UNIVERSITÉ
PARIS-EST
MARNE-LA-VALLÉE

Three to go !

Programmation C – Projet 3ème semestre

Samy WADAN
Fabien LAMBERT

TP6
TP3

L2 Maths / Info
2017 - 2018

Table des matières

Présentation du programme

Description

Les tokens sont des motifs caractérisés par une forme et une couleur. Dans ce programme, il existe 4 couleurs (rouge, vert, bleu, jaune) et 4 formes (cercle, carré, triangle, diamant).

Un token est généré aléatoirement et le joueur doit le placer sur une ligne, il peut décider de le placer à gauche ou à droite de la file.

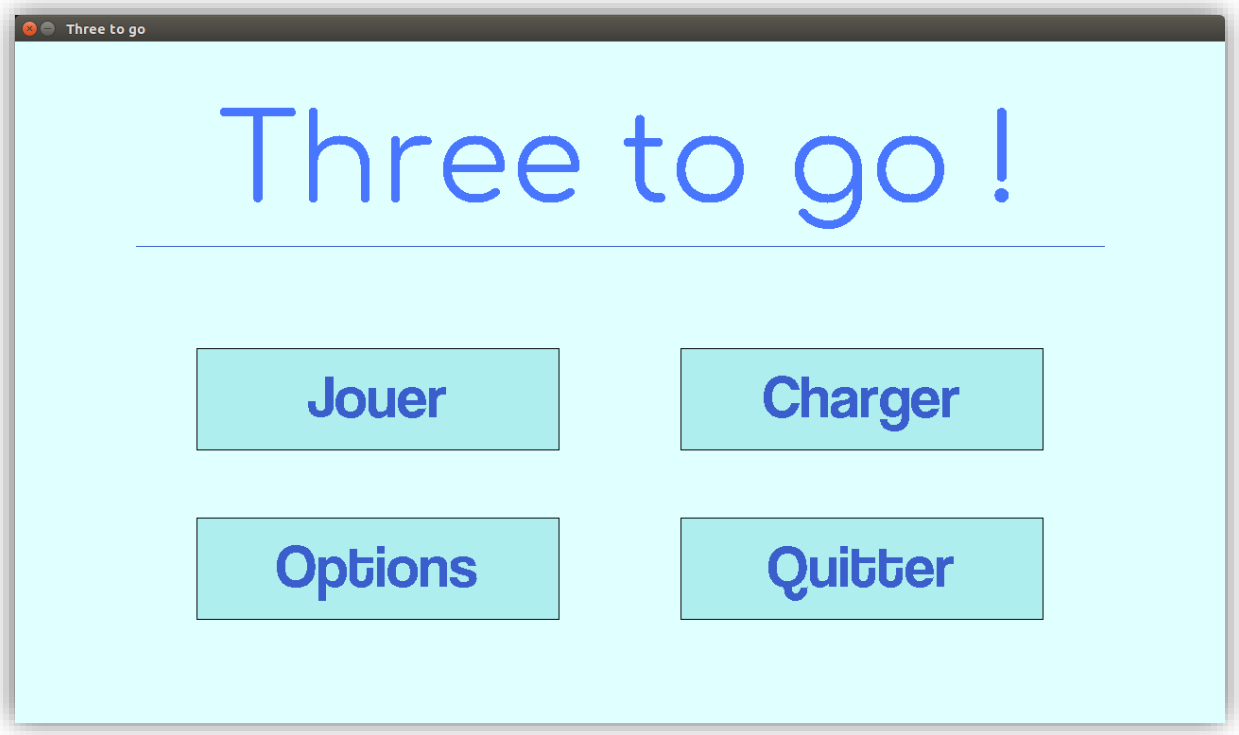
Si 3 tokens ou plus successifs ont la même couleur ou la même forme, ils sont retirés de la file, le joueur marque des points en les retirant. Le début et la fin de la liste sont liés, ainsi 2 tokens bleus à la fin et 1 token bleu au début sont supprimés.

De plus, le joueur peut sélectionner un token et effectuer vers la gauche un décalage circulaire des formes ou des couleurs correspondant au token sélectionné.

L'objectif est de cumuler le plus de points dans un délais de 2 minutes.

Composition

Dès le lancement du programme, le joueur est dirigé vers un menu d'accueil, à partir de là il peut choisir entre : jouer, charger une partie, régler des options et quitter.



