

UNIVERSITÉ DE LORRAINE

PROJET DE C

Partie 1 : Gestion de Projet

Chef de projet :
Raphael OHANIAN

Développeurs :
Julien DERIGNY
Maxime RIÈRE

31 mai 2017

Table des matières

1	Introduction	1
1.1	Le projet	1
1.2	L'équipe	1
2	État de l'art - partie gameplay	1
2.1	Bibliographie	1
2.2	Bref historique et but	1
2.3	Règle du jeu	3
2.4	Stratégie	4
2.4.1	Attaque	4
2.4.2	Défense	4
3	État de l'art - partie IA	5
3.1	Bibliographie	5
3.2	Ordinateur intelligent	5
3.2.1	Algorithme MinMax	5
3.2.2	Algorithme offensif	6
3.3	Intelligence artificielle	6
3.3.1	Deep learning	6
4	Conception du code	7
4.1	Notre jeu	7
4.2	Création du plateau	7
4.3	Création des règles du jeu	7
4.4	Création de la fenêtre	8
4.5	Création du Makefile	8
4.6	Utilisation de Shell	8
4.7	Ordinateur intelligent	9
5	Graphiques	10
6	Qualité du programme	10
6.1	Complexité	10
6.2	Rapidité exécution	10
6.3	Tests	10

7 Outils de GdP	11
7.1 Google Drive	11
7.2 Messenger	11
7.3 Slack	11
7.4 Git	11
7.5 LaTeX	11
7.6 Poste de travail	11
8 Matrice SWOT	12
9 GANTT	12
9.1 Rendu	12
9.2 Tableur	14
10 Cartes Conceptuelles	15
10.1 Carte Conceptuelle du projet	15
10.2 Carte Conceptuelle du jeu	16
11 Difficultés	17
11.1 Estimation du temps de travail	17
11.2 Respect du Gantt	17
11.3 Plages horaire de travail	17
11.4 Gestion des couleurs	17
11.5 Le bug Z	17
12 Enrichissements	18
12.1 Intérêt	18
12.2 Interface graphique	18
12.3 Groupe de travail	18
12.4 Démarrer de rien	19
12.5 Maitrise de Git	19
13 Temps passé sur le projet	20
14 Conclusion	21
15 Sources	22
16 Annexe	23

1 Introduction

1.1 Le projet

Le projet que nous avons choisi est "HEX". Ainsi, il nous faut suivre les consignes du sujet et les adapter à ce jeu. À savoir créer le jeu jouable en 1 contre 1 (deux personnes), et jouable seul (contre l'ordinateur). Nous devons donc créer le plateau, les interactions pour deux joueurs, un algorithme de pose de pion pour l'ordinateur.

1.2 L'équipe

Notre équipe est composée de trois personnes. Nous avons des études légèrement différentes ce qui nous donne de la diversité.

2 État de l'art - partie gameplay

2.1 Bibliographie

Voici les différentes sources que nous utilisons pour cette partie :

<http://www.mattesmedjan.se/hexilla/>

<http://math.univ-lyon1.fr/irem/IMG/pdf/Hex.pdf>

<http://www.lutanho.net/play/hex.html>

<https://fr.wikipedia.org/wiki/Hex>

http://jeuxstrategieter.free.fr/Hex_complet.php

2.2 Bref historique et but

Le premier énoncé de la règle du jeu est l'œuvre de Piet Hein lors d'une conférence en 1942 au Parenthesis, l'association des mathématiciens de l'Université de Copenhague. L'auteur cherchait alors à imaginer un jeu équitable, progressif, fini, clair, stratégique et décisif. Le jeu porte alors le nom de Polygone. En 1948, et de manière indépendante, un jeune mathématicien de l'Université de Princeton du nom de John Nash réinvente le jeu et le diffuse au Fine Hall, une salle commune aux élèves et aux professeurs du département de mathématiques. Il en perçoit l'intérêt en théorie des jeux et

démontre l'existence d'une stratégie gagnante pour le premier joueur. Cependant, sa preuve est non constructive, c'est-à-dire qu'elle n'indique pas de stratégie gagnante. En 1952, la société Parker Brothers commercialise le jeu aux États-Unis.

Le jeu de Hex est un jeu de société combinatoire abstrait pour deux joueurs. Il se joue sur un tablier en forme de losange dont les cases sont hexagonales. Toutes les dimensions du côté du losange sont possibles, la plus traditionnelle est celle composée par 11 hexagones, mais on trouve aussi les valeurs 13 ou 19. L'un des inventeurs du jeu, John Nash, préconise un tablier de taille 14x14. Ce jeu possède des similarités avec le go. Inventé par des mathématiciens, il fait uniquement appel à la logique, à l'image du go ou des échecs. Son étude est source d'inspiration, non seulement en théorie des jeux, mais aussi pour d'autres branches des mathématiques comme la topologie ou la géométrie algébrique. Si l'on sait qu'il existe une stratégie gagnante pour le premier joueur, cette stratégie est inconnue si le tablier n'est pas de petite taille (de côté strictement plus petit que 9). La recherche de stratégies efficaces, à défaut d'une stratégie gagnante, est l'objet d'études en intelligence artificielle.

2.3 Règle du jeu

Au départ, le plateau est vide. Les Blancs ont le privilège de jouer les premiers. À tour de rôle, chacun des joueurs pose un pion de sa couleur sur la case de son choix - à la seule condition que cette case ne soit pas déjà occupée. Une fois posé, un pion ne peut plus être déplacé. Si deux pions se trouvent dans des cases ayant un côté commun, on dit qu'ils sont adjacents ou "liés". Une chaîne de pions liés est appelée "chaîne continue". La partie s'arrête quand un joueur a relié entre eux, par une chaîne continue de pions lui appartenant, les deux bords de sa couleur. Il ne peut y avoir de partie nulle, puisque l'un des deux adversaires finit toujours par y parvenir.

