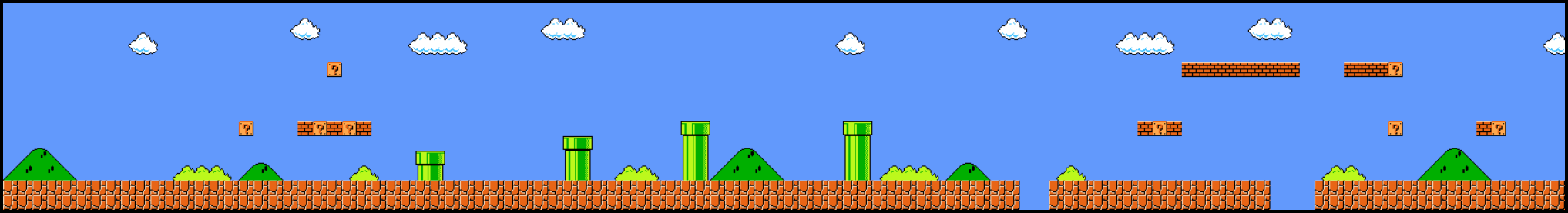
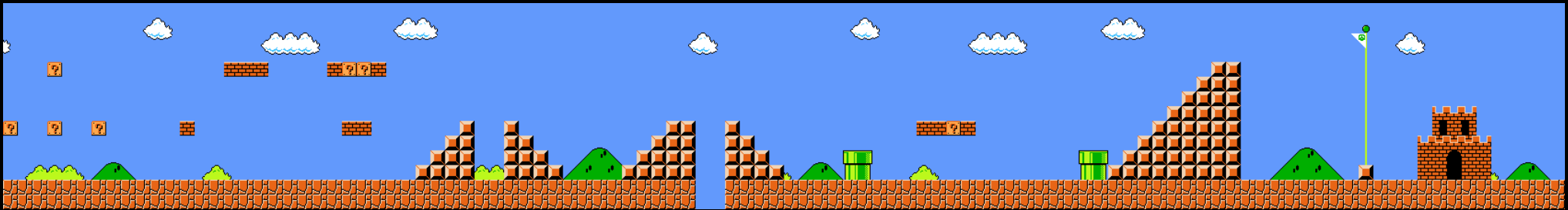


Aperçu des instructions



But ultime
Reproduire le niveau 1-1 de
Super Mario Bros.



