

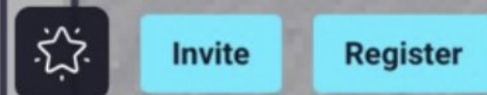
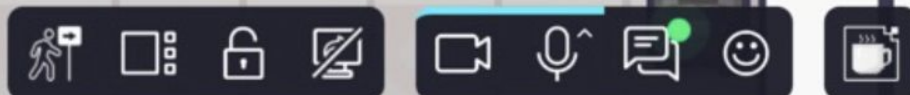
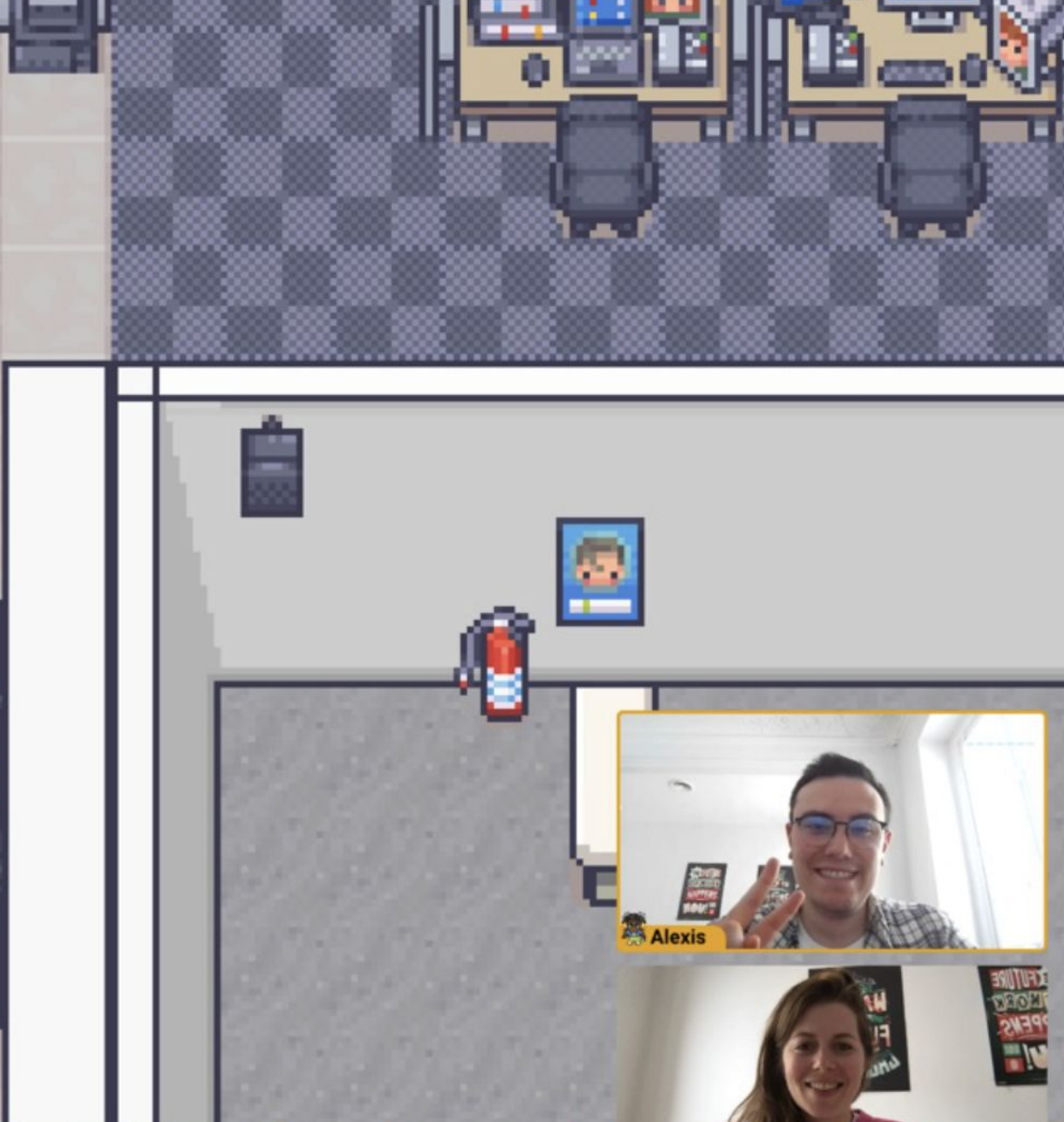
# Créer un escape game avec WorkAdventure et Typescript