Le Plateau :

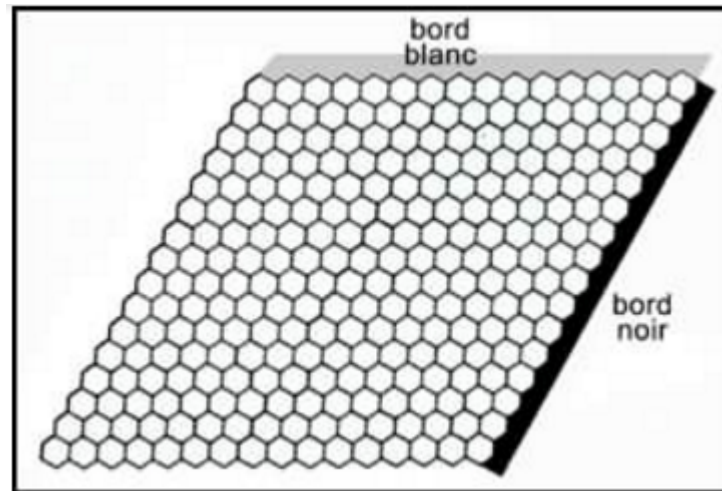


FIGURE 1 – Exemple de plateau de jeu

2.4 Stratégie

Dans ce jeu, il n'y a qu'un pas entre l'attaque et la défense. C'est pourquoi il y a beaucoup de rebondissements. Cependant il existe des dispositions de pions favorisant l'attaque et d'autres favorisant la défense.

2.4.1 Attaque

La stratégie d'attaque la plus efficace et la plus simple à mettre en oeuvre est celle mettant en jeu des pions "pseudo liés". C'est un alignement de pions adjacents à une même case vide. Une fois une telle chaîne réalisée l'adversaire ne peut contrer étant donné le choix de remplissage double : il y a toujours deux cases vides adjacentes à nos deux pions permettant de relier les pions quel que soit le positionnement des pions adverses.

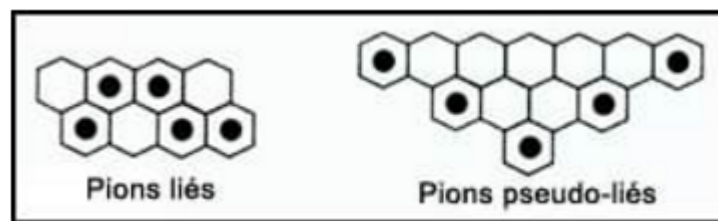


FIGURE 2 – Disposition d'attaque

2.4.2 Défense

Pour un non initié, la défense se résume majoritairement à coller ses pions à ceux de l'adversaire. Cependant avec un peu de logique il est évident que cette technique est inefficace due à la forme hexagonale des pions permettant toujours à l'adversaire de continuer sa chaîne en contournant aisément le pion adverse. Une méthode plus efficace est d'avoir quelques pions positionnés vers un bord adverse afin d'avoir un peu de marge de manoeuvre pour contrer ses mouvements. De plus la stratégie d'attaque majoritaire étant celle des pions pseudo-liés, il faut trouver une technique de défense adaptée à cette attaque. La plus efficace se révèle être la stratégie du "triangle" permettant de contrer n'importe quel mouvement de l'adversaire. Il faut cependant anticiper le chemin de pions pseudo-lié et ainsi forcer l'adversaire à faire un détour.

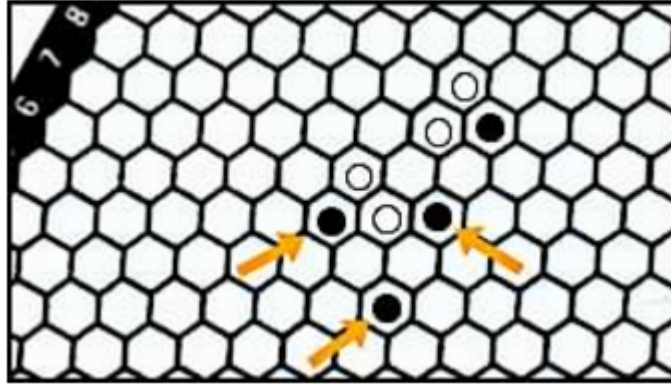


FIGURE 3 – Disposition de défense

3 État de l'art - partie IA

3.1 Bibliographie

Les sources de cet état de l'art sont les suivantes :

<https://www.supinfo.com/articles/single/3447-introduction-intelligence-artificielle-travers-jeu-hex-java-partie-1-2>

https://fr.wikipedia.org/wiki/Apprentissage_profond

<http://images.math.cnrs.fr/Le-jeu-de-Hex.html?lang=fr>

3.2 Ordinateur intelligent

3.2.1 Algorithme MinMax

L'algorithme MinMax est un algorithme adapté aux jeux se jouant à tour de rôle. Il va au début de son tour, calculer le meilleur coup à jouer en fonction des pions déjà présents sur le plateau. Pour cela, il va construire un « arbre de jeu » qui lui permettra de visualiser plusieurs évolutions de jeu possibles. Dans chacune de ces évolutions possibles, il va calculer la force de chaque coup possible de son adversaire. Il saura donc quel est le meilleur coup à jouer en fonction de la valeur attribuée à chaque case. En effet, lors de la simulation, l'algorithme va partir du principe que l'adversaire jouera le cas le plus désavantageux pour lui et va donc poser un pion sur la case à partir de laquelle l'adversaire lui sera le moins défavorable.

