Projet de Programmation : Camlbrick

Université de Poitiers - Année Académique 2023-2024

Noms des membres de l'équipe :

CASTRO MENDOZA, Matias Jose -> 100%

KERAN JOYEUX -> 80%

BENNEGENT Enzo -> 0%

MIRMOHAMMADI Sina -> 0%

Introduction :

Notre objectif dans ce projet est de développer un jeu de casse-briques. Nous contrôlons une raquette se déplaçant horizontalement en bas de l'écran qui renvoie une balle détruisant des briques au contact. Le but est de détruire toutes les briques destructibles du niveau. La difficulté réside dans le fait de diriger la balle vers les briques restantes. Si la balle dépasse la zone de rebond de la raquette, elle est détruite et une vie est perdue. Le jeu est perdu si toutes les balles disparaissent de l'écran.

Notre jeu comprend plusieurs composantes de l'interface graphique :

- La zone de menu avec des boutons et des informations pertinentes.

- Le monde du jeu affichant l'ensemble du niveau.

- La zone des briques où sont affichées les briques à détruire.

- La zone d'évolution des balles où elles peuvent se déplacer.

- La raquette et sa zone de rebond.

Le projet implique également la définition des paramètres du monde du jeu, tels que les dimensions du monde et des briques, la taille initiale de la raquette, et la vitesse du jeu. Nous sommes responsables de la mise en œuvre de la logique du jeu, de l'interface graphique et de la gestion des événements.

Voici les instructions pour lancer le code de ton projet, formatées pour être copiées dans un fichier texte sans marqueurs de formatage :

Instructions pour lancer le code

Pour compiler et exécuter le jeu Camlbrick, suivez les étapes suivantes dans votre terminal :

1. Compilation :

Utilisez la commande suivante pour compiler les fichiers du projet. Cette commande lie les bibliothèques nécessaires et génère l'exécutable :

ocamlfind ocamlopt -o camlbrick -linkpkg -package labltk,unix,graphics camlbrick.ml camlbrick_gui.ml camlbrick_launcher.ml

2. Exécution :

Une fois le projet compilé, vous pouvez lancer le jeu en exécutant :

./camlbrick

Ces étapes vous permettront de démarrer le jeu directement depuis la ligne de commande. Assurez-vous d'avoir tous les paquets nécessaires installés et que les fichiers sources sont dans le répertoire courant avant d'exécuter ces commandes.

Itération 1 : Fonctions initiales et préparation des briques

Préparation des vecteurs : Nous avons défini le type t_vec2 pour représenter des vecteurs en 2D. Fonctions implémentées :

make_vec2(x, y): Construit un vecteur à partir de deux entiers.

vec2_add(a, b): Calcule la somme de deux vecteurs.

vec2_mult(a, b): Multiplie les composantes de deux vecteurs.

Préparation des briques : Structures de données pour les briques avec différents types et effets. Fonctions implémentées :

brick_get(x, y): Renvoie le type de brique selon les coordonnées.

brick_hit(x, y): Simule l'impact d'une balle sur une brique.

brick_color(b): Renvoie la couleur d'une brique selon son type.

Itération 2 : Gestion de la raquette et préparation des balles

Gestion de la raquette : Définition du type t_paddle et intégration dans t_camlbrick pour stocker des informations sur la raquette. Fonctions implémentées :

make_paddle(): Crée une raquette pour les tests.

paddle_x: Renvoie la position gauche de la raquette.

paddle_move_left(), paddle_move_right(): Déplacent la raquette.

Préparation des balles : Définition du type t_ball pour les balles. Fonctions implémentées :

has_ball, balls_count: Gèrent la présence et le nombre de balles.

balls_get, ball_get: Récupèrent les balles.

ball_x, ball_y: Renvoient les coordonnées d'une balle.

ball_size_pixel: Taille de la balle en pixels.

