



Licence Informatique

TER L2  
Travail d'Étude et de Recherche

---

## 2048 version Billard

---

*Auteurs :*

M<sup>me</sup> Julie CAILLER

M. Félix COSTA

M. Julien DAVIAUD-DEMAILLE

M. Yannick HERRERO

*Encadrant :*

M. Lucas ISENMANN

2048  
pool



# Introduction

Dans le cadre de notre seconde année d'étude en Informatique, il nous a été demandé de réaliser l'un des projets proposés par la Faculté des Sciences de Montpellier. Outre le fait que le projet constitue une note importante pour notre moyenne ce semestre, il est également pour nous la possibilité d'appliquer ce que nous apprenons à un cas réel. Ce projet nous permet aussi de mieux appréhender les idées de travailler au sein d'une équipe sur une durée déterminée et d'approfondir nos connaissances dans un domaine précis.

Nous avons sélectionné le projet « 2048 version Billard »<sup>1</sup>, encadré par M. ISENMANN Lucas. Nous avons choisi ce projet car il nous a semblé intéressant de travailler sur quelque chose de graphique et dynamique, ce qui est plutôt éloigné de ce que nous avons vu en cours durant ce semestre et que cela nous permettait d'approfondir nos connaissances dans ce domaine. Nous avons choisi de réaliser ce projet en C++ et d'utiliser la bibliothèque SFML car tous deux correspondaient à ce que nous recherchions. Nous avons également dû imaginer les règles du jeu et les améliorations qui pourraient le rendre plus amusant.

L'équipe de projet est composée de quatre membres en deuxième année de Licence d'informatique : Julie Cailler, Félix Costa, Julien Daviaud-Demaille et Yannick Herrero. Ce projet a été réalisé dans le cadre de l'UE TER L2 de l'Université de Montpellier lors du deuxième semestre de l'année scolaire 2016-2017.

Tout au long de ce rapport, nous allons vous expliquer les règles du « Billard 2048 », mais aussi vous retranscrire les évolutions de notre projet, les différentes versions du jeu ainsi que les difficultés rencontrées lors de ce semestre.

---

1. Aussi appelé "Billard 2048"

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Présentation</b>	<b>4</b>
1.1 Règles du jeu . . . . .	4
1.2 Interface . . . . .	6
1.3 Modes de jeu . . . . .	10
1.3.1 Black . . . . .	10
1.3.2 Ghost . . . . .	11
1.3.3 Hard . . . . .	11
1.3.4 Combinaison de modes : Ghost + Hard . . . . .	12
1.3.5 Un peu plus loin... . . . .	12
<b>2 Développement</b>	<b>13</b>
2.1 Cahier des charges . . . . .	13
2.2 Les différentes versions du jeu . . . . .	14
2.3 Les difficultés rencontrées . . . . .	20
2.3.1 Collision/Fusion . . . . .	20
2.3.2 Pas amusant/Trop simple . . . . .	20
<b>3 Conclusion</b>	<b>21</b>
3.1 Évolution et perspectives . . . . .	21
3.2 Bilan . . . . .	21
<b>Remerciements</b>	<b>22</b>
<b>Bibliographie</b>	<b>23</b>

