirez parti des possibilités des langages utilisés pour rendre automatique et sûre la lecture et l'interprétation des commandes.

### 3.1 Connexion/Déconnexion

Lorsque le client se connecte au serveur il envoie sans attendre la commande `CONNECT/user/` suivie du nom du joueur précisé par l'utilisateur. Le serveur répond par une commande `WELCOME` suivie du nom du joueur (qui peut être un peu différent dans le cas où plusieurs joueurs auraient demandé le même nom) ainsi qu'une commande `CONNECTED` suivie du nom du joueur à tous les joueurs (y compris lui-même).

Les parties se jouent de 2 à *MAX* joueurs. La partie démarre à la connexion du *MAX*ème joueur.

`CONNECT/user/`

(C->S) Nouvelle connexion de "user"

`WELCOME/user/`

(S->C) Signifie au joueur qui a demandé la connexion que celle-ci est acceptée sous le nom "user"

`CONNECTED/user/`

(S->C) Signifie à tous les clients la connexion de "user"

Il peut avoir des déconnexions d'utilisateur en cours de partie : soit si le joueur l'indique explicitement, soit en cas de fermeture de la communication entre un client et un serveur.

`EXIT/user/`

(C->S) Déconnexion de "user"

`EXITED/user/`

(S->C) Signifie à tous les clients le départ de "user"

(+ Gérer le cas où le "exited" concerne le dessinateur)

### 3.2 Déroulement de la partie

Une partie correspond à *MAX* rounds, permettant ainsi à chaque joueur d'être une fois dessinateur.

**Gestion d'un round :**

`NEW_ROUND/dessinateur/user/mot/`

(S->C) Le nouveau round démarre avec l'indication de qui est le dessinateur à ce tour, auquel on ajoute le mot à dessiner pour le client dessinateur

`GUESS/mot/`

(C->S) Mot proposé par un client

`GUESSED/mot/user/`

(S->C) Signifie aux clients le mot proposé par un autre

`WORD_FOUND/joueur/`

(S->C) Signifie que le client joueur a trouvé le mot (en supprimant l'output du mot sur les autres clients)

`WORD_FOUND_TIMEOUT/timeout/`

(S->C) Signifie aux autres clients que le timeout de la fin du round est enclenché suite à un premier joueur ayant trouvé le mot caché

`END_ROUND/user/mot/`

(S->C) Signifie aux clients que le tour est fini avec nom du vainqueur et le vrai mot à trouver

`SCORE_ROUND/user1/score1/users2/score2/.../userN/scoreN/`

(S->C) Donne le score de tous les joueurs

## Arrêt exceptionnelle d'un round :

PASS

(C->S) le dessinateur décide d'arrêter de dessiner en envoyant la commande PASS au serveur; cette commande produit un effet seulement si aucun joueur n'a déjà trouvé le mot caché, sinon le round s'arrêtera à la fin du temps limite. Personne ne gagne de points à ce tour.

même effet en cas de déconnexion du dessinateur (EXIT/user/ ou déconnexion brutale).

CHEAT/user/

(C->S) un client (sauf le dessinateur) trouve que le dessinateur ne respecte pas l'esprit du jeu et l'indique au serveur; si au moins  $n$  joueurs trouvent qu'un dessinateur n'est pas correct, ce dernier est alors sorti du jeu et personne ne gagne des points à ce tour même si le mot avait été trouvé. Seul le dessinateur peut être sanctionné durant son tour.

$n$  peut être fixé comme paramètre du serveur, par défaut il est de 3.

## Dessin :

SET\_COLOR/r/g/b/

(C->S) du dessinateur au serveur indiquant un changement de couleur

SET\_SIZE/s/

(C->S) du dessinateur au serveur indiquant un changement de taille de trait

SET\_LINE/x1/y1/x2/y2/

(C->S) du dessinateur au serveur indiquant le tracé d'un segment

LINE/x1/y1/x2/y2/r/g/b/s/

(S->C) du serveur à tous indiquant un segment à dessiner de  $(x1,y1)$  à  $(x2,y2)$  de la couleur  $(r,g,b)$  de taille  $s$

## 3.3 Fin de partie

Lorsqu'un joueur se déconnecte, tous les autres joueurs sont prévenus. Si c'était un chercheur, les autres joueurs sont prévenus de son départ. Si c'était le dessinateur, s'il y avait déjà au moins un joueur ayant trouvé le mot on attend la fin du round (temps limite après une découverte du mot caché), et sinon on indique la fin du round à tous les joueurs sans qu'il y ait le moindre gain de points.