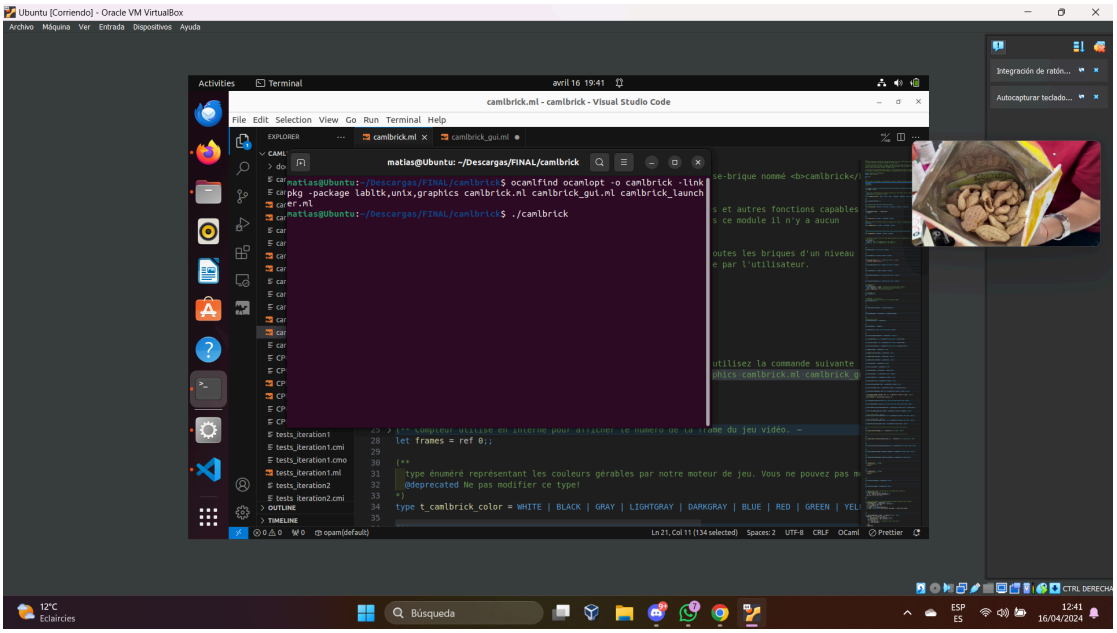
ball_color: Couleur d'une balle.

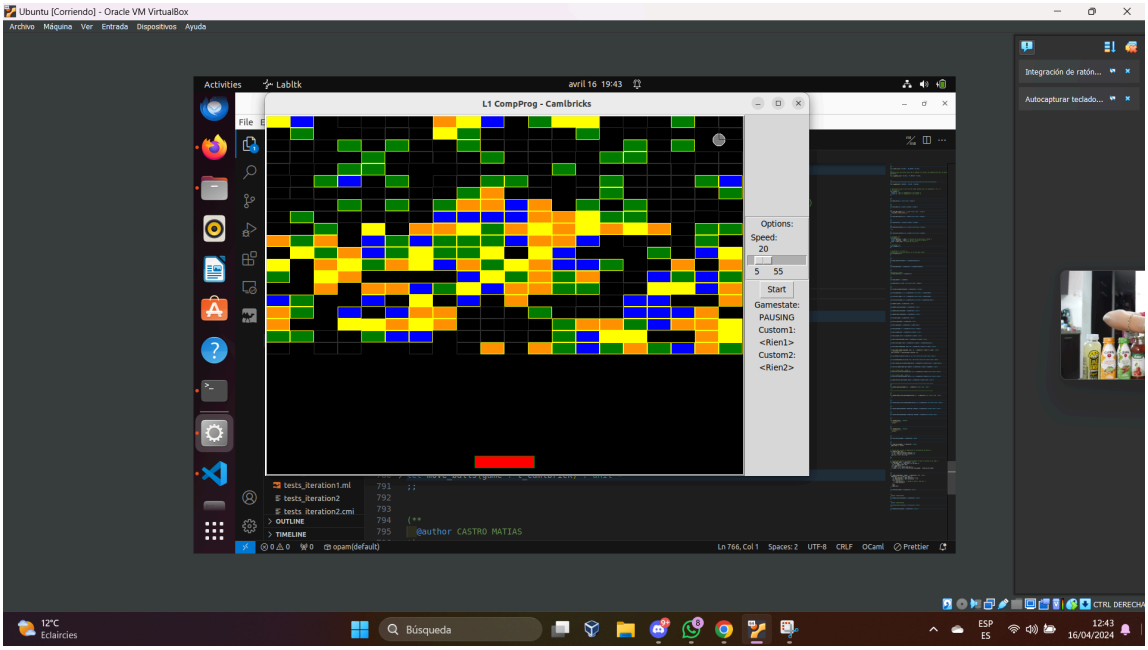
Itération 3 : Gestion des collisions et animations