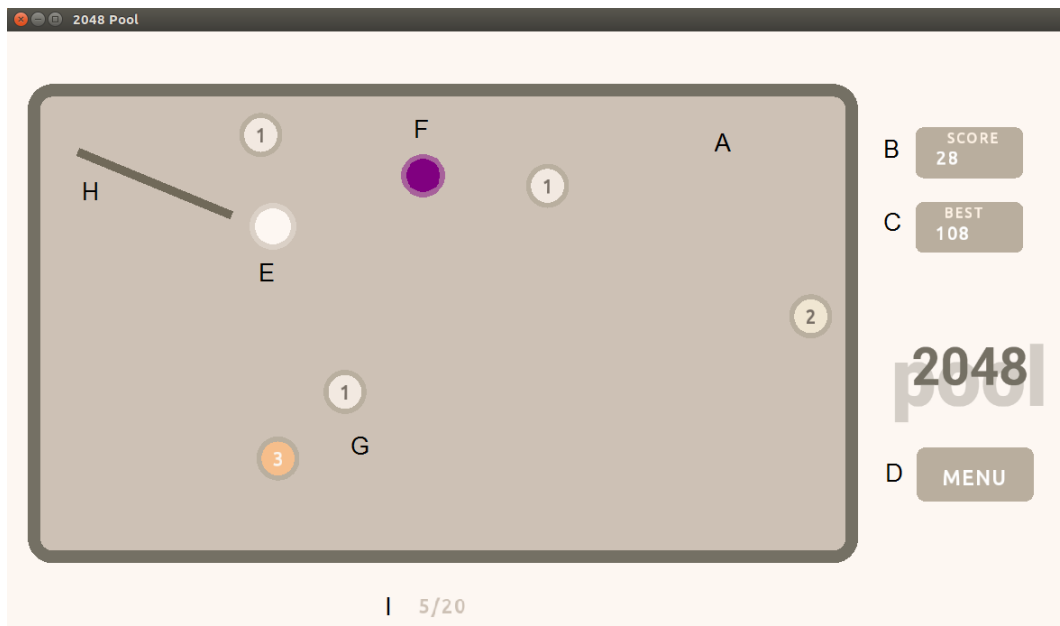




## 1.2 Interface

Dans un premier temps, nous avons songé à un design réaliste à la manière d'un véritable billard, (c'est à dire terrain vert et boules de différentes couleurs). Mais finalement, il nous a paru intéressant et logique de faire en sorte que le design du « Billard 2048 » fasse écho à celui du « 2048 » de base. C'est donc pour cette raison que nous avons opté pour un design simple, sobre et possédant la même gamme de couleur que celle du « 2048 ». (Nous avons également choisi un design plutôt rond, pas trop agressif, qui rappellerait les boules de billard.)

### Mode normal

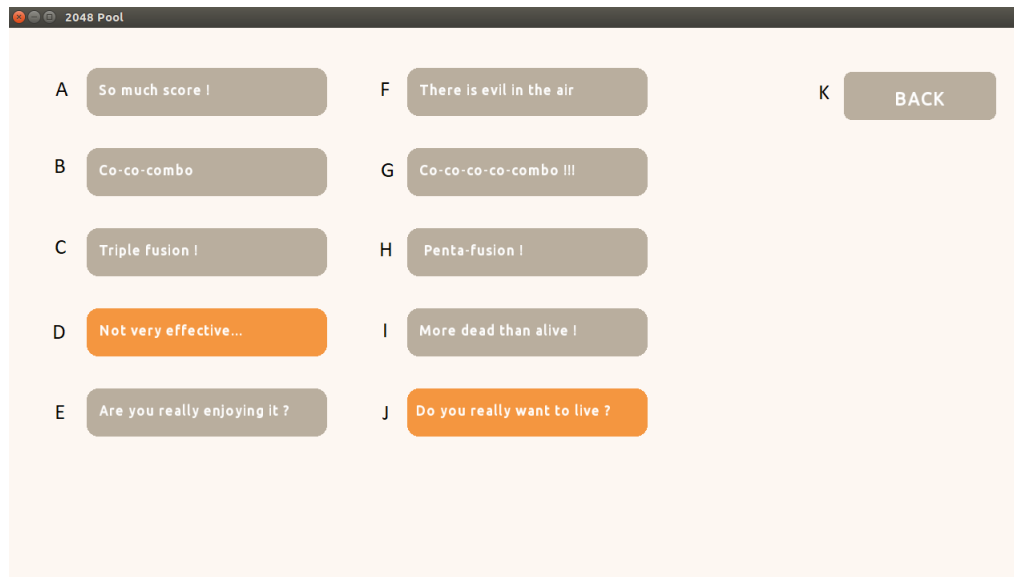


- A : Le plateau de jeu.
- B : Le score actuel.
- C : Le meilleur score enregistré.
- D : Le bouton pour accéder au menu et mettre le jeu en pause.
- E : La boule blanche.
- F : La boule violette.
- G : Les autres boules numérotées.
- H : La queue.
- I : Le nombre de boules à numéro actuellement sur le terrain et le nombre de boules maximal.





## Succès



- A : Avoir atteint la boule 11<sup>3</sup>.
- B : Avoir fait trois combos en un tour.
- C : Avoir fait trois fusions en un tour.
- D : N'avoir fait aucune fusion durant un tour.
- E : Avoir passé plus d'une heure sur le jeu.
- F : Avoir un score de 666<sup>4</sup>.
- G : Avoir fait cinq combos en un tour.
- H : Avoir fait cinq fusions en un tour.
- I : Avoir plus de boules noires que de boules normales sur le terrain.
- J : Perdre au premier tour.
- K : Revenir au menu - quitter l'écran des succès.