La fin de partie arrive lorsque tous les rounds prévus ont été joués ou quand il ne reste plus qu'un seul joueur.

## 3.4 Lancement du serveur et des clients

Le serveur prendra en paramètre le nombre de joueurs maximal, la durée du timeout une fois le mot trouvé et le dictionnaire à passer de la manière suivante :

```
./serveur -max n -timeout t -port p -dico filename
```

par défaut :  $n$  vaut 4,  $t$  30 secondes, *port* 2013 et l'on testera avec le dictionnaire indiqué sur la page Web où l'on trouve un mot par ligne.

Le client prendra en paramètre le numéro de port pour se connecter au serveur et le nom du joueur :

```
./client -port p -user name
```

par défaut le port vaut 2013 et *name* pc2r.

## 4 Extensions

Ces extensions sont à réaliser une fois que le noyau du jeu fonctionne. Elles seront prises en compte pour l'évaluation de ce devoir de programmation.

### 1. Dessins enrichis :

On cherche à pouvoir dessiner des courbes, pour cela on implantera les courbes de Bézier<sup>1</sup> pour la partie dessin les points de contrôle en transmettant quatre points de contrôle

`COURBE/x1/y1/x2/y2/x3/y3/x4/y4/r/g/b/s/`

(S->C) du serveur à tous indiquant une courbe de Bézier à partir des quatre points de contrôle : (x1,y1), (x2,y2), (x3,y3), et (x4,y4) de la couleur (r,g,b) de taille s

### 2. Discussion instantanée :

Le client doit afficher une petite zone de discussion instantanée façon IRC. Lorsque le joueur a écrit son message et l'envoi, le client envoie une commande `TALK` au serveur suivie du texte : `TALK/texte/`. Le serveur transmet alors à tous les clients la commande `LISTEN` suivie du nom du joueur puis du message : `LISTEN/joueur/texte/`. Chaque client doit alors afficher le message dans sa console de discussion.

**Note :** le serveur peut envoyer lui-même des messages (par exemple en utilisant le nom "(broadcast)"). Cela peut être utile pour ajouter des décomptes ou des commentaires.

### 3. Comptes utilisateurs :

En plus du mode anonyme, on veut permettre à un joueur de réserver son nom, en protégeant son utilisation par un mot de passe.

Pour ceci, on autorise deux commandes de connexion supplémentaires :

– `REGISTER/n/p/` se connecte sous le nom *n* et réserve ce nom pour de futures connexions, en le protégeant par le mot de passe *p*. Si l'utilisateur existe déjà dans la base, le client reçoit une commande `ACCESSDENIED` et la connexion est terminée.

– `LOGIN/n/p/` se connecte sous le nom *n*, qui doit être enregistré dans la base d'utilisateurs et dont le mot de passe doit être *p*. Si l'utilisateur n'existe pas ou si le mot de passe ne correspond pas, le client reçoit une commande `ACCESSDENIED` et la connexion est terminée.

**Notes :** le choix d'implantation de la base d'utilisateurs est laissée libre (fichier texte, base SQLite, donnée sérialisée, etc.).

En option, vous pouvez établir un protocole plus sûr que la transmission du mot de passe en clair.

### 4. Mode Spectateur :

On rajoute une commande de connexion `SPECTATOR`. Un spectateur n'intervient pas dans la partie mais peut voir toutes les actions de tous les joueurs. Le serveur utilise pour cela le jeu de commandes suivantes :

Si un spectateur arrive en cours de partie, le serveur doit lui renvoyer à ce moment toutes les commandes qu'ils auraient reçues s'il était arrivé au tout début afin de rattraper son retard.

### 5. Serveur HTTP de statistiques :

On veut, en accédant au port 2092 de la machine où est lancé le serveur depuis un navigateur Web, obtenir une page HTML indiquant pour chaque joueur inscrit dans la base de données son nombre de victoires et son nombre de défaites.

**Indication :** pour vous éviter la lecture de la spécification du protocole HTTP, interrogez quelques sites Web directement avec la commande telnet pour vous donner une première idée de ce que doit renvoyer votre serveur.

---

1. wikipedia : [http://fr.wikipedia.org/wiki/Courbe\\_de\\_B%C3%A9zier](http://fr.wikipedia.org/wiki/Courbe_de_B%C3%A9zier)

cours P7 : <http://www.liafa.jussieu.fr/~carton/Enseignement/InterfacesGraphiques/MasterInfo/Cours/Swing/splines.html>