ELC D3 - Application Web

Rapport de projet informatique

BE - Breach

Groupe ELC D3
Saban Suresh
Adrien Tan

Chargés
Daniel Muller
René Chalon

07/04/21

1 Introduction

1.1 Objectif

Il s'agissait ici de mettre en pratique nos connaissances, notamment en HTML, CSS et Javascript, afin de réaliser une application de jeu multijoueurs. Cela passera par l'exploitation de l'élément <canvas> et du module socket.io qui permet la communication de données entre les utilisateurs.

1.2 Langues utilisées

Par habitude et dans un but de professionnalisation, nos noms de variables ou de fonctions sont en anglais. Cependant, afin de faciliter la compréhension dans le cadre de ce cours, nous avons décidé de conserver les commentaires en français.

1.3 Comment jouer ?

Après avoir téléchargé les différents fichiers, accéder au dossier backend du dossier Breach à l'aide de l'invite de commandes. Une fois cette opération effectuée, tapez "node server.js" ou bien "npx nodemon server.js" pour lancer le serveur. Il vous suffit désormais d'ouvrir le fichier index.html dans votre navigateur préféré et vous voilà prêt à jouer !

2 Présentation du jeu

2.1 Le concept

Pour notre projet, nous avons opté pour un jeu dynamique et non statique. Ainsi, l'état du jeu évolue avec le temps, même sans l'action des joueurs. Nous nous sommes tournés vers les jeux des années 70 : Pong, qu'on ne présente plus et Breakout, qui a lancé le genre du casse-briques.

Le jeu réalisé, que nous avons nommé Breach, est donc un mélange de ces grands classiques. Cependant, contrairement à ses modèles, l'objectif principal de Breach est de pouvoir jouer en ligne contre ses amis (même si le jeu en local a été implémenté).

2.2 L'interface

L'application de jeu présente une page d'accueil qui permet de faire le choix entre les 4 options disponible : créer une partie en ligne, rejoindre une partie en ligne, jouer en local contre un ami, affronter l'ordinateur (CPU).

L'écran de jeu est similaire à celui du jeu Pong auquel on a rajouté des obstacles qui sont les briques de Breakout. Ainsi, on trouve deux raquettes latérales (à gauche et à droite) contrôlable par souris (et clavier pour le deuxième joueur en local), chacune avec une balle au début du jeu.

2.3 Fonctionnalités

Le jeu présente un système de vies et se gagne au bout de trois manches remportées sachant que le perdant d'une manche est le premier joueur à ne pas renvoyer une des deux balles. Il s'agit donc d'un BO5 (Best of 5), ce qui est un bon format pour avoir des parties courtes et dynamiques.

Par ailleurs, les briques renferment des bonus et malus. Au lieu d'apparaître et de "tomber" vers les raquettes, ceux-ci dépendent de la couleur des briques et s'activent immédiatement, ce qui favorise les tirs précis.

3 Réalisation du jeu

Nous avons utilisé le motif d'architecture logicielle Model - View - Controller (MVC) pour réaliser notre application Web. Le modèle est géré par NodeJs et Socket.io via les fichiers server.js et game.js ; la vue est gérée en HTML et CSS via les fichiers index.html et style.css ; le contrôleur réside dans le fichier index.js qui utilise le Javascript pour gérer la liaison et l'affichage dynamique.

3.1 La gestion des rebonds

Si au cours de la phase de développement, la balle était toujours renvoyée de manière symétrique par rapport à la verticale (dx devient -dx), ce fonctionnement montre vite ses limites car deviner la trajectoire devient très simple après 2-3 parties.