Les succès en oranges sont ceux ayant été réalisés. Ceux en gris n'ont pas encore été atteint.

Lorsque que l'on débloque un succès, il demeure tant que l'on reste sur le jeu. Il disparaît si on ne sauvegarde pas la partie.

---

3. Ce qui correspond à la tuile 2048 du jeu originel

4. Il faut atteindre le score exact

## Game Over



- A : Game Over !<sup>5</sup>
- B : Nombre de tours de jeu effectués durant cette partie.
- C : Quitter le jeu.
- D : Recommencer une nouvelle partie.

Il y a deux façons d'atteindre cet écran : en touchant la boule violette avec la boule blanche ou en atteignant le nombre maximum de boules sur le terrain.

---

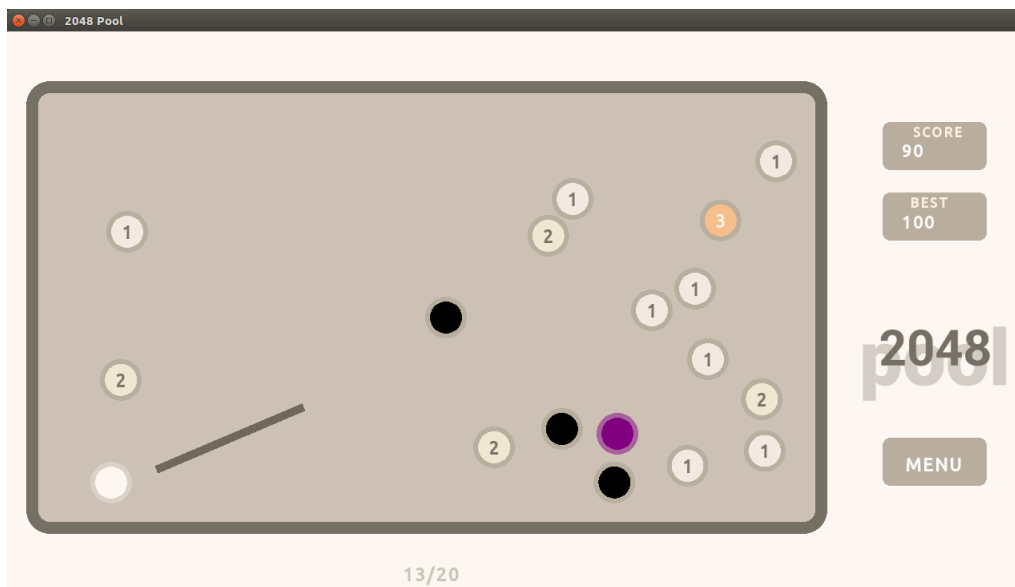
5. La partie est perdue. Il n'est pas possible de sauvegarder la partie.

## 1.3 Modes de jeu

Le jeu comporte trois modes de jeu : Ghost, Hard et Black. Ces modes de jeu ont pour but de rendre le jeu plus intéressant et difficile. Lorsqu'on active un mode de jeu, le score obtenu lors d'une fusion est multiplié par 1,5 fois le nombre de modes de jeux activés. Le changement de modes de jeu réinitialise automatiquement la partie.

### 1.3.1 Black

Lorsque le mode Black est activé, à chaque tour, il y a une probabilité qu'une boule noire et non-fusionnable<sup>6</sup> apparaisse. Ces boules ne comptent pas dans le total de boules à ne pas dépasser.