# Workadventure

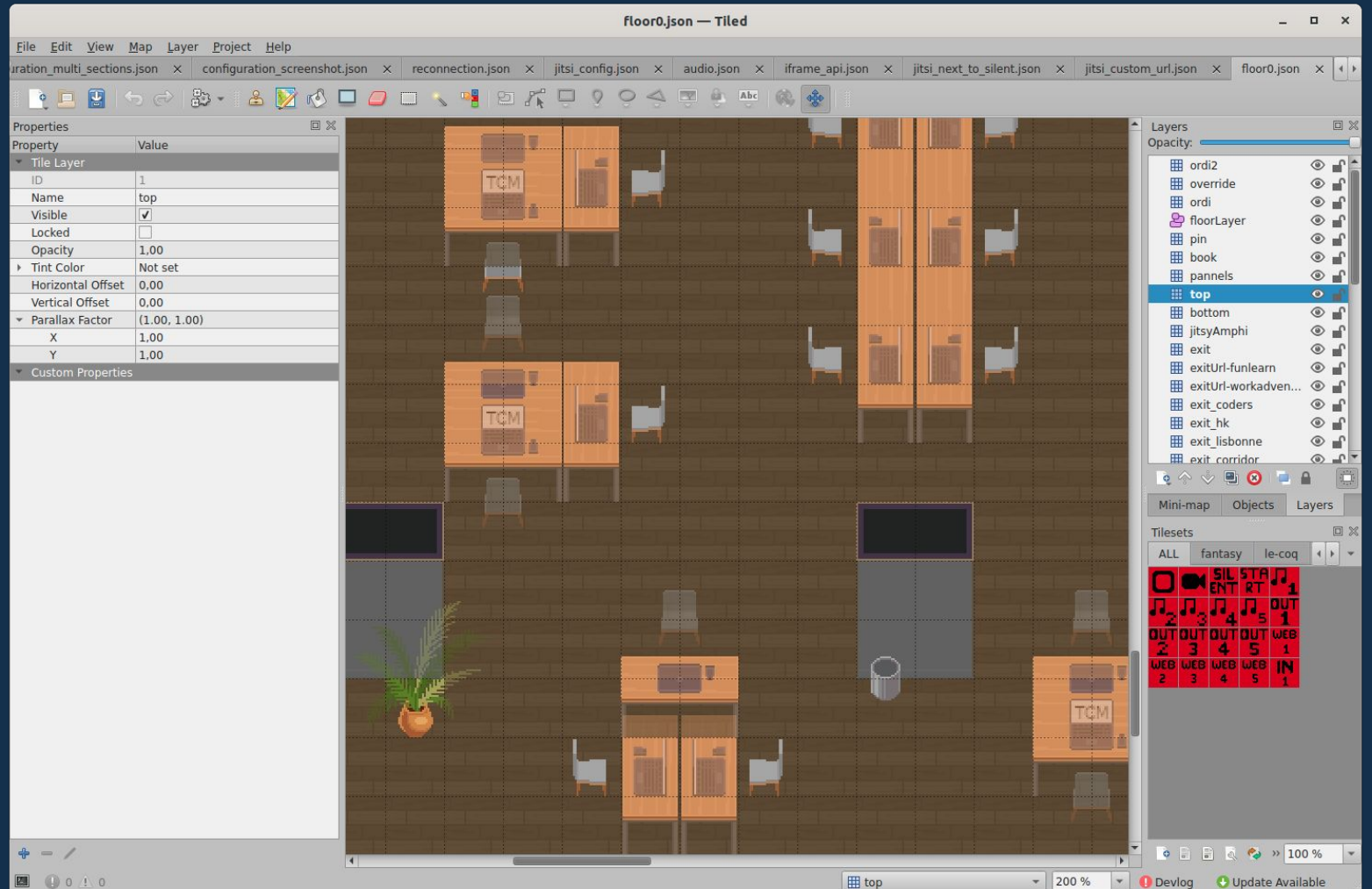
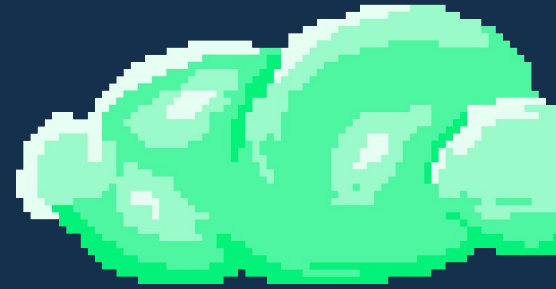
?