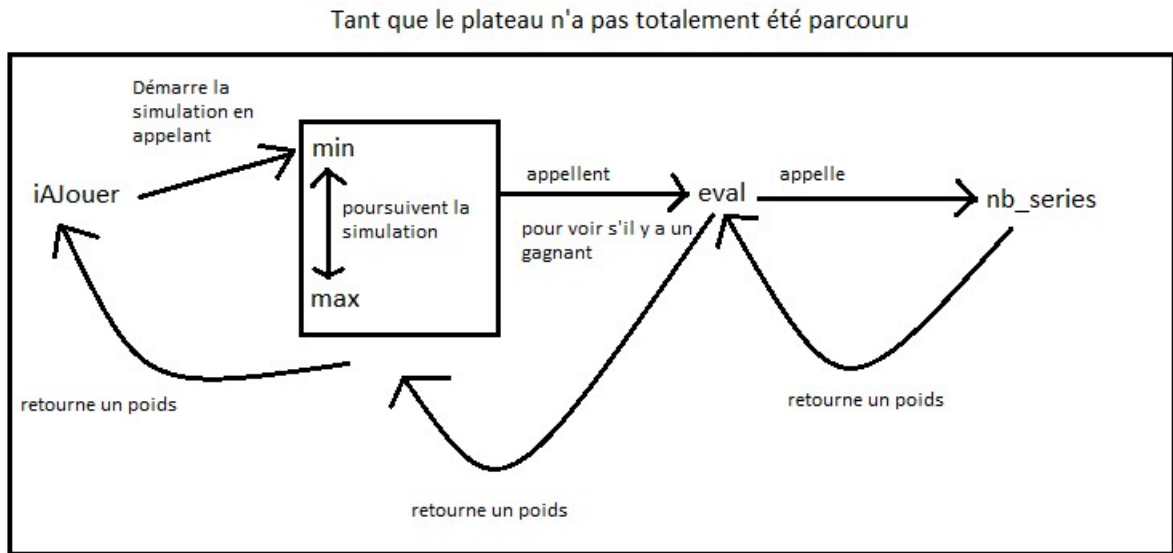


FIGURE 4 – algorithme MinMax

3.2.2 Algorithme offensif

Un autre moyen de coder une intelligence artificielle pour le jeu hex et de faire un algorithme n’essayant que de gagner la partie sans se soucier des coups adverses et sans essayer de les bloquer. Étant donné que le jeu est résolu, il est possible de perfectionner ce type d’IA et de la rendre invincible si elle joue en premier. Cependant la résolution du jeu n’a à ce jour été explicitée que pour des plateaux de taille inférieure ou égale à 9x9. C’est Jin Yang, informaticien à l’université du Manitoba au Canada, qui a réussi à détailler des stratégies gagnantes pour le premier joueur pour des plateaux allant jusqu’à la taille 9x9, mais cette dernière requiert la description de 715 « tactiques locales ».

3.3 Intelligence artificielle

3.3.1 Deep learning

L’apprentissage profond (en anglais deep learning) est un ensemble de méthodes d’apprentissage automatique tentant de modéliser avec un haut niveau d’abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. C’est grâce au Deep learning qu’a été

mis au point alphGo, l'intelligence artificielle qui a battue le champion du monde de Go en mars 2016.

4 Conception du code

4.1 Notre jeu

4.2 Création du plateau

Pour la création de notre plateau, nous avons tout d'abord pensé à créer 2 structures :

- Une structure Case comprenant la valeur de la case (0 si personne n'a joué sur la case, -1 si c'est le joueur violet et 1 si c'est le joueur orange) et une structure plateau contenant 121 cases . Cependant, nous nous sommes rendus compte en travaillant que nous n'avions pas besoin de tout cela et nous avons opté pour une autre solution.
- Modéliser le plateau par un tableau de 121 entiers, avec pour valeur 0 si la case est vide, -1 si le joueur violet a joué dessus et 1 si le joueur orange a joué dessus.

4.3 Création des règles du jeu

Nous avons implémenté les règles du jeu à l'aide de 3 fonctions :

1. La fonction ajouterPiece, prenant comme arguments la position où on veut ajouter notre pièce, le plateau de jeu, le joueur en train de jouer, l'écran et la pile , qui nous permet de déterminer si on peut ajouter la pièce à l'endroit souhaité (ie la case où on veut jouer est vide) et qui l'ajoute si c'est le cas. Cette fonction renvoie un entier, 0 si la pièce n'a pas pu être ajoutée et 1 si elle l'a été ;
2. La fonction Parcours, prenant comme arguments le joueur actuel, le plateau, un tableau de 121 entiers et la position à partir de laquelle on veut parcourir le tableau. Cette fonction récursive va, en partant de la case donnée en paramètre, se propager à toutes les cases adjacentes où le joueur passé en paramètre à un pion. Chaque case ainsi parcourue sera mise à jour dans le tableau d'entier mis en paramètre.

3. la fonction `finPartie` prenant comme arguments le joueur actuel et le plateau. Cette fonction va, en fonction du joueur actuel, appliquer la fonction `parcours` 11 fois (les 11 cases de la première ligne si le joueur est violet et les 11 cases de la première colonne si le joueur est orange). La fonction va parcourir le tableau à partir de ces 11 cases, puis elle regarde les 11 cases opposées. Si au moins l'une d'entre elles a été touchée par `parcours`, c'est que la partie est gagnée. La fonction rend un entier, 0 si la partie n'est pas finie et 1 si elle est finie.

Nous avons donc grâce à ajouter pièce l'assurance que les pièces ne peuvent être ajoutées qu'aux bons endroits, et grâce à `parcours` et `fin de partie` la condition de victoire de notre partie.

4.4 Création de la fenêtre

Grâce à la bibliothèque `SDL` nous avons pu créer une interface graphique pour notre jeu. Celui-ci s'ouvre sur un menu qui nous permet de naviguer entre les différentes options, une fois les options choisies nous affichons le plateau de jeu, s'actualisant à chaque coup joué. Le menu reste visible et utilisable pendant la partie.

4.5 Création du Makefile

Nous avons utilisé un `Makefile` pour compiler et rendre exécutable notre projet. Le `Makefile` est pour nous un moyen d'éviter de taper des lignes de commandes répétitives et complexes dans le `bash`. Pour compiler notre projet nous devons tout d'abord compiler tous les fichiers `*.c`, suite à cela il faut créer exécutable. Ces commandes sont toujours longues et identiques ainsi, nous les écrivons dans le `Makefile` pour, par la suite, ne plus avoir qu'à appeler le `Makefile` pour les effectuer. De plus nous avons rajouté deux fonctionnalités, "`make clean`" et "`make mrpropre`", "`clean`" permet de nettoyer le dossier avant de compiler. "`mrpropre`" est une variante de "`clean`" incluant en plus le fichier exécutable.

4.6 Utilisation de Shell

Le module s'intitulant "`C & Shell`" nous avons utilisé un peu de `Shell` pour notre code. En effet, la fonction permettant d'ajouter une case de couleur sur la case cliquée, dans "`ajouter.c`", a été faite grâce à un script `Shell`. Nous

voulions une variable pour chaque case du tableau (donc 121 variables car le tableau est en 11x11) que nous avons générée grâce au Shell (nous aurions également pu faire un tableau de 121 valeurs). Puis, il fallait affecter à chaque variable l'image de la case de la couleur correspondante (ie lors d'un clique ajouter la case de sa couleur à l'emplacement du clic). Ainsi, il y a trois cas : orange, violet et blanc (pour la fonction annuler). Ce qui nous mène à trois cas par cases. Nous avons donc généré ces lignes de code grâce à un script Shell.

```
#!/bin/sh
cd #HOME/Bureau/Shell
cd tt
echo > tes
for i in `seq 0 120`
do
echo 'if (pos=='$i' && strcmp(joueuractuel, "rouge") == 0) {\n    case'$i' = load_image("rouge.bmp"); \n
    apply_surface(xPos+49+(pos%11)*26+(pos/11)*13,yPos+57+(pos/11)*19,case'$i',screen); \n
    modif=1;\n
} \n
if (pos=='$i' && strcmp(joueuractuel, "bleu") == 0) {\n
case'$i' = load_image("bleu.bmp"); \n
    apply_surface(xPos+49+(pos%11)*26+(pos/11)*13,yPos+57+(pos/11)*19,case'$i',screen); \n
    modif=1; \n
} \n
if (pos=='$i' && strcmp(joueuractuel, "blanc") == 0) {\n
case'$i' = load_image("blanc.bmp"); \n
    apply_surface(xPos+49+(pos%11)*26+(pos/11)*13,yPos+57+(pos/11)*19,case'$i',screen); \n
}' >> tes
done
```