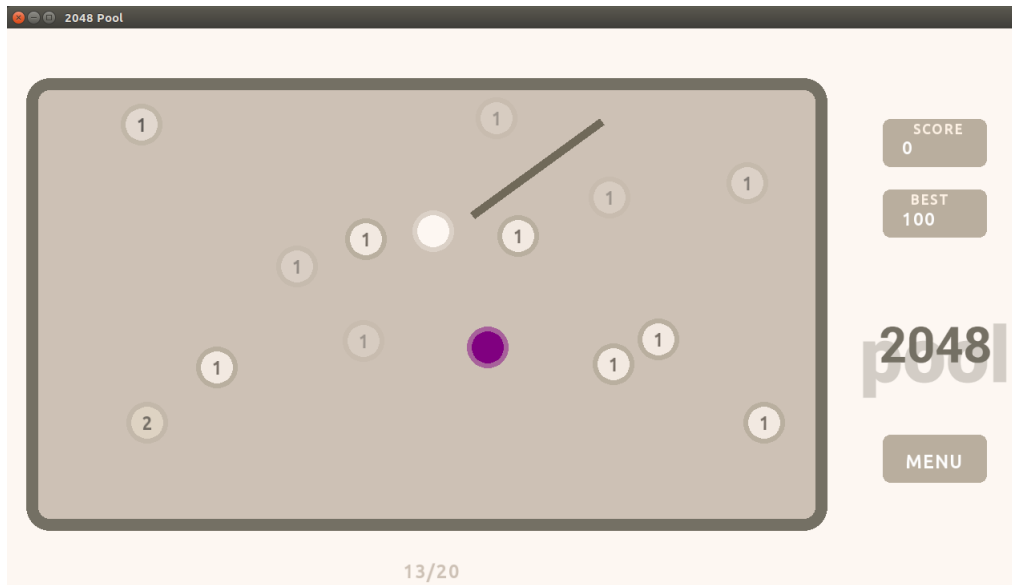


---

6. Ces boules noires ne peuvent se fusionner avec aucune autre boule, y compris d'autres boules noires.

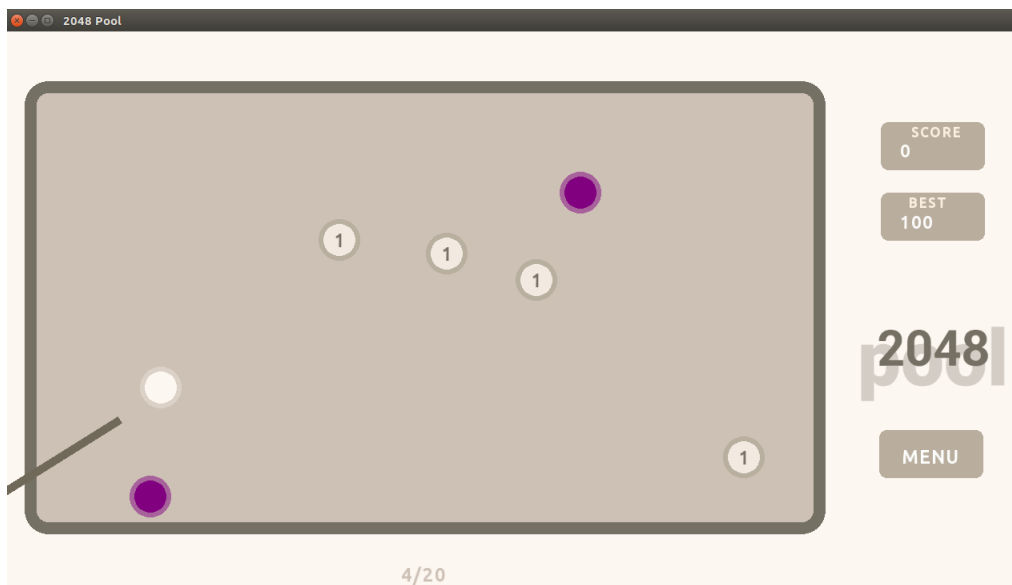
### 1.3.2 Ghost

Lorsque le mode Ghost est activé, certaines boules deviennent invisibles quand elles ne bougent plus, et réapparaissent lorsqu'elles entrent en mouvement