Une solution pour éviter cela est de faire varier l'angle des tirs en fonction de la position de la balle par rapport à l'obstacle ou la raquette et notamment le centre de ces objets. Nous sommes finalement arrivés à ces formules (à une gestion du signe de dx près) :

$$point_{impact} = \frac{ball.y - (user.y - user.height/2)}{user.height/2}$$

$$\begin{cases} dx = ball.speed \cdot \cos(point_{impact} \cdot \frac{\pi}{4}) \\ dy = ball.speed \cdot \sin(point_{impact} \cdot \frac{\pi}{4}) \end{cases}$$

3.2 La gestion du jeu en ligne

Les parties en ligne se gèrent via un identifiant aléatoire qui est créé lorsqu'un joueur initie une partie. En transmettant cet identifiant, il défie un(e) ami(e) qui peut donc rejoindre la même partie. Cette "astuce" permet de lancer un grand nombre de parties et à de nombreux utilisateurs de jouer en simultané.

3.3 La gestion du CPU

Un mode solo contre l'ordinateur a également été implémenté. Celui-ci connaissant la vitesse de la balle à tout instant (selon l'axe vertical), il serait en théorie impossible à

battre. Cependant, il ne faut pas oublier qu'il y a deux balles en jeu ce qui rend, en fait, la tâche plus compliquée.

Mais, nous avons recherché une formule "équilibrée" pour que l'ordinateur représente un challenge suffisant et stimulant tout en évitant des discontinuités incohérentes et nous avons abouti à la formule suivante :

$$y_{new} = y_{old} + (y_{ball} - y_{old}) \cdot 0.1 + dy_{ball} \cdot 0.6$$

La nouvelle position de la raquette dépend donc de son écart à la balle et la vitesse de la balle, la balle considérée étant la plus proche du mur de droite.

4 Améliorations du jeu

Parmi les plusieurs pistes d'améliorations possibles, nous en avons envisagé quelques-unes mais nous n'avons pas eu le temps de les réaliser, notamment l'hébergement du jeu pour pouvoir vraiment jouer en ligne (faisable via Heroku).

4.1 Jeu et gameplay

Concernant le jeu, toutes les briques sont identiques (si on ignore les bonus). On a pensé à l'ajout éventuel d'un attribut solidité des briques (facile à faire puisque stocké en JSON) et il serait nécessaire de les toucher plusieurs fois pour les casser.

Aussi, nous n'avons implémenté que deux bonus/malus : les briques vertes qui agrandissent les balles et les briques jaunes et qui les accélèrent. On peut en ajouter d'autres : rétrécissement, décélération, puissance (pour casser les briques solides), vitesse variable, dédoublement, ...

Enfin, on a pensé à une gestion différée du trackpad et de la souris, qui s'est révélée très difficile à réaliser car il faut identifier le périphérique, les deux événements correspondant au même mousemove. Mlaheureusement, l'utilisation tactile via touchmove équivaut aussi au mousemove.

4.2 Expérience utilisateur

Pour améliorer le confort de jeu, plusieurs fonctionnalités peuvent être ajoutées comme l'ajout d'un décompte avant de commencer chaque partie ou l'ajout d'une possibilité de rematch directement à la fin d'une partie.

De plus, deux modules supplémentaires de Hall of Fame et de Chat peuvent être implémentés. Nous n'avons pas vu l'intérêt d'un Chat puisque c'est un jeu dynamique mais il pourrait servir entre deux parties. Le Hall of Fame peut se gérer facilement via un document texte ou une base de données comme nous l'avions fait pour le jeu du pendu en INF tc2.