FIGURE 5 – Script Shell

4.7 Ordinateur intelligent

L'ordinateur intelligent a été effectué via une disjonction de cas. On ne peut donc pas considérer cela comme une "Intelligence Artificielle" (il n'apprend pas de son adversaire et de ses erreurs). Pour gérer les réactions de l'ordinateur en fonction du jeu du joueur, il a fallu faire de nombreux cas qui ont été testés en jeu, puis améliorés en fonction des résultats obtenus. Pour faire ces tests il a fallu demander à différentes personnes plus ou moins expérimentées dans le jeu. On peut ajouter que l'ordinateur joue en position défensive : il ne cherchera pas à gagner sauf s'il ne lui manque qu'une case pour gagner.

5 Graphiques

Pour ce qui est de la partie graphique, nous avons trouvé le background sur un site libre de droits. Pour le reste, tout a été fait par nos soins. Nous avons créé chaque pion, le plateau, le menu, les messages et les couleurs sur Photoshop. Ce travail est une part très importante de notre projet. En effet, nous voulions faire un jeu à la fois ergonomique et visuellement agréable. Le temps consacré à la création des images, ainsi qu'à leurs retouches successives, a donc été très élevé. Cependant, nous pensons que notre objectif sur ce point a été atteint et que ce temps n'a donc pas été perdu. L'utilisateur peut naviguer à sa guise dans le menu, que ce soit avant la partie, pendant la partie, voire même celle-ci terminée (sans avoir besoin de relancer le jeu). De plus les couleurs utilisées (orange, violet et marron) sont les couleurs de Télécom Nancy.

6 Qualité du programme

6.1 Complexité

La Complexité de notre programme dépend de la complexité de parcours et de ajouterpiece. Finpartie utilise parcours dans des boucles for, et l'appelle racine de taille de plateau fois. Addcase et parcours sont toutes deux en grand O de la taille de notre plateau. Ajouterpiece est aussi en grand O de taille de plateau puisqu'elle appelle une fois Addcase. La complexité de notre programme est donc grand O de $(\text{taille du plateau}) * (\text{racine de taille du plateau})$.

6.2 Rapidité exécution

Le jeu tourne de manière relativement fluide. Cela permet une interaction efficace entre le joueur et l'interface pour un meilleur plaisir de jeu.

6.3 Tests

Nous avons effectué des tests "à la main" en jouant de très nombreuses fois contre l'IAs ou entre nous. Cela nous a permis de découvrir plusieurs bugs que nous avons su corriger, et certains que nous avons dû contourner (cf bug z).

7 Outils de GdP

7.1 Google Drive

Pour partager certains fichiers, surtout dans les débuts du projet, nous avons utilisé un Google Drive "Projet C". Nous avons ainsi pu partager efficacement des fichiers et les modifier en parallèle.

7.2 Messenger

Pour communiquer de manière efficace nous avons utilisé Messenger. En effet, nous avons un groupe de discussion "Projet de C".

7.3 Slack

En complément de Messenger, nous avons également un Slack, même si celui-ci nous a été beaucoup moins utile que Messenger.

7.4 Git

Git a été un outil essentiel à notre projet. En effet, il nous a permis de se transmettre des fichiers et garder des sauvegardes de manière très efficace et rapide.

7.5 LaTeX

LaTeX est un outil de traitement de texte des plus prisés. Il permet d'obtenir un rendu de très bonne qualité. Ainsi, nous avons rédigé tous nos comptes-rendus, le rapport, et le manuel en format .Tex.

7.6 Poste de travail

Pour travailler, nous avons tous un PC personnel. Cependant, n'ayant pas réussi à installer SDL dessus, nous avons utilisé les PC de l'école, sur Linux, pour le code. Tout ce qui ne touche pas au code (graphique, GdP, Manuel, ...) a été majoritairement effectué chacun de son côté, sur des PC Windows.

8 Matrice SWOT



FIGURE 6 – Matrice SWOT

9 GANTT

9.1 Rendu

Nom de la Tâche		15-mars-2017	20-mars-2017	27-mars-2017	3-avr-2017	10-avr-2017	17-avr-2017	24-avr-2017	1-mai-2017	8-mai-2017	15-mai-2017	22-mai-2017	29-mai-2017						
		L	M	J	V	S	D	L	M	J	V	S	D	L	M	J	V	S	D
Section 0 - Gestion de Projet																			
<div><div><div>Section 0 - Gestion de projet (le projet)</div><div><div>Sous-tâche 1 - Elements de gestion de projet</div><div><div>Sous-tâche 2 - SWOT</div><div>Sous-tâche 3 - Chartes conceptuelles</div><div>Sous-tâche 4 - GANTT</div><div>Sous-tâche 5 - RACI</div><div>Sous-tâche 6 - Etat de l'art</div></div></div></div></div>																			
Section 1 - Moteur de jeu minimaliste																			
<div><div><div>Section 1 - Moteur de jeu minimaliste</div><div><div>Sous-tâche 1 - Papiers Graphique</div><div><div>Sous-tâche 2 - Plateau</div><div>Sous-tâche 3 - Papiers</div><div>Sous-tâche 4 - Papiers intermédiaires</div><div>Sous-tâche 5 - Papiers d'essai</div><div>Sous-tâche 6 - Papiers d'essai et de jeu</div><div>Sous-tâche 7 - Papiers Fonctionnels</div><div>Sous-tâche 8 - Implémenter les règles</div><div>Sous-tâche 9 - Choix du premier joueur</div><div>Sous-tâche 10 - Règles du plateau</div></div></div></div></div>																			
Section 2 - Test																			
<div><div><div>Section 2 - Test</div><div><div>Sous-tâche 1 - Test de la partie graphique</div><div>Sous-tâche 2 - Test de la partie intermédiaire</div><div>Sous-tâche 3 - Test de la partie fonctionnelle</div></div></div></div>																			
Section 3 - Intelligence Artificielle																			
<div><div><div>Section 3 - Intelligence Artificielle</div><div><div>Sous-tâche 1 - IA de base</div><div>Sous-tâche 2 - Niveau de difficulté</div></div></div></div>																			
Section 4 - Test IA																			
<div><div><div>Section 4 - Test IA</div><div><div>Sous-tâche 1 - Tester les différents niveaux</div><div>Sous-tâche 2 - Vérifier la difficulté de jouer l'IA officielle</div></div></div></div>																			
Section 5 - Rendu																			
<div><div><div>Section 5 - Rendu</div><div><div>Sous-tâche 1 - Vérifier que le code est bien</div><div>Sous-tâche 2 - Vérifier que le code est bien</div><div>Sous-tâche 3 - Rédiger le rapport</div></div></div></div>																			