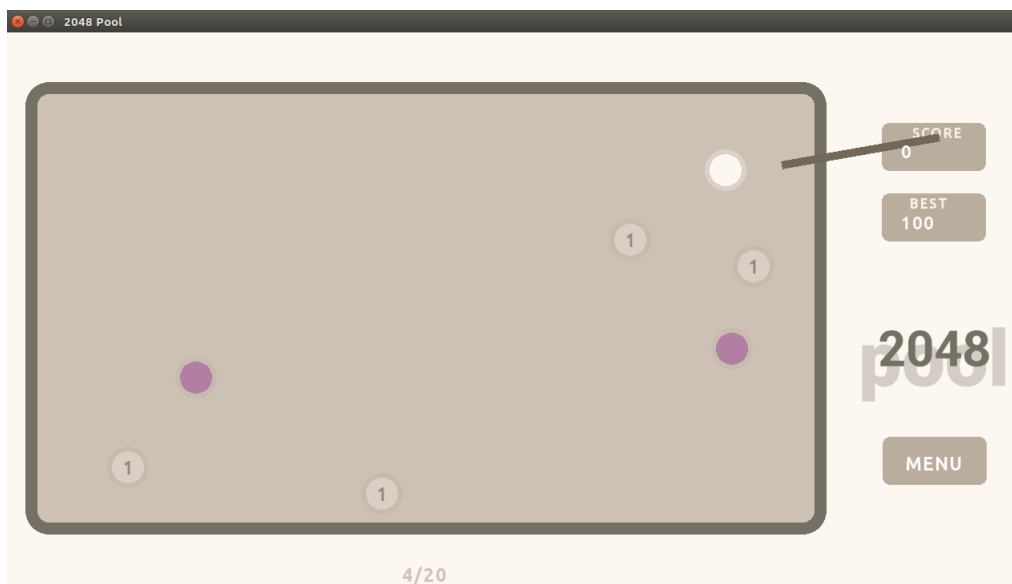
### 1.3.3 Hard

Lorsque le mode Hard est activé, il y a deux boules violettes sur le terrain et deux boules qui apparaissent à chaque tour.



### 1.3.4 Combinaison de modes : Ghost + Hard

Lorsque les modes Ghost et Hard sont activés en même temps, toutes les boules subissent l'effet de Ghost.



### 1.3.5 Un peu plus loin...

Taper "FOUR" au clavier pendant une partie pour débloquer une surprise...

# Développement

## 2.1 Cahier des charges

Le cahier des charges au début du projet était simple : nous avions trois mois pour produire un jeu de 2048 version billard fonctionnel et amusant. Nous avons défini lors d'un premier rendez-vous les principales tâches à effectuer :

- Définir les règles du jeu (quelles libertés par rapport aux jeux d'origine ?)
- Faire un jeu fonctionnel au niveau des interactions entre les boules.
- Faire une interface graphique esthétiquement acceptable.
- Rendre le jeu amusant.

Par ailleurs, nous avons rencontré notre encadrant de projet toutes les deux semaines, dans l'optique qu'il puisse suivre de près la réalisation de notre travail et nous guider de la meilleure façon possible.

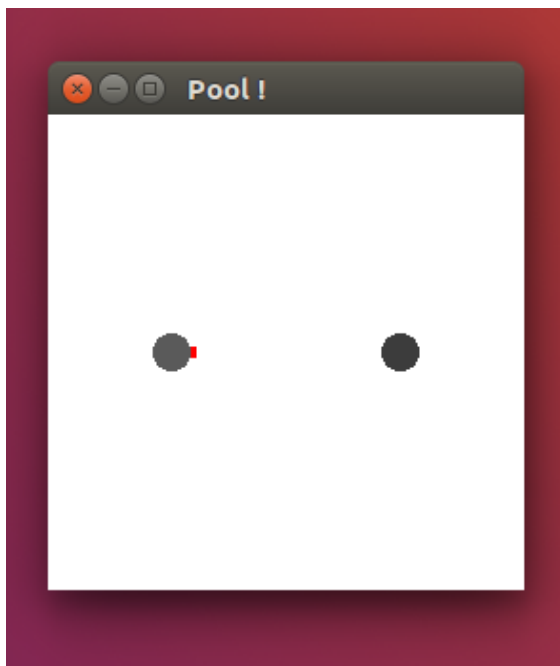
## 2.2 Les différentes versions du jeu

### Conventions et mise au point

Nous avons eu un premier rendez-vous avec notre encadrant, M. ISENMANN Lucas, durant lequel il nous a expliqué le principe du jeu et ce que nous allions devoir faire. Après cela, nous nous sommes réunis au sein du groupe afin de nous mettre au point sur les principaux points : la définition des conventions, le langage que nous allions employer (C++), la bibliothèque utilisée (la SFML) définition des objectifs et la répartition des tâches.

### Gestion des collisions

Notre première version du jeu fut une version de test du moteur physique du jeu. Nous n'avions qu'un "main" qui faisait apparaître deux boules et les mettait en mouvement. Cette première version nous a permis de mettre au point les collisions entre les boules et avec les murs.