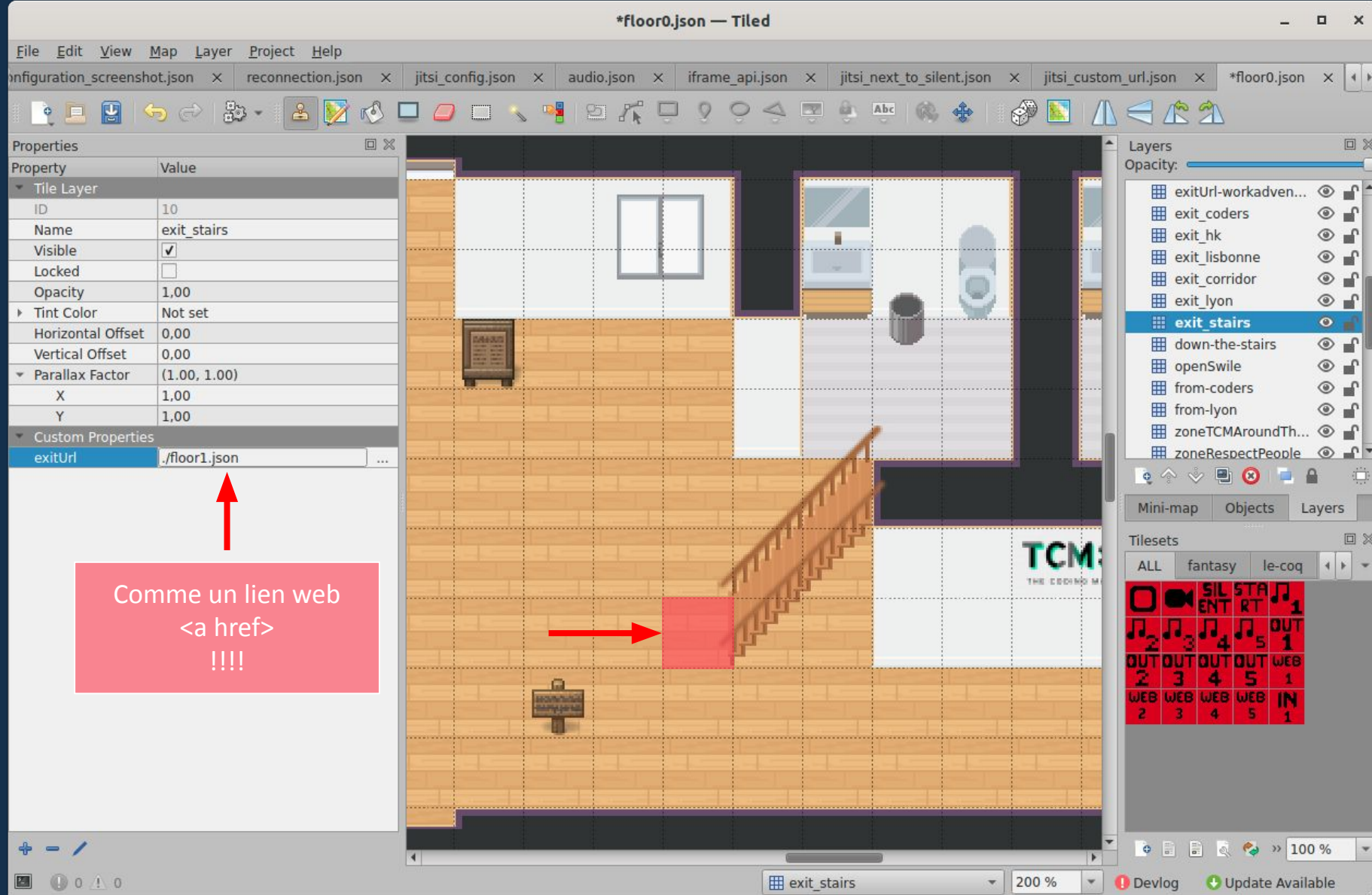






# L'éditeur de cartes