9.2 Tableur





































En danger	Nom de la Tâche	Date de Début	Date de Fin	Assignée à	Durée	Commentaires
	 Section 0 - Gestion de Projet	09/03/17	24/03/17		12j	
	 Sous-tâche 1 - Eléments de gestion de projet	09/03/17	14/03/17	Tous	4j	
	Sous-sous-tâche 2 - SWOT	09/03/17	14/03/17	Raphael Ohanian	4j	
	Sous-sous-tâche 3 - Cartes conceptuelles	09/03/17	14/03/17	Julien - Maxime	4j	
	Sous-sous-tâche 4 - GANTT	09/03/17	14/03/17	Tous	4j	
	Sous-sous-tâche 5 - RACI	09/03/17	14/03/17	Julien Derigny	4j	
	Sous-tâche 2 - Etat de l'art	09/03/17	24/03/17	Tous	12j	
	 Section 1 - Moteur de jeu minimaliste	14/03/17	13/04/17		23j	
	 Sous-tâche 1 - Partie Graphique	14/03/17	13/04/17	Raphael Ohanian	23j	
	Sous-sous-tâche 1 - Plateau	14/03/17	13/04/17	Raphael Ohanian	23j	
	Sous-sous-tâche 2 - Pions	14/03/17	13/04/17	Raphael Ohanian	23j	
	 Sous-tâche 2 - Partie Interface	14/03/17	13/04/17	Julien Derigny	23j	
	Sous-sous-tâche 1 - Page d'accueil	14/03/17	13/04/17	Julien Derigny	23j	
	Sous-sous-tâche 2 - Menu pendant le jeu	14/03/17	13/04/17	Julien Derigny	23j	
	 Sous-tâche 3 - Partie Fonctionnalités	14/03/17	13/04/17	Maxime Rièrè	23j	
	Sous-sous-tâche 1 - Implémenter les règles	14/03/17	13/04/17	Maxime Rièrè	23j	
	Sous-sous-tâche 2 - Choix du premier joueur	14/03/17	13/04/17	Maxime Rièrè	23j	
	Sous-sous-tâche 3 - Règle du gâteau	14/03/17	13/04/17	Maxime Rièrè	23j	
	 Section 2 - Test	13/04/17	20/04/17		6j	
	Sous-tâche 1 - Test de la partie graphique	13/04/17	20/04/17	Tous	6j	
	Sous-tâche 2 - Test de la partie interface	13/04/17	20/04/17	Tous	6j	
	Sous-tâche 3 - Test de la partie Fonctionnalités	13/04/17	20/04/17	Tous	6j	
	 Section 3 - Intelligence Artificielle	20/04/17	18/05/17		21j	
	Sous-tâche 1 - IA de base	20/04/17	18/05/17	Tous	21j	
	Sous-tâche 2 - Niveaux de difficulté	20/04/17	18/05/17	Tous	21j	
	 Section 4 - Test IA	18/05/17	23/05/17		4j	
	Sous-tâche 1 - Tester les différents niveaux	18/05/17	23/05/17	Tous	4j	