5 Captures d'écran

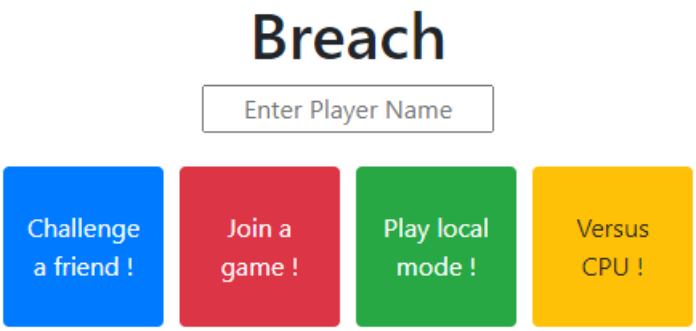

FIGURE 1 – Écran d'accueil

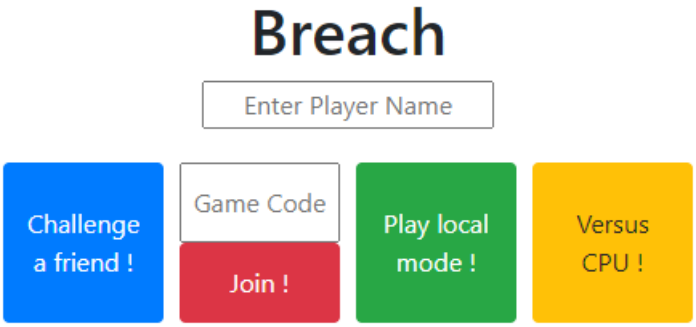

FIGURE 2 – Pour rejoindre une partie

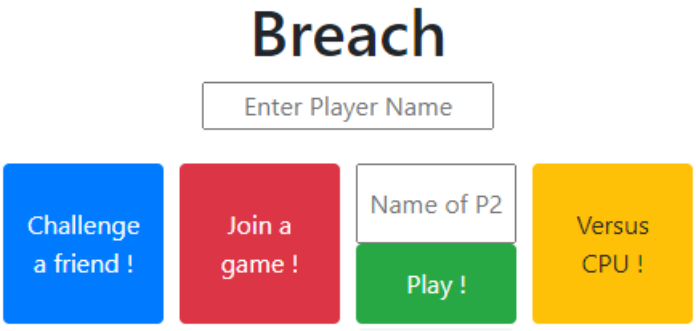

FIGURE 3 – Pour jouer en local

FIGURE 4 – En pleine partie

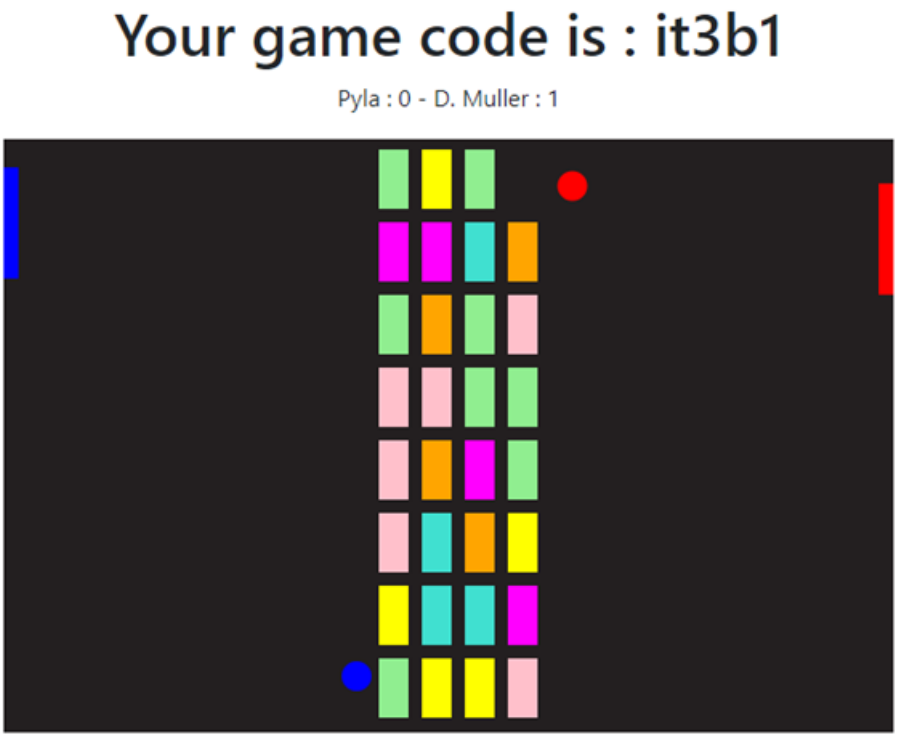

FIGURE 5 – Face à l'ordinateur