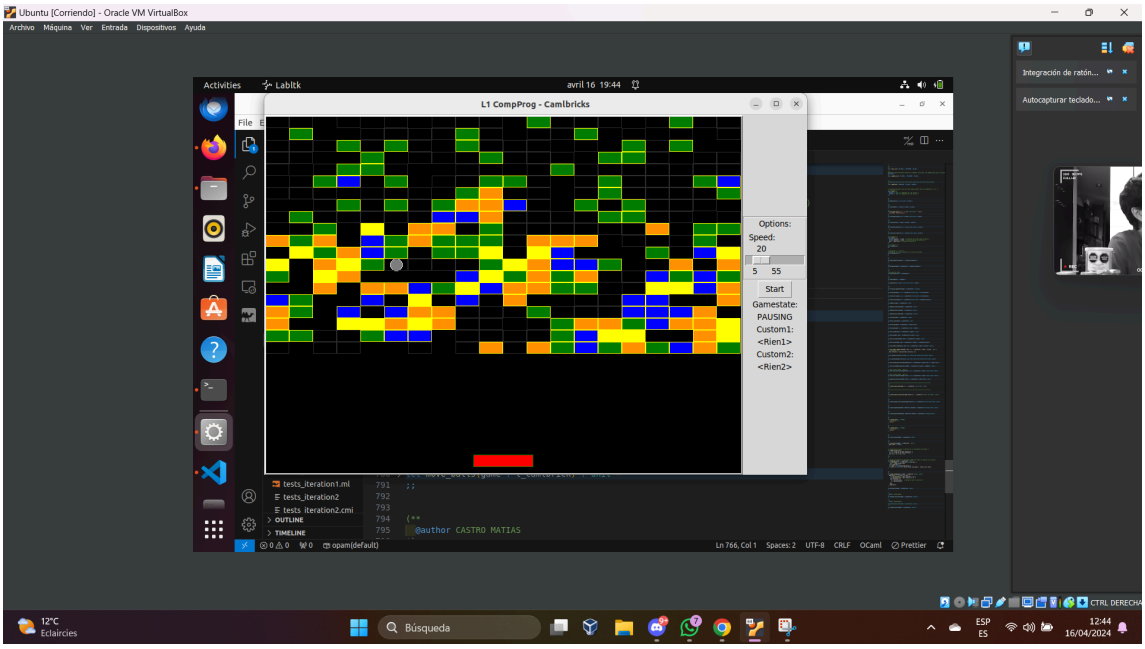
Animation des balles : Évolution de la position des balles à chaque image. Gestion des collisions avec les bords de l'écran.

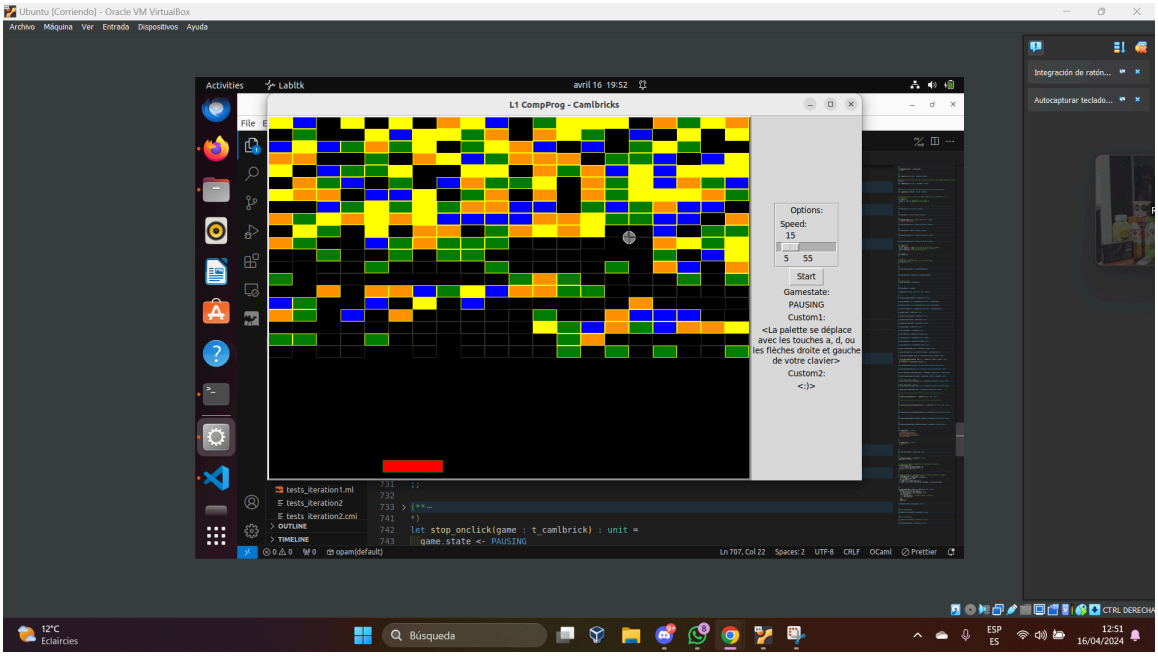
Gestion des collisions : Approches pour la détection des collisions avec les briques, soit par le centre de la balle, soit par un disque représentant la balle. Adaptation des réactions en fonction du type de collision.