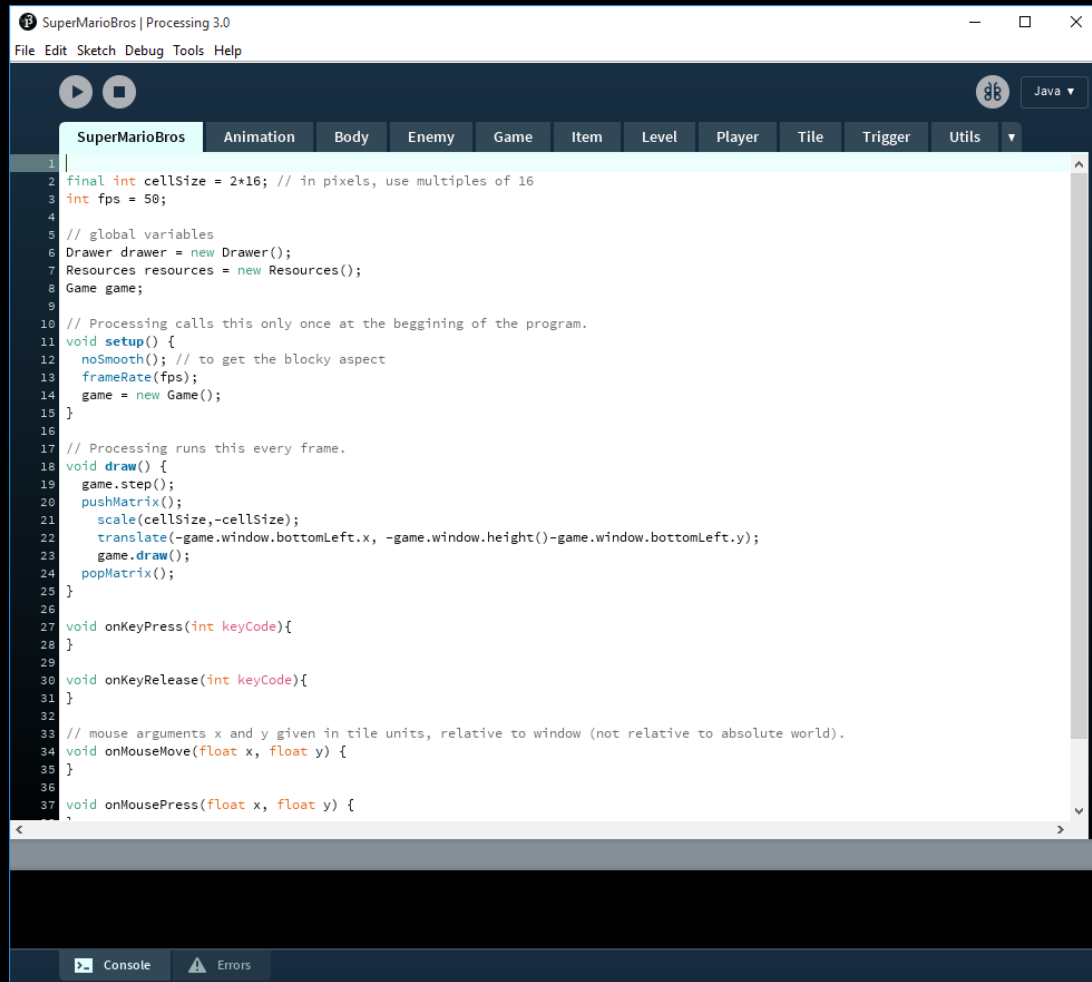
Processing 3.0

Basé sur Java

Interface simple
et épurée

Déjà installé sur
vos ordinateurs

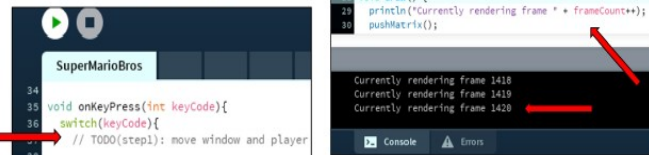
Base de code à
compléter



```
1 |
2 final int cellSize = 2*16; // in pixels, use multiples of 16
3 int fps = 50;
4
5 // global variables
6 Drawer drawer = new Drawer();
7 Resources resources = new Resources();
8 Game game;
9
10 // Processing calls this only once at the beginning of the program.
11 void setup() {
12   noSmooth(); // to get the blocky aspect
13   frameRate(fps);
14   game = new Game();
15 }
16
17 // Processing runs this every frame.
18 void draw() {
19   game.step();
20   pushMatrix();
21   scale(cellSize,-cellSize);
22   translate(-game.window.bottomLeft.x, -game.window.height()-game.window.bottomLeft.y);
23   game.draw();
24   popMatrix();
25 }
26
27 void keyPressed(int keyCode){
28 }
29
30 void onKeyRelease(int keyCode){
31 }
32
33 // mouse arguments x and y given in tile units, relative to window (not relative to absolute world).
34 void onMouseMove(float x, float y) {
35 }
36
37 void onMousePress(float x, float y) {
```

Objectif 1: Créer un simple monde de tuiles

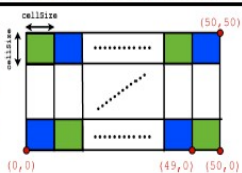
Familiarisez-vous au code en travaillant à quatre pour compléter le premier objectif : créer un damier et un "joueur" rectangulaire qui s'y promène.



Conseil 1: Recherchez les TODO dans le code. Conseil 2: Affichez à la console avec `println(...)`.

1. Vous allez construire le monde 2D de Mario à partir de *cellules*: 1x1 mètre carré, alignées sur une grille. Modifiez `Game.draw()` pour créer un tableau de 50x50 cellules en un damier bleu et vert. Assurez-vous de dessiner des tuiles sans bordure à l'aide de `strokeWeight(0)`.

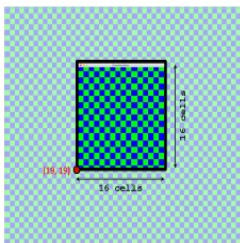
Une cellule est représentée par `cellSize x cellSize` pixels sur l'écran.



2. En tout temps, votre fenêtre de jeu n'affichera que 16x16 cellules sur l'entièreté du damier de 50x50. Éventuellement, quand Mario se déplacera dans le monde, la fenêtre devra se déplacer avec lui. Commencez par redimensionner et positionner la fenêtre à (19,19).

Ensuite, créez un joueur très simple dans notre monde: premièrement, placez votre joueur à (25,25) dans le monde dans `Game.init()`.