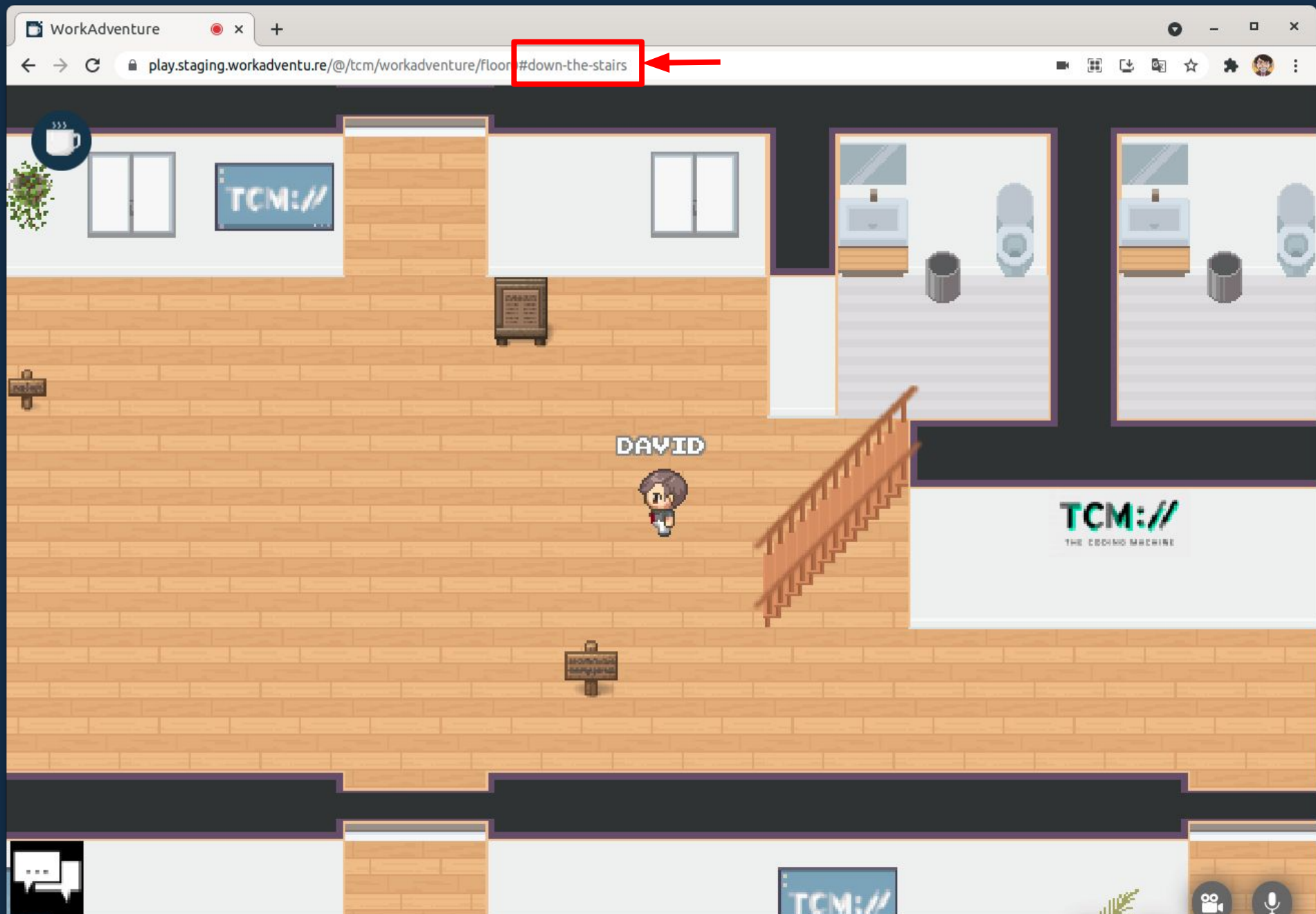




Comme un lien web  
<a href>  
!!!!