Les options

Les options proposées permettent de modifier la taille de la fenêtre, 3 résolutions sont disponibles : *640*360*, *1280*720* et *1920*1080*. Il est également possible d'afficher la fenêtre en plein écran.

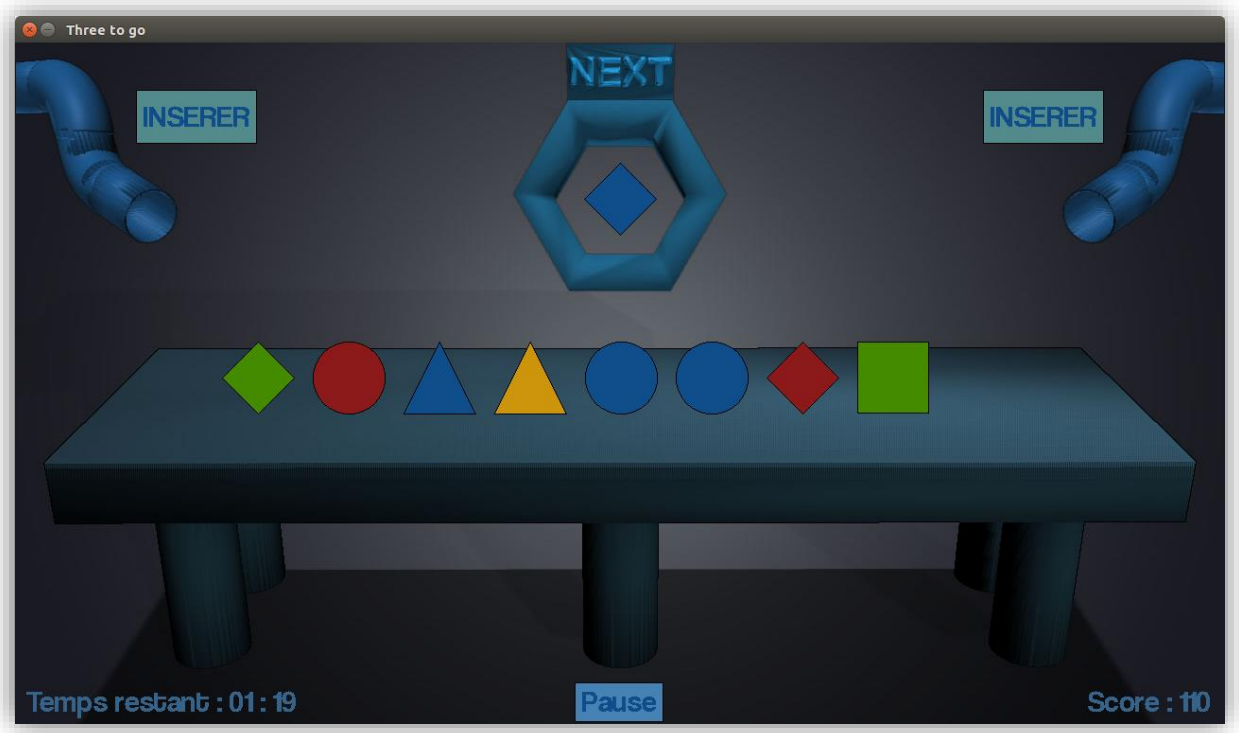
Charger une partie

Charger une partie permet de lancer le jeu avec la progression enregistrée antérieurement par le joueur dans un fichier texte.

Le jeu

Le token généré aléatoirement est centré en haut de l'écran. Avec la souris le joueur doit cliquer sur un des deux boutons « INSERER » pour ajouter ce token à gauche ou à droite de la file. Lorsqu'il clique sur un token de la file, un bouton au-dessus et en-dessous de celui-ci s'affiche, ils permettent d'effectuer les décalages circulaires des couleurs et des formes.

A chaque clique le temps restant est affiché en bas à gauche et le score en bas à droite. Au bout de 2 minutes, même si le joueur ne fait rien, la partie s'arrête.
À tout moment le joueur peut cliquer sur « PAUSE » ce qui aura pour effet d'afficher un menu permettant de : reprendre la partie, la sauvegarder, ou la quitter. Pendant ce temps-là, le temps ne s'écoule pas.