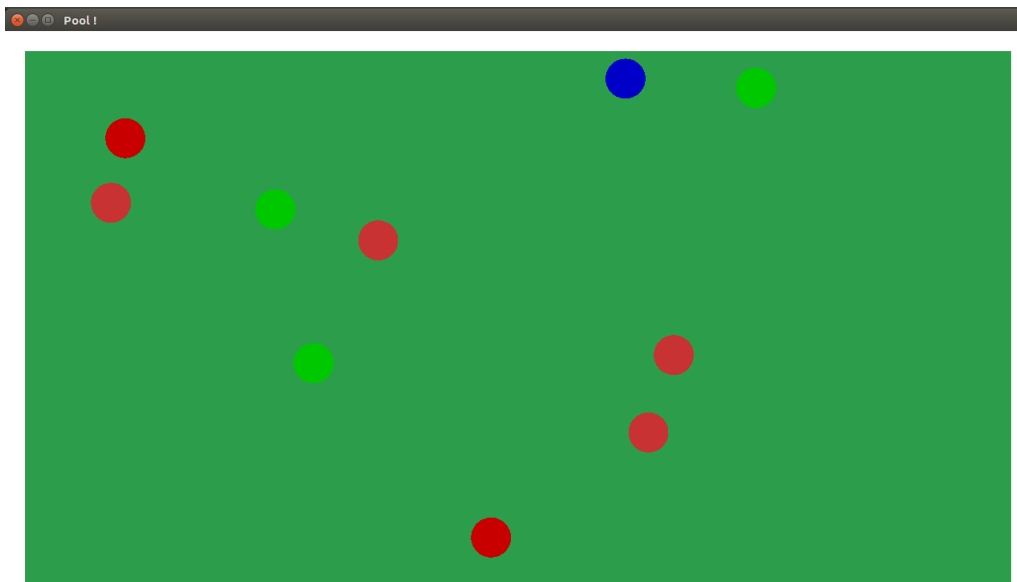


## Modularisation et création du terrain

Après avoir validé notre modèle physique, nous sommes passé au jeu à proprement parler. Nous avons décidé, au retour des vacances, de mettre en place un dépôt Git afin de pouvoir travailler plus efficacement.

Nous avons fait un code modulaire et avons créé un premier programme qui faisait apparaître le terrain de jeu et cinquantes boules. Cette version nous a permis de tester les collisions avec un grand nombre de boules mais aussi la fusion et les couleurs.

Elle a été le point de départ du projet, elle a également mis en évidence de nombreux problèmes de collision entre les boules.



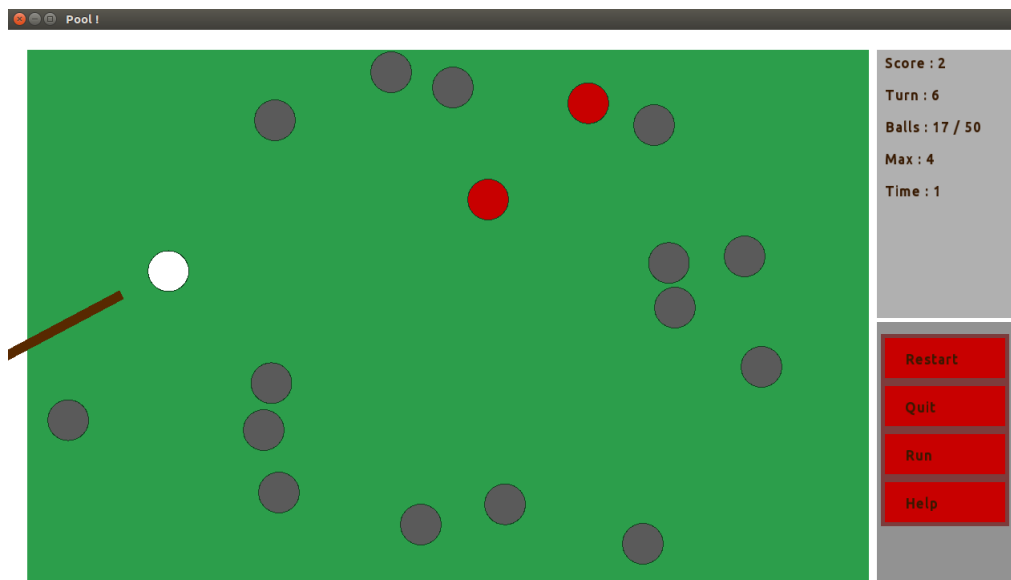


## Boule blanche, queue, statistiques et boutons

Cette version a vu naître la boule blanche et la queue. Nous avons également ajouté de la musique.

De plus, pour rendre l'interface plus intuitive, nous avons créé des boutons permettant de mettre le jeu en pause, de recommencer la partie, d'afficher l'aide et de quitter le jeu. Nous avons également mis en place un cadre d'affichage des statistiques<sup>1</sup> et un "game over".

A ce stade du jeu, nous avons une version fonctionnelle qui correspondait à la problématique de départ.



---

1. Comme le score, le tour de jeu ou le nombre de boules sur le terrain.



## Améliorations diverses

Après l'apparition des mode de jeux, et dans la lignée de la refonte du code, nous avons apporté quelques améliorations au jeu afin de le rendre plus intuitif pour le joueur.