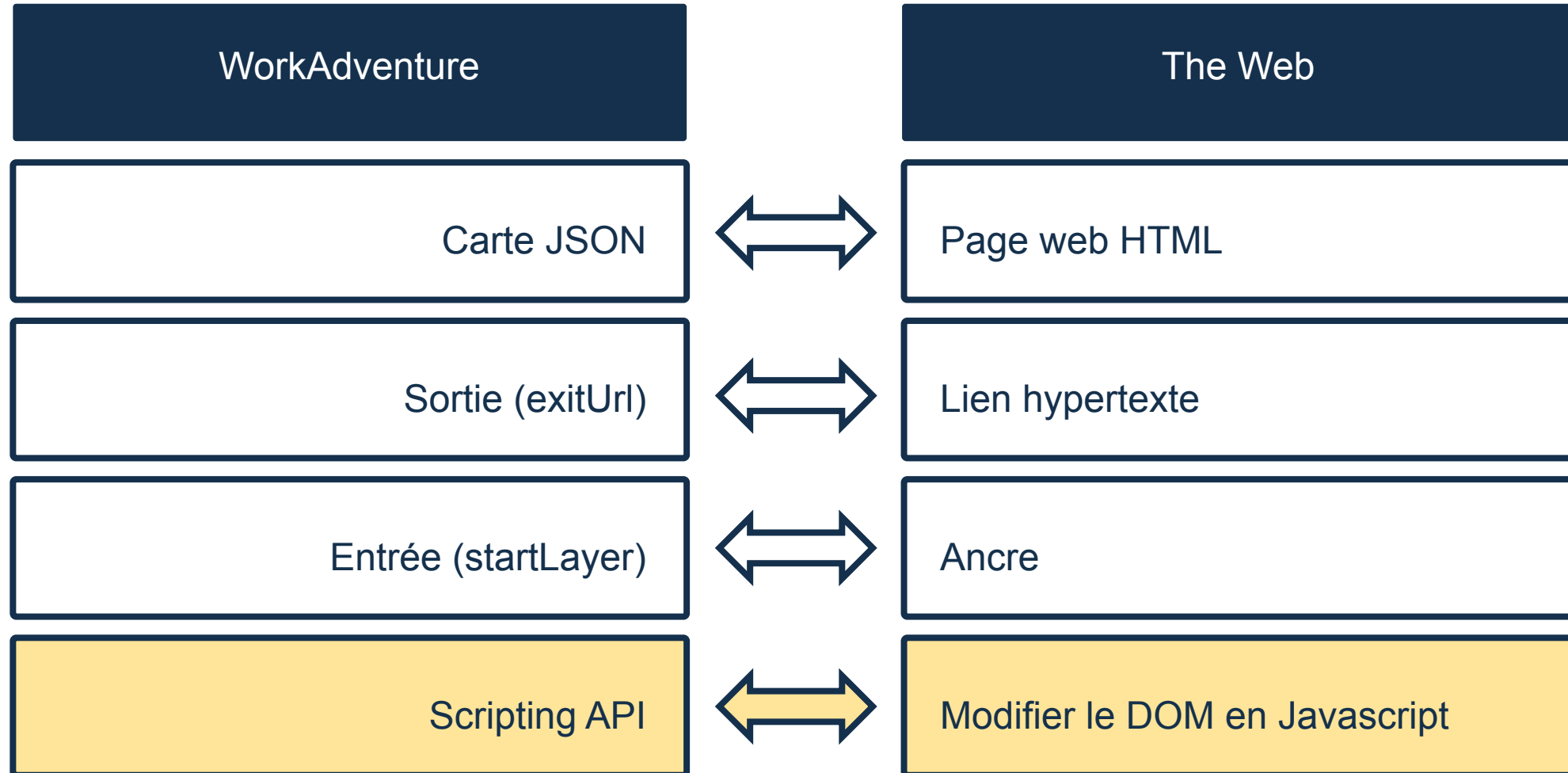
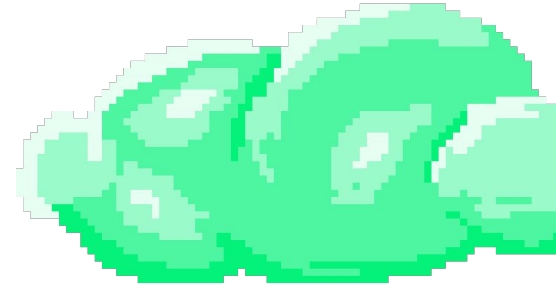