Pour l'instant, votre joueur sera un simple et ennuyeux rectangle 2x1: initialisez sa taille dans `Game.init()` et appelez `player.draw()` dans `Game.draw()`. Finalement, dessinez le joueur dans `Body.draw()`.



3. Il est temps de permettre au joueur de se déplacer avec les touches du clavier.

Modifiez `onKeyPress` dans `SuperMarioBros.pde` pour déplacer le joueur de 1/2 cellule dans les quatre directions, utilisant les touches W, A, S et D. Vous pouvez procéder en ajoutant un vecteur de déplacement à la position du joueur avec `game.player.pos.add(new Vec2(...))`.

Similairement, ajoutez du code pour bouger la fenêtre dans le monde en utilisant les touches fléchées (`keycodes UP, DOWN, LEFT et RIGHT`), encore une fois de 1/2 cellule. Ici, vous ajoutez un vecteur de déplacement en utilisant `game.window.translateBy(new Vec2(...))`.

Finalement, utilisez P pour activer/désactiver `game.play` et court-circuiter la fonction `Game.step()` lorsque `game.play == false`.



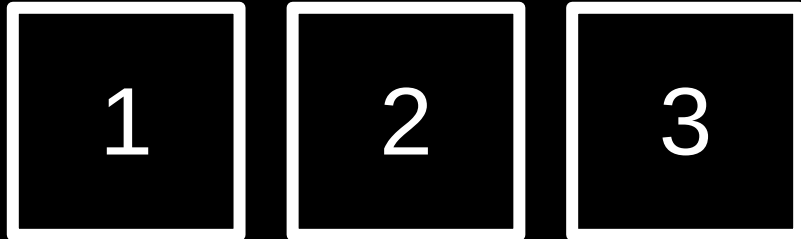
9 objectifs

Instructions claires et dirigées



Instructions vagues et libres

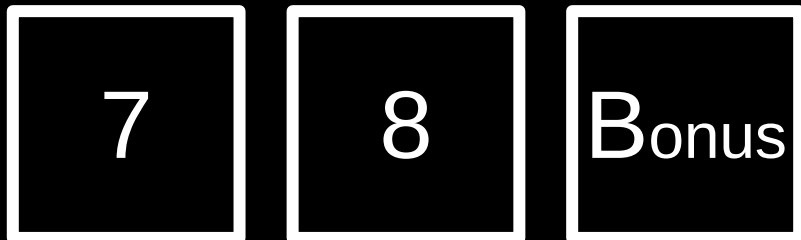
Vous ne pourrez pas tout faire



Tout faire

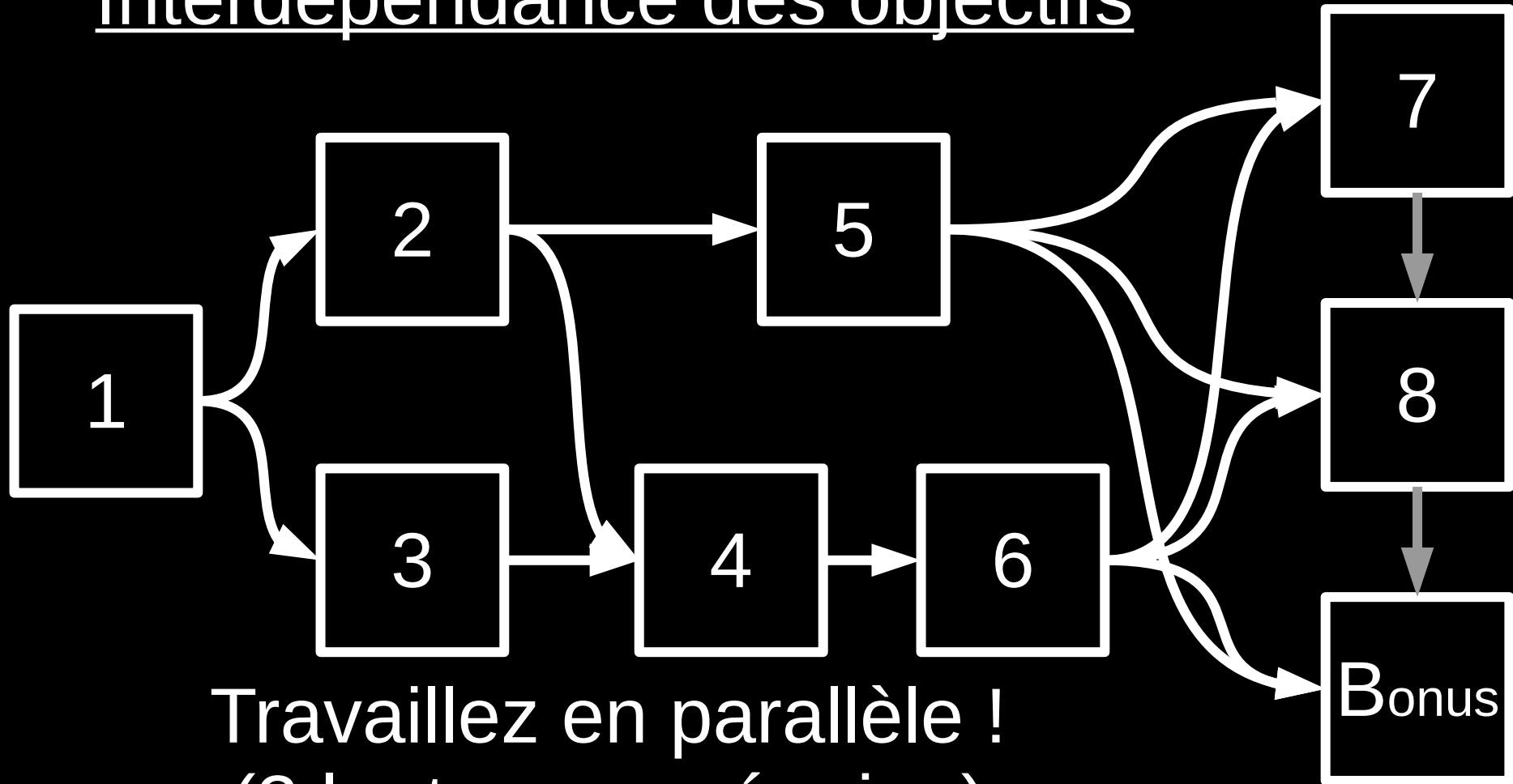


Faire la majorité



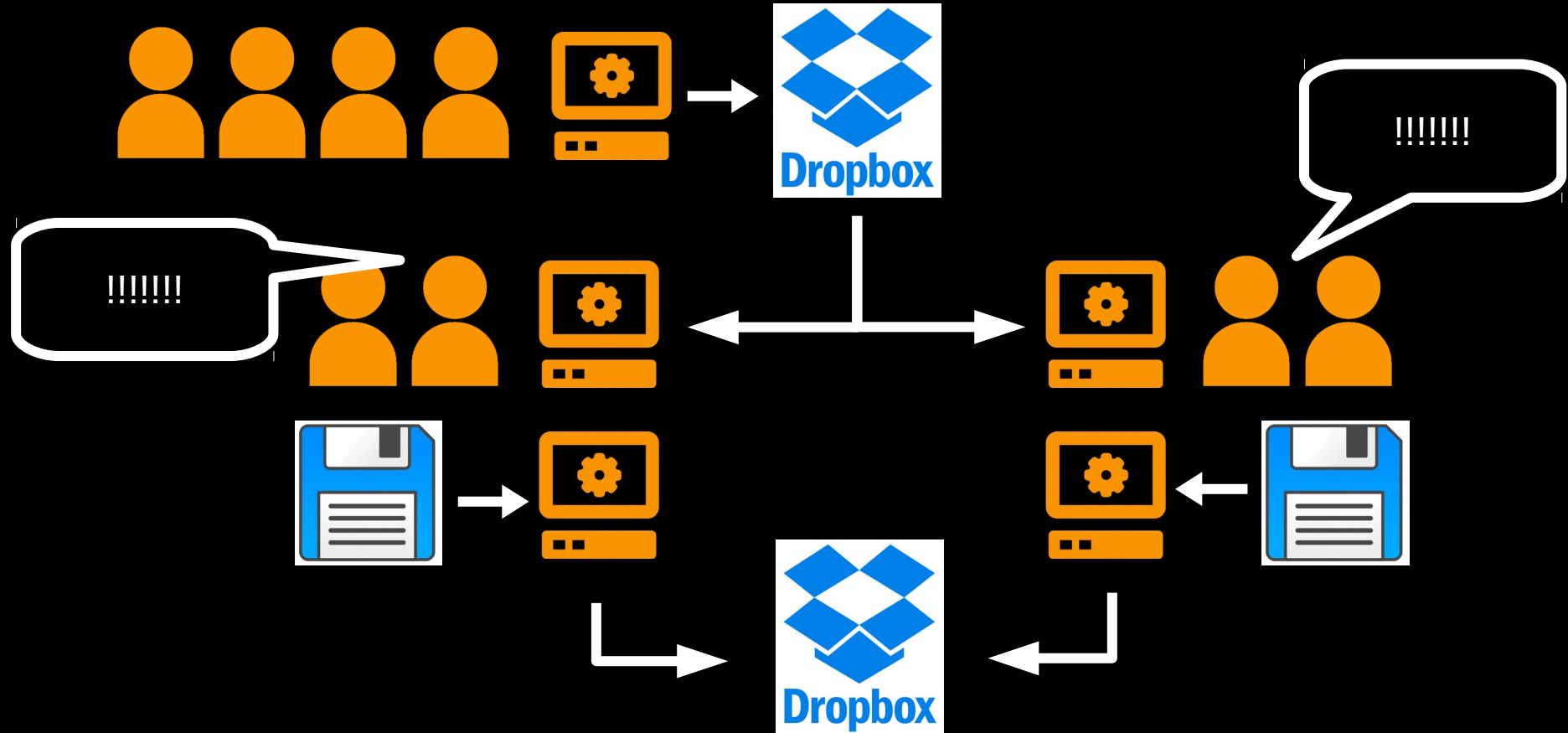
Choisir judicieusement

Interdépendance des objectifs

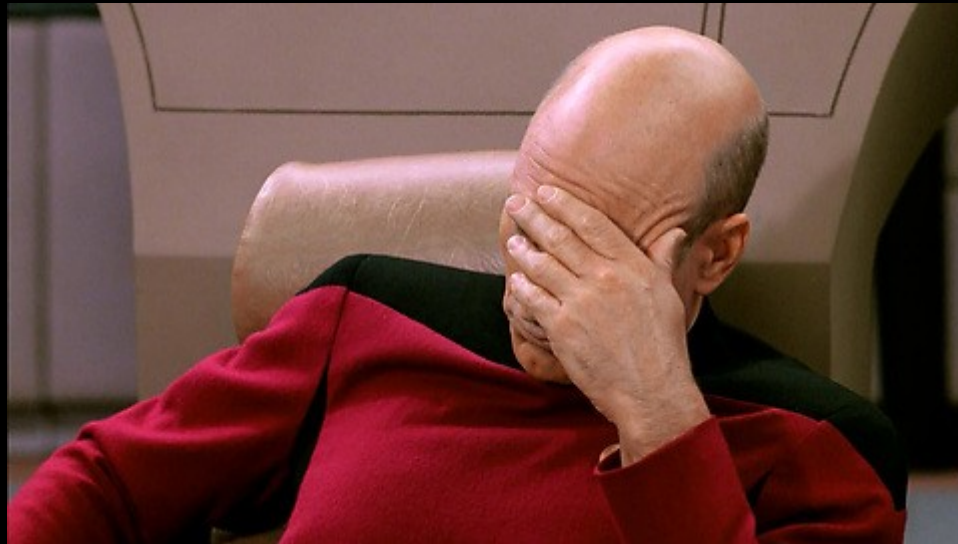


Travaillez en parallèle !
(2 laptops par équipe)

Séparer le travail



Jamais joué à
Super Mario Bros. !?!



Voir instructions pour
version jouable en ligne

Erreurs à ne pas faire

- Ne pas faire de sauvegardes
- Passer trop de temps sur un objectif
- Passer trop de temps sur des détails
- Vouloir en faire trop
- Ne pas faire de sauvegardes

Présentations

2 minutes par équipe

Code soumis dans un dossier spécial
avant une certaine heure limite



Bibliothèque de mathématique et informatique



Pas de nourriture



Boissons avec couvercle
fermé seulement



Garder un niveau de
bruit respectueux

Horaire (sujet à modifications)

Samedi

10h-12h : code

12h-13h : diner

13h-17h30 : code (+pause)

17h30-18h30 : souper

18h30-23h30 : code (+pause)

Dimanche

10h-10h30 : déjeuner

10h30-12h : code

12h-13h : diner

13h-14h45 : code

14h45-15h30 : présentations

~15 heures pour coder : pas le temps de niaiser

Oui, c'est une compétition !

Plaisir !

Expérience !

...

Prix substantiels pour 3 premières places

Évaluation :

- proximité au jeu original
- fonctionnalités bonus
- structure du code
- travail d'équipe

Instructions disponibles dans quelques minutes

hackathon.iro.umontreal.ca/instructions_fr.pdf

hackathon.iro.umontreal.ca/instructions_en.pdf

hackathon.iro.umontreal.ca/introSlides.pdf

hackathon.iro.umontreal.ca/horaire.pdf



DIRO

présente