

|  |
| --- |
| **Conditions d’utilisations :** SUPINFO International University vous permet de partager ce document. Vous êtes libre de :   * Partager — reproduire, distribuer et communiquer ce document * Remixer — modifier ce document   **A condition de respecter les règles suivantes :**  Indication obligatoire de la paternité — Vous devez obligatoirement préciser l’origine « SUPINFO » du document au début de celui-ci de la même manière qu’indiqué par SUPINFO International University – Notamment en laissant obligatoirement la première et la dernière page du document, mais pas d'une manière qui suggérerait que SUPINFO International University vous soutiennent ou approuvent votre utilisation du document, surtout si vous le modifiez. Dans ce dernier cas, il vous faudra obligatoirement supprimer le texte « SUPINFO Official Document » en tête de page et préciser notamment la page indiquant votre identité et les modifications principales apportées.  En dehors de ces dispositions, aucune autre modification de la première et de la dernière page du document n’est autorisée.  **NOTE IMPORTANTE :** Ce document est mis à disposition selon le contrat CC-BY-NC-SA Creative Commons disponible en ligne http://creativecommons.org/licenses ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA modifié en ce sens que la première et la dernière page du document ne peuvent être supprimées en cas de reproduction, distribution, communication ou modification. Vous pouvez donc reproduire, remixer, arranger et adapter ce document à des fins non commerciales tant que vous respectez les règles de paternité et que les nouveaux documents sont protégés selon des termes identiques. Les autorisations au-delà du champ de cette licence peuvent être obtenues à support@supinfo.com.  © SUPINFO International University – EDUCINVEST - Rue Ducale, 29 - 1000 Brussels Belgium . www.supinfo.com |

Table of contents

1 Group summary 4

1.1 Group members 4

2 Project report 4

3 solution manual 6

4 technical documentation 6

# Group summary

## Group members

Campus: Strasbourg

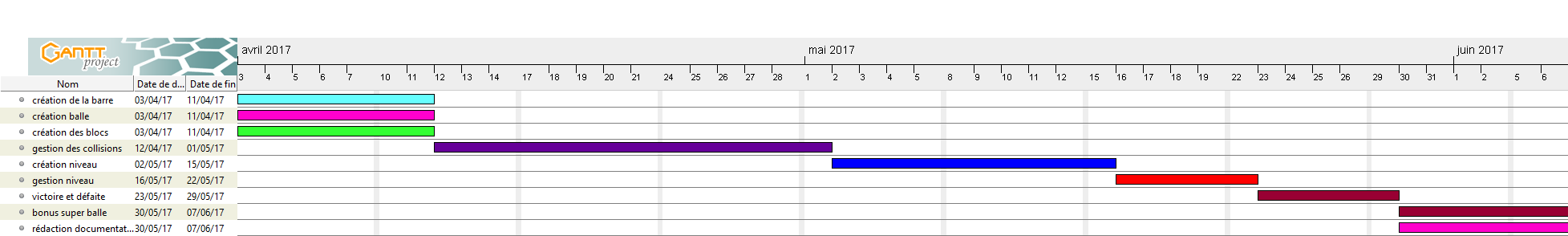
Class: B2

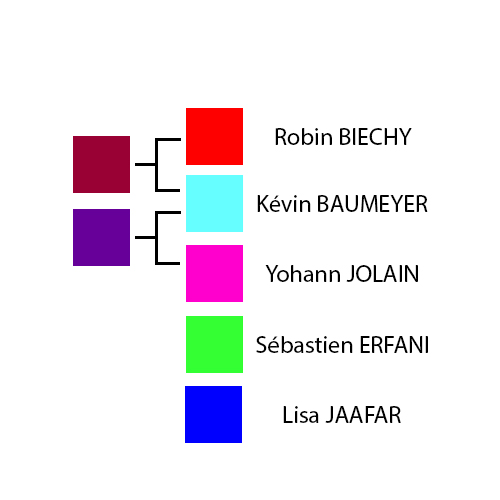
|  |  |  |  |
| --- | --- | --- | --- |
| ID Open Campus | Last Name | First Name | Photo |
| 213969 | Erfani | Sébastien |  |
| 214666 | Biechy | Robin |  |
| 214411 | Baumeyer | Kévin |  |
| 213841 | Jafaar | Lisa |  |
| 213090 | Jolain | Yohann |  |

# Project report

*Note: Fill this part with member presentations and task repartition.*

*Language: English/French, at your option.*



**

# solution manual

Pour utiliser notre jeu, utilisez le fichier .nes et lancez le sur l’émulateur NES de votre choix (nous recommendons l’utilisation de Fceux).

Appuyez sur la touche qui correspond à la touche A de la manette de la NES pour lancer la balle sur les briques.

Appuyez sur les flèches directionnelles gauche et droite pour manipuler la barre en bas et la faire aller à gauche, ou à droite.

Le but du jeu est de casser toutes les briques affichées, sans faire tomber la balle en bas de l’écran.

# technical documentation

Gestion des entrées utilisateurs:

La gestion de entrées utilisateurs dans notre programme est faite de la manière suivante: Nous regardons à un endroit spécifique de la RAM, l’adresse $4016. Nous prenons les valeurs stockées, et on applique un masque pour savoir quelles touches sont appuyées, vu que chaque touche change la valeur d’un bit.

Détection des collisions:

Pour la collision avec les briques, ce qui a été fait est qu’on a pris la position des coins de la balle (dont on a approximé la hitbox par un carré), et on a regardé si ces positions sont dans une brique (on regarde si une brique est présente à cette position), si oui, on enlève la brique, on regarde si on a tapé sur le coté, le bas, ou le haut et on rebondit en conséquence. On replace avant la balle au bord de la brique pour éviter les problèmes de collisions et rebonds faux.

Pour les rebonds, selon la manière dont on arrive sur la brique, on inverse la vitesse verticale (si on arrive sur le bas ou le haut de la brique) ou horizontale (si on arrive sur le coté).

Pour savoir si une brique est là, on regarde la ligne de briques qui correspond (en faisant une division par 8, les briques ayant 8 px de hauteur) et on regarde sur cette ligne si une brique est présente sur l’axe x qui correspond.

La collision sur les bords et haut de l’écran sont de simples conditions sur la position de la balle, si la balle est trop haute par rapport à l’écran, trop à gauche ou trop à droite, on remet la balle sur le bord en question, et on inverse sa vitesse en x (si on est sur les bords) ou en y (si on est en haut).

Cross compilation:

Pour compiler notre projet (écrit en C), pour un processeur 6502 NES, on a utilisé un compilateur nommé cc65 qui permet de convertir un code C en assembleur 6502. cc65 permet ensuite d’assembler cet assembleur et d’éditer les liens (avec les outils (ca65 et ld65). Nous avons simplifier ce processus en créant un script batch pour Windows et un script bash pour les systèmes Unix.