FIGURE 7 – Tableur du GANTT

10 Cartes Conceptuelles

10.1 Carte Conceptuelle du projet

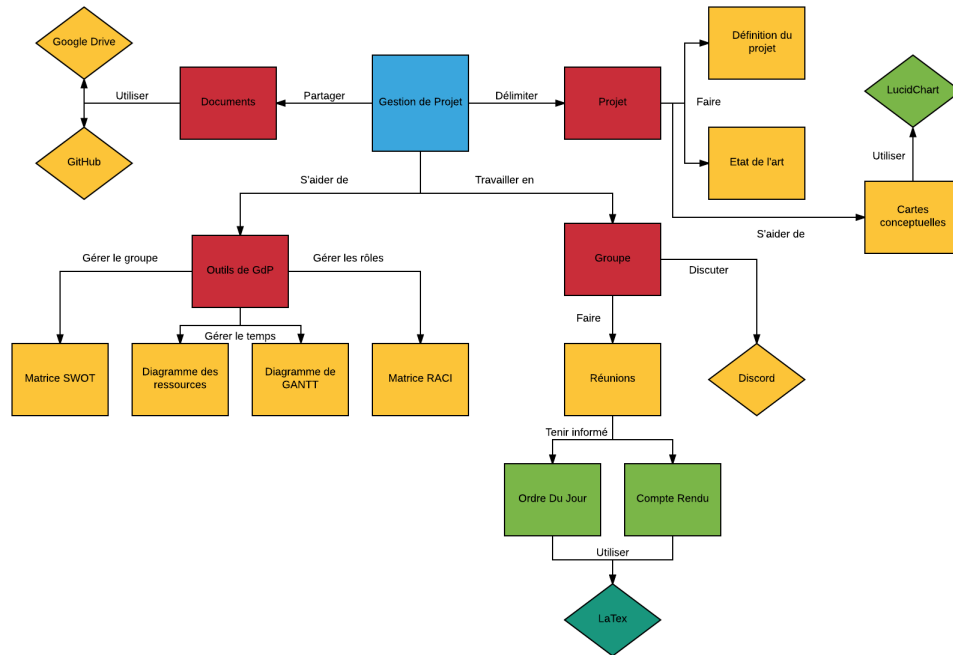


FIGURE 8 – Projet

10.2 Carte Conceptuelle du jeu

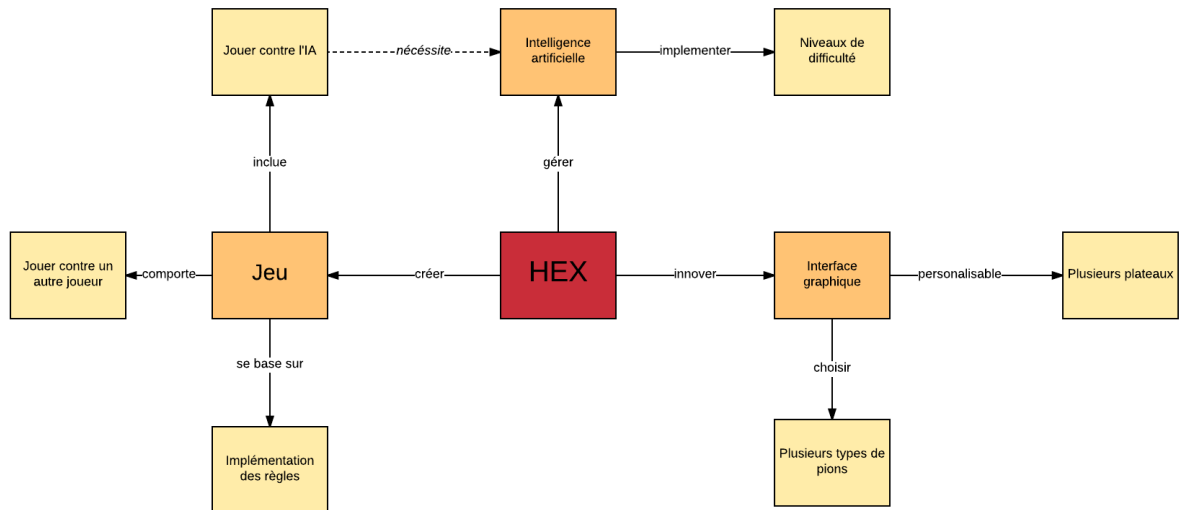


FIGURE 9 – Jeu

11 Difficultés

11.1 Estimation du temps de travail

Le projet nécessite l'utilisation d'une bibliothèque graphique. Or nous n'avions aucune compétence dans ce domaine au début du projet. Il nous a donc premièrement fallu apprendre à utiliser cette bibliothèque (SDL).

11.2 Respect du Gantt

L'installation de la SDL nous a posé beaucoup de problèmes. Ainsi dès le début du projet nous avons eu un décalage avec le GANTT que nous n'avons pas pu rattraper.

11.3 Plages horaire de travail

Se retrouver tous ensemble pour coder en salle PI n'a pas été un très gros problème. Cependant, la réduction du temps pour le projet nous à entraînée dans un "rush" car la superposition du rendu du projet de POO et des deux partiels a réduit considérablement notre temps de travail disponible.

11.4 Gestion des couleurs

Le plateau de départ que nous avons utilisé était le plateau de base avec un joueur bleu et un joueur rouge. Nous avons cependant choisi de changer cette idée en cours de route et nous avons pris le parti de prendre les couleurs de l'école. Malheureusement cette transition a entraîné des difficultés de cohérence : nous avons gardé les valeurs rouge et bleu pour joueur actuel, ce qui a forcé quelques confusions. Cela à somme toute été vite réglé et nous avons laissé rouge et bleu puisque nous n'avions plus de soucis avec eux.

11.5 Le bug Z

Nous avons rencontré un bug que nous n'avons pas compris. Lors du lancement du jeu, quand nous affrontons l'IA, la partie se termine après le premier coup du joueur et le premier coup de l'IA. Le jeu annonce que L'IA a gagné et nous ne savons pas pourquoi. En relançant une partie contre l'IA sans fermer le jeu, la partie se déroule normalement. Ce problème n'est jamais

apparu en jouant contre un joueur humain, mais est systématique lors de la première partie au lancement du jeu contre l'IA. Ne sachant pas d'où vient ce bug, nous le contournons en utilisant la variable z : nous initialisons la variable z à 1 et elle passe à 0 après le tour de l'IA. Nous l'utilisons ensuite dans le test de victoire, pour pouvoir regarder si la partie se termine, il faut d'abord que z vaille 0 ie que l'IA ait déjà joué un coup. Grâce à ce contournement, le bug n'est plus visible et nous pouvons jouer contre l'IA sans problème, mais nous savons qu'il existe toujours.

12 Enrichissements

Grâce à ce projet nous avons appris à coder une interface graphique et à interagir avec celle-ci.

12.1 Intérêt

Ce projet nous a ouvert au langage C, de manière un peu plus poussée que les TD que nous avions en cours. Cela a été un bon complément à notre module de C.

12.2 Interface graphique

L'interface graphique ne se résume pas juste à l'affichage d'un plateau. En effet, il faut premièrement créer le plateau, les pions, les menus. Nous les avons tous fait nous-même avec photoshop. Nous avons donc perfectionné nos compétences en photoshop. Puis, il a fallu gérer l'affichage grâce à SDL. Enfin, nous avons créé des interactions avec les affichages : cliquer sur "New game" ouvre un autre menu, cliquer sur une case affiche un pion dans celle-ci,

12.3 Groupe de travail

Pour ce projet, le choix du groupe nous a été laissé. Ainsi, la cohérence dans notre groupe a été présente, nous permettant, lorsqu'il l'a fallu, se mettre un coup de pression et accélérer de manière significative notre travail.

12.4 Démarrer de rien

Un aspect de ce projet qui nous a beaucoup plu, en comparaison avec les autres projets que nous avons eu l'occasion de faire, est le fait de commencer de rien. Nous n'avions aucune contrainte prédéfinie dans un squelette donné. La liberté que nous avons eue a pu laisser place à notre imagination nous permettant de faire un jeu ergonomique et design.

12.5 Maîtrise de Git

Au début nous avions tous un dossier différent dans le Git. Puis, nous avons commencé à tous travailler dans le même dossier. Nous avons eu beaucoup de merge à gérer. Le premier fut un peu laborieux, mais les suivants se firent très facilement.

