



# W3 - Rattrapages

---

W-RAT-010

## Isola

---

Rattrapages rush 2

1.0



# Isola

repository name: Rattrapages\_Rush\_02  
repository rights: ramassage-tek  
language: Node, JS, PHP



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

## ISOLA

### + DÉTAILS ADMINISTRATIFS

- Le projet est à réaliser seul.
- Les sources doivent être rendues avec BLIH.
- Répertoire de rendu : **Rattrapages\_Rush\_02**

### + INTRODUCTION

Lors des rushs de rattrapages, vous devez travailler seul sur un projet. Du Vendredi soir au Dimanche soir. L'objectif de l'exercice est de créer un jeu Isola à deux joueurs : [http://www.wikiwand.com/fr/Isola\\_\(jeu\)](http://www.wikiwand.com/fr/Isola_(jeu))  
Pour cela, vous aurez besoin, d'utiliser une représentation en 2D du plateau de jeu et des pions des joueurs.

### + RESTRICTIONS

- Les dates du BLIH font foi pour l'heure de rendu de votre projet.



Vous pouvez utiliser Phaser ou toutes autres librairies que vous jugerez plus performante pour l'affichage.

## + ETAPE 1

L'objectif est de:

- Modéliser le plateau de jeu et les pions des joueurs.
- Mettre en place les règles de jeu.
- Afficher un écran de défaite.

### LA MODÉLISATION

Le plateau de jeu devrait être symbolisé par des 7\*7 carrés de même taille. Les pions des joueurs par deux cercles de couleurs différentes du même diamètre que les carrés.

### LES RÈGLES

Au début du jeu, les pions sont placés au milieu de deux côtés opposés.

Les joueurs jouent les uns après les autres.

A chaque tour un joueur doit :

- Déplacer son pion vers une case libre adjacente ou touchant sa case de départ par un coin.
- Détruire une case du jeu non occupée.
- Le premier joueur qui ne peut pas déplacer son pion perd la partie.



Checkpoint : Attention, il est inutile de poursuivre si vous n'êtes pas arrivé correctement à ce stade. En cas de doute, vérifiez à nouveau votre travail.

## + ETAPE 2

L'objectif est de:

- Mettre en place une base de données.
- Permettre à deux joueurs de jouer sur deux PC distants.

### NODE.JS

Le serveur doit posséder une représentation de l'état du plateau de jeu à chaque instant. Chaque joueur doit pouvoir récupérer l'état du plateau de jeu à chaque début de tour, puis modifier l'état courant du jeu dans le serveur à la fin de son tour.



Utilisez les sockets pour tous les échanges de données entre client et serveur.

### + BONUS 1

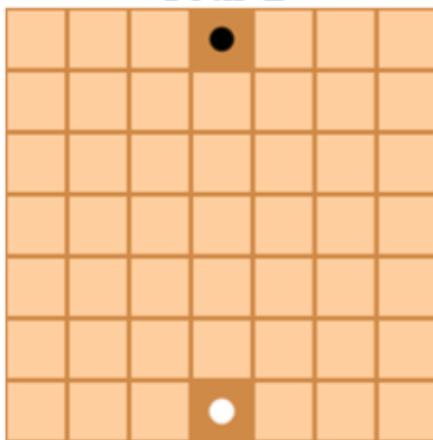
Implémentez une solution pour 4 joueurs.

### + BONUS 2

Utilisez SimpleWebRTC pour une implémentation WebRTC de votre jeu.

### + EXEMPLE

Tour 1



Tour 3