# WorkAdventure

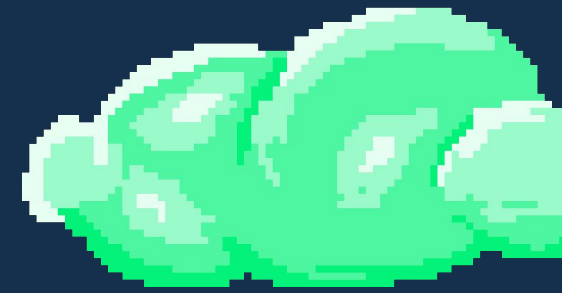
# VS the Web



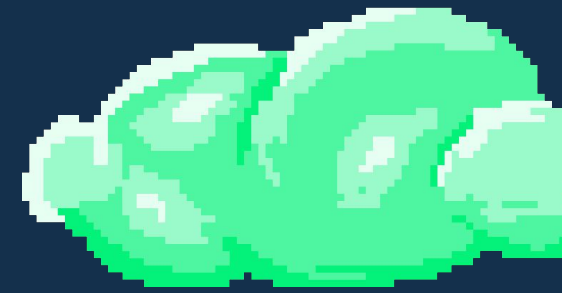


# Scripting maps



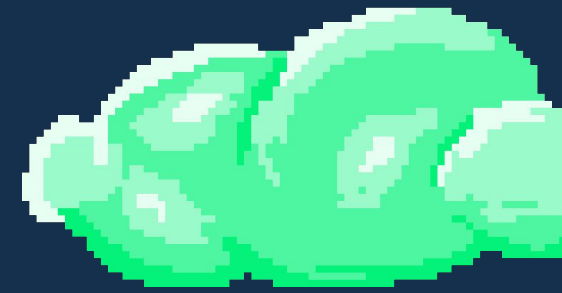