Nous avons ainsi créé une animation de disparition des boules lors des fusions, mis en place la disparition progressive des boules lors du mode Ghost, modifié la gestion des boules noires dans le mode Black, mis en place la notion de combo qui influence le score, fait apparaître les numéros sur les boules<sup>2</sup> et mis en place la gestion d'un meilleur score.

Nous avons également offert au joueur la possibilité de changer de musique.

## Succès et sauvegarde

Toujours dans le but de rendre le jeu plus amusant, nous avons mis en place les succès qui sont des actions particulières à accomplir par le joueur.

Nous avons également mis en oeuvre la possibilité de sauvegarder la partie et de la recharger plus tard.

---

2. À l'arrêt uniquement, car cela rendait mieux graphiquement parlant.

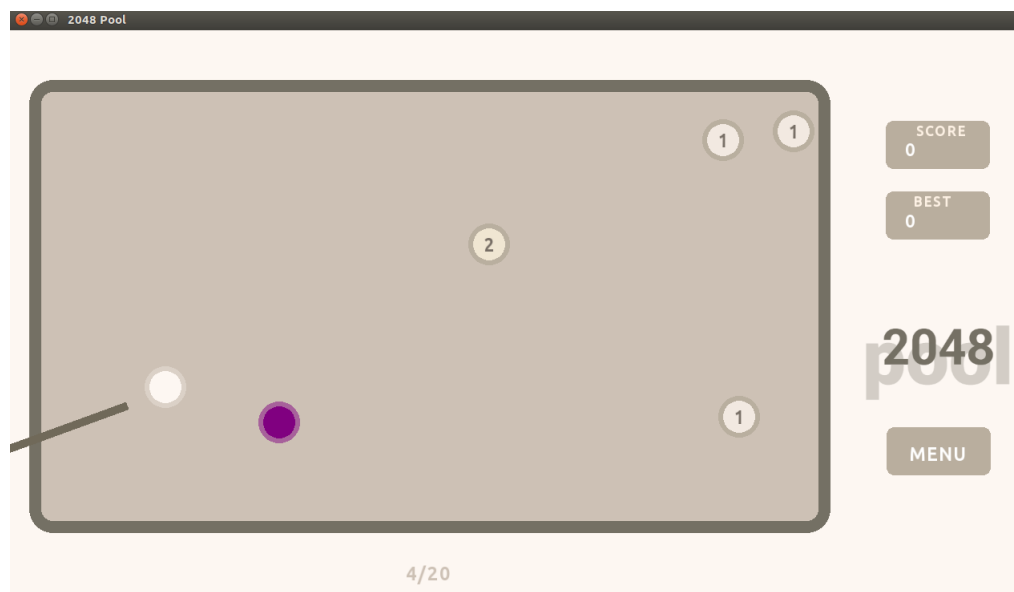
## Mise à jour graphique et boule violette

La dernière mise à jour majeure a révolutionné l'aspect graphique de la fenêtre, afin de se rapprocher du thème proposé dans le jeu de 2048.

Afin de rester dans l'ambiance générale, nous avons décidé de changer la sélection de musiques et de sons.

Afin de conserver une cohérence dans le menu, accessible via le bouton menu, nous avons retiré les modes de jeux Hurry et Time, peu intéressants et amusants. Cette mise à jour a vu une augmentation importante de la difficulté du jeu, avec le passage de cinquante à vingt boules au maximum, ainsi que l'apparition des balles violettes qui créent un "game over" instantané lorsque la boule blanche les heurte.

La dernière innovation apportée par cette mise à jour est la sauvegarde des succès.



## 2.3 Les difficultés rencontrées

### 2.3.1 Collision/Fusion

Le premier problème que nous avons rencontré était la présence de soucis de collision. Ce problème a perduré longtemps, il était complexe car composé de nombreuses petite erreurs réparties dans tout le code.

En effet, les fonctions de collisions sont nombreuses et dépendent toutes plus ou moins les unes des autres. De plus, elles font partie des premières fonctions créées, il a donc fallu les adapter aux évolutions du code.

Ce problème n'a pas été résolu d'un coup, il a été corrigé tout au long du projet, au fur et à mesure que les erreurs apparaissaient. À l'heure actuelle, il peut parfois arriver que des problèmes de collision apparaissent.

### 2.3.2 Pas amusant/Trop simple

L'autre gros problème que nous avons rencontré, et qui a grandement participé à l'avancement du jeu, est la difficulté : il était trop facile.