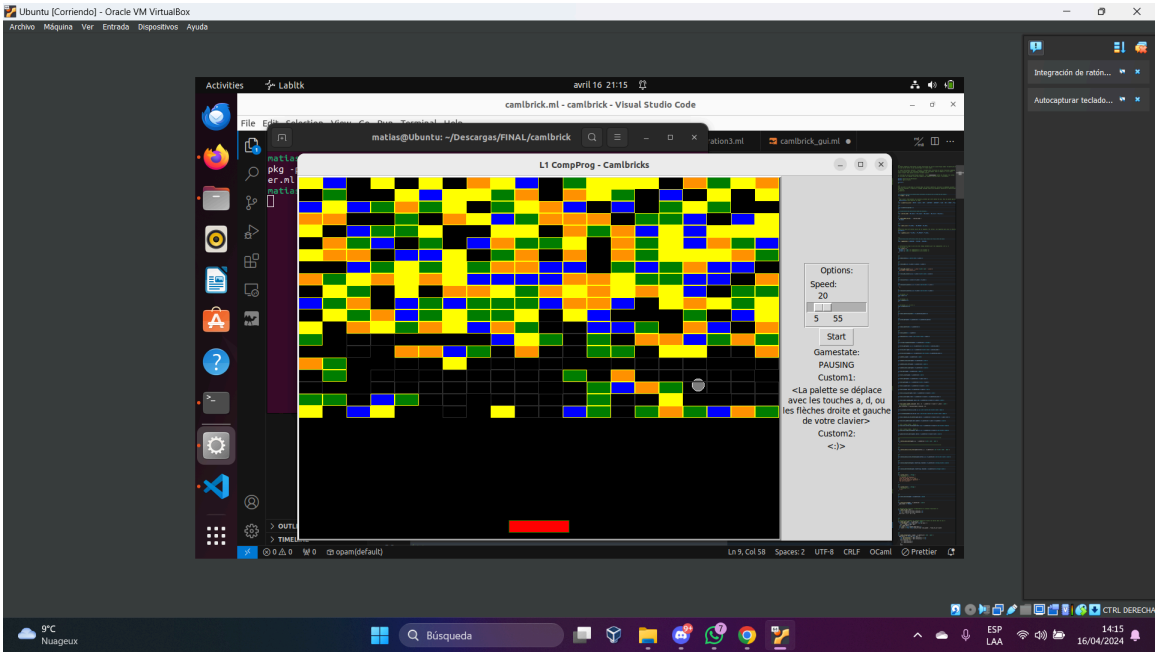
PHOTOS






Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities
Labltk
avril 16 19:44
L1 CompProg - Camlbricks
Options:
Speed:
20
5 55
Start
Gamestate:
PAUSING
Custom1:
<Rien1>
Custom2:
<Rien2>
tests_iteration1.ml
tests_iteration2
tests_iteration2.cmi
OUTLINE
TIMELINE
(**
@author CASTRO MATIAS
Integración de ratón...
Autocapturar teclado...
12°C
Eclaircies
Búsqueda
ESP
ES
12:44
16/04/2024


Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities
Labltk
avril 16 19:52
L1 CompProg - Camlbricks
Options:
Speed:
15
5 55
Start
Gamestate:
PAUSING
Custom1:
<La palette se déplace avec les touches a, d, ou les flèches droite et gauche de votre clavier>
Custom2:
<:)>
tests_iteration1.ml
tests_iteration2
tests_iteration2.cmi
OUTLINE
TIMELINE
let stop_onclick(game : t_camlbrick) : unit =
game.state <- PAUSING
Integración de ratón...
Autocapturar teclado...
12°C
Eclaircies
Búsqueda
ESP
ES
12:51
16/04/2024



Conclusion

Ce projet fut une expérience très enrichissante qui a grandement contribué à notre apprentissage et à l'amélioration de notre capacité à analyser des problèmes de programmation. Bien que nous ayons rencontré des difficultés initiales, notamment avec l'exécution du code et l'installation des paquets nécessaires pour visualiser l'environnement graphique, nous avons su surmonter ces obstacles. De plus, malgré les défis liés à la communication et à la collaboration avec certains membres de l'équipe, nous avons réussi à mener à bien ce projet.

L'effort collaboratif a nécessité une grande adaptabilité et une gestion efficace des ressources humaines, renforçant ainsi notre compréhension des dynamiques de groupe dans un contexte de développement logiciel. Ce projet nous a également permis de mieux comprendre l'importance de la persévérance et de la résolution créative de problèmes dans un environnement de programmation complexe.

En résumé, ce projet n'a pas seulement amélioré nos compétences techniques, mais a également renforcé notre capacité à travailler en équipe et à naviguer à travers les défis inhérents à tout projet de développement logiciel. Nous sommes fiers des résultats obtenus et confiants dans l'application de ces apprentissages dans nos futures entreprises académiques et professionnelles.