Ecran de fin

Lorsque la partie s'achève, c'est-à-dire au bout de 2 minutes ou lorsque le joueur décide de le faire, le joueur est dirigé vers un écran de fin de partie, affichant le score obtenu et proposant 2 choix : rejouer une nouvelle partie ou retourner à l'écran d'accueil.
Le score est calculé comme tel :

- 3 tokens retirés : `+10 * combo`
- 4 tokens retirés : `+30 * combo`
- 5 tokens retirés : `+70 * combo`

`combo` étant le nombre de délétion précédente durant la même action.

Construction du code

Organisation

Pour plus de facilité, nous avons choisis d'héberger nos fichiers sur une plateforme qui sauvegarde l'historique des modifications et qui nous permet de travailler en même temps sur un fichier, le site est accessible en cliquant ici.

Suivant l'énoncé, nous avons commencé par réaliser le niveau 1, puis le niveau 2, puis le niveau 3, enfin nous avons ajouté quelques fonctionnalités telles que la modification de la taille de l'écran ou la sauvegarde d'une partie. Nous avons également pris le temps de soigner l'aspect visuel avec notamment des menus et des animations.

Création de la liste des tokens

Suivant la structure imposée modélisant les tokens, nous avons commencé par créer une fonction `Token* allouer_token()` dont le token suivant et précédent est lui-même. Ensuite, les fonctions `int ajouter_a_gauche()` et `int ajouter_a_droite()` permettent de remplir la liste en ajoutant un token dont on connais la couleur et forme. Après leur ajout, la fonction `void actualise_chainage()` est appelée pour établir les précédents et suivants forme/couleur de chaque token.

Nous avons utilisé le chainage simple (token.suivant) pour parcourir la liste et le chainage double pour déterminer les précédent/suivant couleur/forme

Décalages circulaires

Avec l'interface graphique uniquement, lorsqu'un joueur clique sur un token, il peut effectuer un décalage circulaire vers la gauche des couleurs ou des formes. C'est le but des fonctions `int decalage_circulaire_couleur()` et `int decalage_circulaire_forme()`.

Suivant le type de décalage, ces fonctions ne regardent que les suivants et précédents du type et décalent les tokens vers la gauche de la même façon qu'un simple tableau.

A la fin, elles appellent la fonction `void actualise_chainage()` pour redéterminer les précédents/suivants couleur/forme.

Suppression des tokens successifs ayant un critère en commun

A chaque ajout de token ou modification avec les décalages circulaires, la fonction `int verif_max_consecutives()` est appelée pour détecter s'il y a ou non minimum 3 tokens successifs ayant un critère en commun et si c'est le cas, elle renvoie par return le nombre de tokens validant la condition et par adresse l'adresse de ces tokens.

Si la valeur retournée est supérieure ou égale à 3, les tokens sont extraient de la liste avec `Token* extrait_token()` et sont libérés avec la fonction de base `free()`.

Sauvegarde et récupération des données

Lorsque le joueur, entre dans le menu pause, il peut sauvegarder sa partie. Cela a pour effet d'appeler `void sauvegarder()` qui enregistre dans un fichier *sauvegarde.txt* le nombre de tokens présent dans la liste, le temps écoulé, le score, le prochain token, et les tokens composants la liste.

Par la suite, le joueur peut décider de charger ces données, appelant `int charger()` qui aura pour conséquence de lancer une nouvelle partie avec les données enregistrées dans *sauvegarde.txt*.

Graphismes

Les graphismes ont leur part importante dans le projet.
Premièrement, chaque élément est disposé en fonction de la taille de la fenêtre, ce qui permet de modifier la taille de celle-ci sans changer leur disposition, à condition

de respecter la proportion, c'est pourquoi les tailles proposées dans le menu des options n'est pas libre à toutes dimensions.

Dans l'écran de jeu, un fond est présent, c'est une image créée personnellement qu'on inclut avec la bibliothèque MLV. Bien sûr cette image ainsi que les polices d'écriture utilisées sont libérées à chaque fin de leur utilisation.

Autre point, des animations apparaissent lorsqu'on affiche le token suivant, qu'on en ajoute un ou quand on effectue un décalage circulaire. Cependant cette partie est sensible car la plupart du temps il n'y a pas de fonction d'attente telle que `MLV_wait_milliseconds()`, la complexité de la fonction la remplace. En effet pour animer un token on l'affiche, puis on efface la zone avec `MLV_draw_partial_image()`, et cela à chaque pixel. En conséquence l'animation est rapide quand la taille de la fenêtre est petite et plus lente quand celle-ci augmente.