13 Temps passé sur le projet

	J.Derigny	R.Ohanian	M.Rière
Gestion de projet			
Élaboration des documents de Gestion de Projet	6h	1h	4h
Réunions	10h	10h	10h
Stand-Up Meetings	2h	2h	2h
Rédaction des comptes rendus	3h30	2h	0h
Rédaction des ordres du jour	1h	1h	0h
Documentation			
Apprentissage de SDL	10h	10h	5h
Relecture du code des partenaires	7h	7h	7h
Rédaction du Manuel de Jeu	0h	2h	0h
Rédaction des rapports	2h	3h	2h
Implantation - Logique Interne			
Règles	0h	2h	15h
Interface graphique	20h	20h	0h
Menu	12h	0h	0h
Makefile	5h	0h	0h
Shell	0h	1h	1h
Ordinateur Intelligent	0h	20h	10h
Total Global			
Total GdP, Doc, Implantation et Tests	78h30	81h	56h

14 Conclusion

Ce projet nous a apporté de nombreuses connaissances. Nous avons visé un jeu ergonomique et design. Nous avons un ordinateur qui va répondre au coup du joueur sans chercher à créer son propre réseau de pions adjacents. Les fonctionnalités que nous aurions aimé ajouter sont les suivantes :

- Règle du gâteau (lors de son premier tour, le joueur 2 peut choisir de prendre possession du pion du joueur 1. Ce sera ensuite au tour du joueur 1. Cela permet d'équilibrer le jeu en empêchant le joueur 1 de prendre une case "trop forte", car il encourt le risque d'échanger son pion avec son adversaire)
- Ajout d'interactions sonores

15 Sources

- <https://www.libsdl.org/>
- <http://loka.developpez.com/>
- <https://openclassrooms.com/>
- <http://stackoverflow.com/>
- <http://www.supinfo.com/>
- <https://wikipedia.org/>
- <http://hdw.eweb4.com/out/465985.html>

16 Annexe

Vous retrouverez en annexe les compte-rendus des réunions que nous avons faites.

- CR1
- CR2
- CR3
- CR4
- CR5
- CR6
- Manuel utilisateur

Compte-rendu de réunion

Minutes for Jeudi 9 mars 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

Ordre du jour

- Choisir les éléments de GdP à réaliser
- Idées pour le jeu

Elements de Gdp

Nous avons décidé quels éléments de gestion de projet nous allons réaliser pour ce projet, à savoir une matrice SWOT, une matrice RACI, un GANTT, un diagramme des ressources et des cartes conceptuelles.

Idées pour le jeu

Nous avons mis en commun les quelques sites déjà trouvés à propos du jeu. De plus, nous nous sommes familiarisé avec le gameplay et l'esprit du jeu.

A faire pour la prochaine réunion

- Carte Conceptuelle du jeu (créer un compte sur lucidchart pour faire un document partagé)
- RIERE M.
- Faire une matrice SWOT - OHANIAN R.
- Faire une matrice RACI - DERIGNY J.
- Commencer un état de l'art - Tous
- Prochaine réunion jeudi 14 mars à 14h

Compte-rendu de réunion

Minutes for Jeudi 14 mars 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

Ordre du jour

- Retour sur ce qu'il y avait à faire
- Faire un GANTT

Retour ce le travail a faire

Nous avons maintenant deux cartes conceptuelles, l'une sur la Gdp plutôt complète, l'autre sur le jeu en lui-même avec peu d'idées mais nous n'en trouvons aucune autre. De plus, nous avons une matrice SWOT plutôt complète, malgré un manque de point dans la case menace externe. La matrice RACI n'est pas finie, nous y reviendrons la semaine prochaine.

Faire le GANTT

Tout d'abord, nous avons listé toutes les étapes possibles pour ce projet. Puis nous les avons incrémenté dans un logiciel pour créer des GANTT, et avons réparti les tâches et durées.

A faire pour la prochaine réunion

- Faire une matrice RACI - DERIGNY J.
- Avancer un état de l'art - Tous
- Commencer le code - Tous
- Prochaine réunion jeudi 23 mars à 14h

Compte-rendu de réunion

Minutes for Jeudi 30 mars 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

Ordre du jour

- Retour sur ce qu'il y avait à faire
- Travail à faire

Retour sur le travail à faire

Julien à fait un GANTT.

Travail fait

Nous avons choisi de coder en utilisant la librairie SDL. Nous avons également commencé à faire une base des fonctions qui seront à faire et du déroulement d'une partie du point de vue algorithmique. Ce fichier est sur le drive.

A faire pour la prochaine réunion

- Avancer un état de l'art - Tous
- Commencer le code - Tous

Compte-rendu de réunion

Minutes for Jeudi 13 avril 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

Ordre du jour

- Avancement

Avancement

Nous avons des difficultés à installer SDL. Cela retarde nous programme par rapport au GANTT.

- Avancer un état de l'art - Tous
- Coder - Tous

Compte-rendu de réunion

Minutes for Jeudi 28 avril 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

Ordre du jour

- Avancement

Avancement

Suite aux difficultés pour l'installation de la SDL, nous avons commencé à faire du pseudo code.

- Avancer un état de l'art - Tous
- Coder - Tous

Compte-rendu de réunion

Minutes for Jeudi 11 mai 2017

Present: DERIGNY J. RIERE M. OHANIAN R.

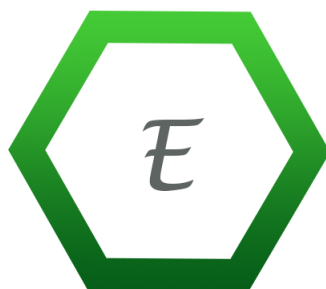
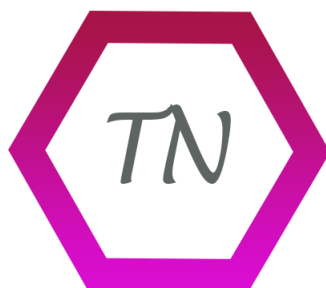
Ordre du jour

- Avancement

Avancement

Nous avons finalement décidé de coder sur les PC de l'école, où la SDL est déjà installée. Le projet forge a été créé, et des push ont été faits. En passant par les PC de l'école, nous nous forçons ainsi à utiliser l'éditeur de texte et à faire des makefile. Nous aurons ainsi moins de difficultés à rendre le projet selon les conditions imposées (en passant par code blocks il nous aurait ensuite fallu faire des makefile à partir du projet, ce qui semble assez compliqué d'après les retours que nous avons eu des deuxièmes années).