Notre encadrant nous avait prévenu au début du projet que nous devrions rendre le jeu amusant, mais nous avons été surpris de voir à quel point il était difficile (voire impossible) de perdre au début.

En effet, Le joueur pouvait passer des heures sur le jeu sans perdre et cela même si le mode Hard était activé. Nous devions modifier cela, sans quoi ces parties interminables auraient pu causer chez l'utilisateur une certaine lassitude , pour finalement aboutir à l'abandon du jeu par celui-ci.

Cela nous a amené à mettre en place les modes de jeux afin d'augmenter la difficulté, les succès pour donner plus d'intérêt au jeu et également à réduire le nombre de boules nécessaire sur le terrain pour perdre et de créer la boule violette, que le joueur devrait éviter sous peine de perdre la partie immédiatement si la boule blanche venait toucher cette dernière. Ce choix apportait une certaine difficulté au jeu, ce qui lui manquait cruellement jusqu'alors, ainsi qu'une certaine réflexion de la part du joueur durant une partie, car il devrait orienter sa manière de jouer en fonction de la position de la boule violette.

# Conclusion

## 3.1 Évolution et perspectives

Parmis les améliorations possibles de notre jeu, nous aurions pu ajouter un système de bonus, basé sur les boules. Par exemple, une boule qui, lorsqu'elle est heurtée par une autre exploserait, ou pourrait fusionner avec n'importe quelle boule, pourrait faire disparaître tout ou partie des boules noires. Les possibilités ne sont limitées que par notre imagination.

Il aurait aussi été envisageable d'ajouter des vies, que l'on pourrait gagner à l'aide de bonus ou à certains paliers de score. Ces vies pourraient protéger d'une collision contre la boule violette, afin d'éviter le "game over" instantané.

Nous avons également pensé en faire une version portable sous Windows, iOS ou Android.

## 3.2 Bilan

Ce projet inscrit sur la durée, réalisé dans le but de créer un jeu de Billard 2048, nous a permis de mieux appréhender la programmation orientée objet, d'une façon différente de celle abordée en cours.

Ce projet a également été pour nous l'occasion de travailler sur quelque chose de graphique et de dynamique, ce qui est un aspect de la programmation de nous n'avons pas abordé en cours. Ce fût très intéressant d'apprendre à utiliser une bibliothèque telle que la SFML et de voir quelques notions de programmation graphique et d'interface utilisateur, de réfléchir à la façon dont les joueurs allaient jouer, de prendre en compte les demandes de l'encadrant et des personnes qui testaient le jeu, de voir ce qu'ils attendaient du jeu, l'ergonomie des menus, la pertinence des modes de jeux, etc.

Nous avons réalisé au total une vingtaine de version du jeu. Ce projet fut très enrichissant, tant sur le plan personnel et professionnel que scolaire. Il a constitué une véritable mise en situation de conduite de projet et nous a permis de nous confronter à la réalité du travail en équipe.

# Remerciements

Nous tenons à remercier notre encadrant, M. ISENMANN Lucas pour son aide ainsi que pour cette collaboration qui s'est, de notre point de vue, merveilleusement bien passée. Toutes les réunions se sont déroulées dans une atmosphère agréable et détendue, mais tout en restant très professionnelle. Son suivi nous a apporté l'aide dont nous avons besoin lors des différentes phases de développement. Il nous a également permis de ne pas dévier de notre objectif et ainsi garder notre cap dans le but de rendre un programme cohérent avec les attentes de notre projet.

Nous remercions aussi toutes les personnes qui nous offrent un peu de leur temps afin de tester le jeu. Cela nous a permis à plusieurs reprises de le remettre en question et ainsi de l'améliorer.

Nous remercions également la Faculté des Sciences de Montpellier pour nous avoir offert l'opportunité de réaliser ce projet et pour tout ce que nous avons appris en le réalisant.

Merci à M. Gilles Daviaud d'avoir composé la musique du jeu, ce qui nous a permis que notre "Billard 2048" possède une musique libre de droits.

Merci à Rémi "Overdriv" Bèges qui a fait la classe RoundedRectangle et qui l'a partagée publiquement sur Internet.

# Bibliographie

Documentation SFML : <https://www.sfml-dev.org/documentation/2.4.2/>

Site sur lequel nous avons trouvé la classe RoundedRectangle :  
<https://github.com/SFML/SFML/wiki/Source:-Draw-Rounded-Rectangle>