- Avancer un état de l'art - Tous
- Coder - Tous



MANUEL

Table des matières

1	Avertissement sur l'épilepsie	3
2	Précautions à prendre lors de l'utilisation d'un jeu vidéo	3
3	Comment démarrer le jeu	4
4	Lancer une partie	4
5	Jouer au jeu	5
6	Règle et victoire	6
7	Menu en jeu	6
8	Touches clavier	7
8.1	Touche Echap	7
8.2	Règle du gâteau	7
8.3	Annuler son coup	7
9	Gestion du son	8
10	A vous de jouer !	8

1 Avertissement sur l'épilepsie

Certaines personnes sont susceptibles de faire des crises d'épilepsie comportant, le cas échéant, des pertes de conscience à la vue, notamment, de certains types de stimulations lumineuses fortes : succession rapide d'images ou répétition de figures géométriques simples, d'éclairs ou d'explosions. Ces personnes s'exposent à des crises lorsqu'elles jouent à certains jeux vidéo comportant de telles stimulations, alors même qu'elles n'ont pas d'antécédent médical ou n'ont jamais été sujettes elles-mêmes à des crises d'épilepsie. Si vous-même ou un membre de votre famille avez déjà présenté des symptômes liés à l'épilepsie (crise ou perte de conscience) en présence de stimulations lumineuses, consultez votre médecin avant toute utilisation. Les parents se doivent également d'être particulièrement attentifs à leurs enfants lorsqu'ils jouent avec des jeux vidéo. Si vous-même ou votre enfant présentez un des symptômes suivants : vertige, trouble de la vision, contraction des yeux ou des muscles, trouble de l'orientation, mouvement involontaire ou convulsion, perte momentanée de conscience, il faut cesser immédiatement de jouer et consulter un médecin.

2 Précautions à prendre lors de l'utilisation d'un jeu vidéo

En tout état de cause, veuillez respecter les règles suivantes lors de l'utilisation d'un jeu vidéo :

- évitez de jouer si vous êtes fatigué ou si vous manquez de sommeil ;
- assurez-vous que vous jouez dans une pièce bien éclairée ;
- en cours d'utilisation, faites des pauses de dix à quinze minutes toutes les heures.

3 Comment démarrer le jeu

Pour démarrer, ouvrir le bash dans le dossier avec le Makefile. Une fois celui-ci ouvert, taper la requête "make" puis "./project". Le jeu est maintenant ouvert. Amusez vous bien !

Si l'envie vous prenait de vouloir "nettoyer" votre dossier, deux options s'offrent à vous :

- "make mrpropre" permet de supprimer tous les fichiers compilés (les fichiers *.o) mais laisse le fichier exécutable, il est donc toujours possible de lancer le jeu.
- "make clean" supprime les fichiers compilés et supprime en plus le fichier exécutable.

4 Lancer une partie

Une fois le jeu lancé, une page avec un menu s'affiche.



FIGURE 1 – Menu

Cliquez sur "New game" et choisissez si vous voulez jouer à 2, ou contre l'ordinateur. Chaque joueur (si vous jouez à 2) pourra ensuite choisir sa couleur. Le joueur 1 choisit en premier.

Suite à cette étape, la partie se lance.

5 Jouer au jeu

Une fois le plateau affiché, le joueur 1 clique sur une case. Son pion s'affichera.



FIGURE 2 – Plateau de jeu

6 Règle et victoire

Au départ, le plateau est vide. Le joueur 1 a le privilège de jouer le premier. À tour de rôle, chacun des joueurs pose un pion de sa couleur sur la case de son choix - à la seule condition que cette case ne soit pas déjà occupée. Une fois posé, un pion ne peut plus être déplacé. Si deux pions se trouvent dans des cases ayant un côté commun, on dit qu'ils sont adjacents ou "liés". Une chaîne de pions liés est appelée "chaîne continue". La partie s'arrête quand un joueur a relié entre eux, par une chaîne continue de pions lui appartenant, les deux bords de sa couleur. Il ne peut y avoir de partie nulle, puisque l'un des deux adversaires finit toujours par y parvenir.

7 Menu en jeu

Pendant la partie, le menu change légèrement. Vous retrouverez l'onglet "New game" et "Quit". Vous trouverez de plus le joueur actif.



FIGURE 3 – Menu en jeu

8 Touches clavier

8.1 Touche Echap

La touche Echap correspond au retour en arrière. Si le menu est dans son état principal, Echap fermera la fenêtre.

8.2 Règle du gâteau

Lors de son premier tour, le joueur 2 peut choisir de prendre possession du pion du joueur 1. Ce sera ensuite au tour du joueur 1. Cela permet d'équilibrer le jeu en empêchant le joueur 1 de prendre une case "trop forte", car il encourt le risque d'échanger son pion avec son adversaire. C'est une touche cachée, il faudra appuyer sur "space".

8.3 Annuler son coup

Il y a une fonctionnalité cachée : annulation du dernier coup. Pour l'utiliser, appuyer sur la touche "delete" de votre clavier et le tour est joué!

9 Gestion du son

Vous avez peut-être remarqué lors du lancement du jeu le petit fond sonore ? Si la folie vous prend de couper ce magnifique son, rien de plus simple : faites un clic sur l'icône son en bas à droite de votre fenêtre (à côté de l'heure), une barre horizontale apparaîtra, vous n'avez plus qu'à la glisser vers le gauche !

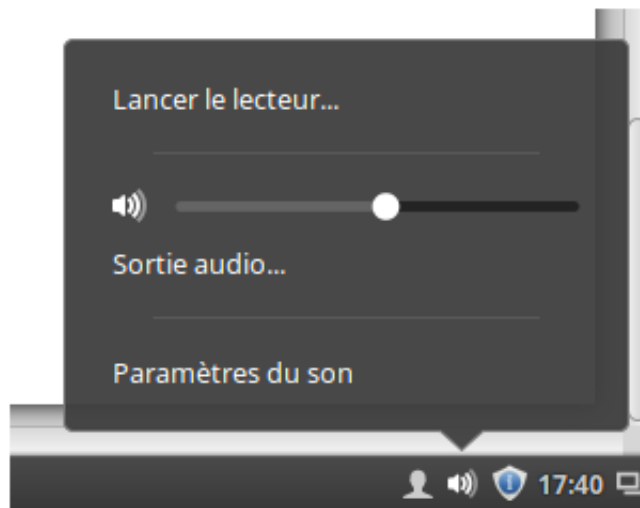


FIGURE 4 – Réduire le son du jeu

10 A vous de jouer !

Seul ou à plusieurs, amusez-vous bien sur HEX !

PS : Vous trouverez différents "Easter eggs" dans le jeu. A vous de les trouver !