Dans WorkAdventure, la “scripting API” est  
exécutée *côté client*



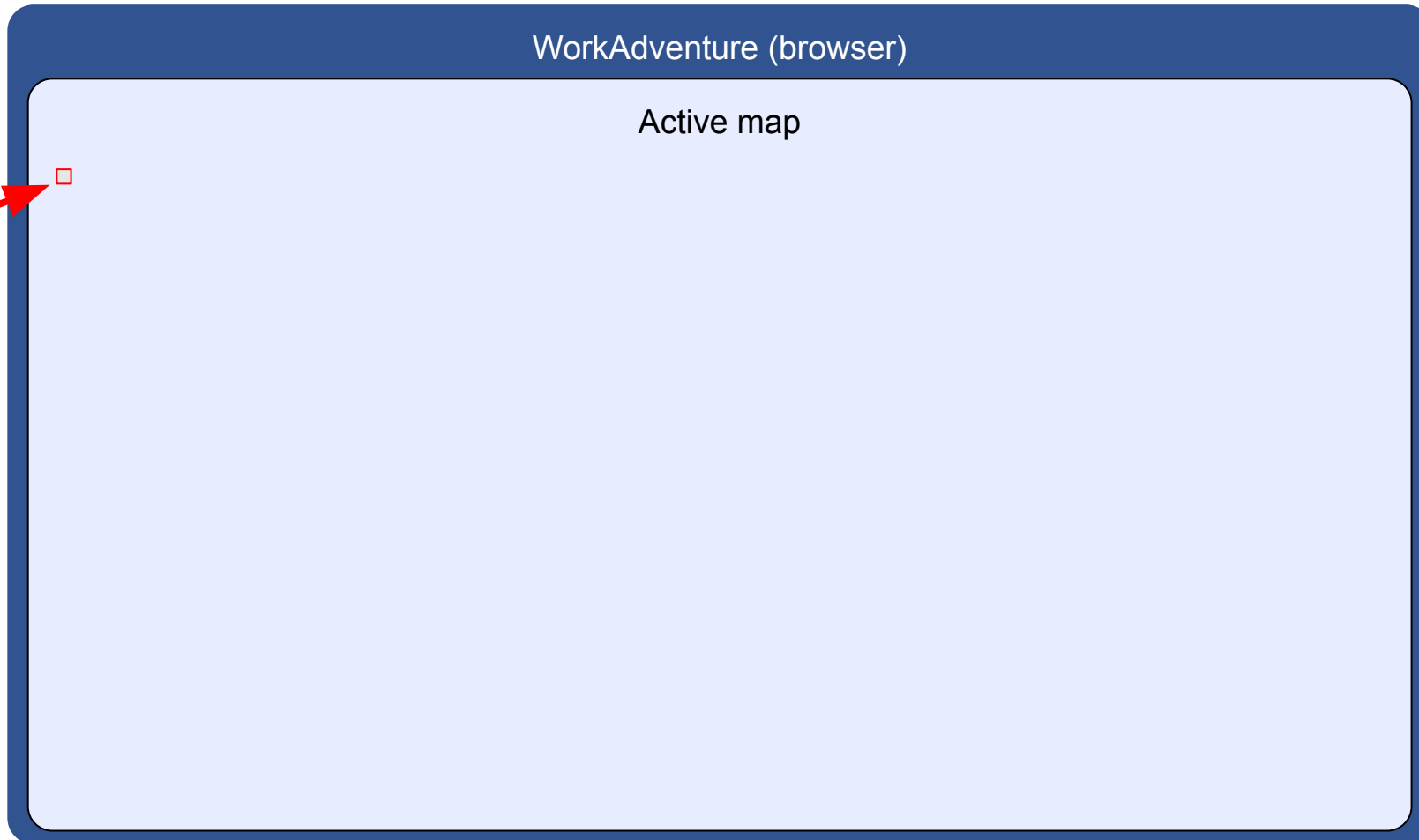
Besoin d'éviter les injections XSS

Besoin d'**isoler** le script de la plateforme



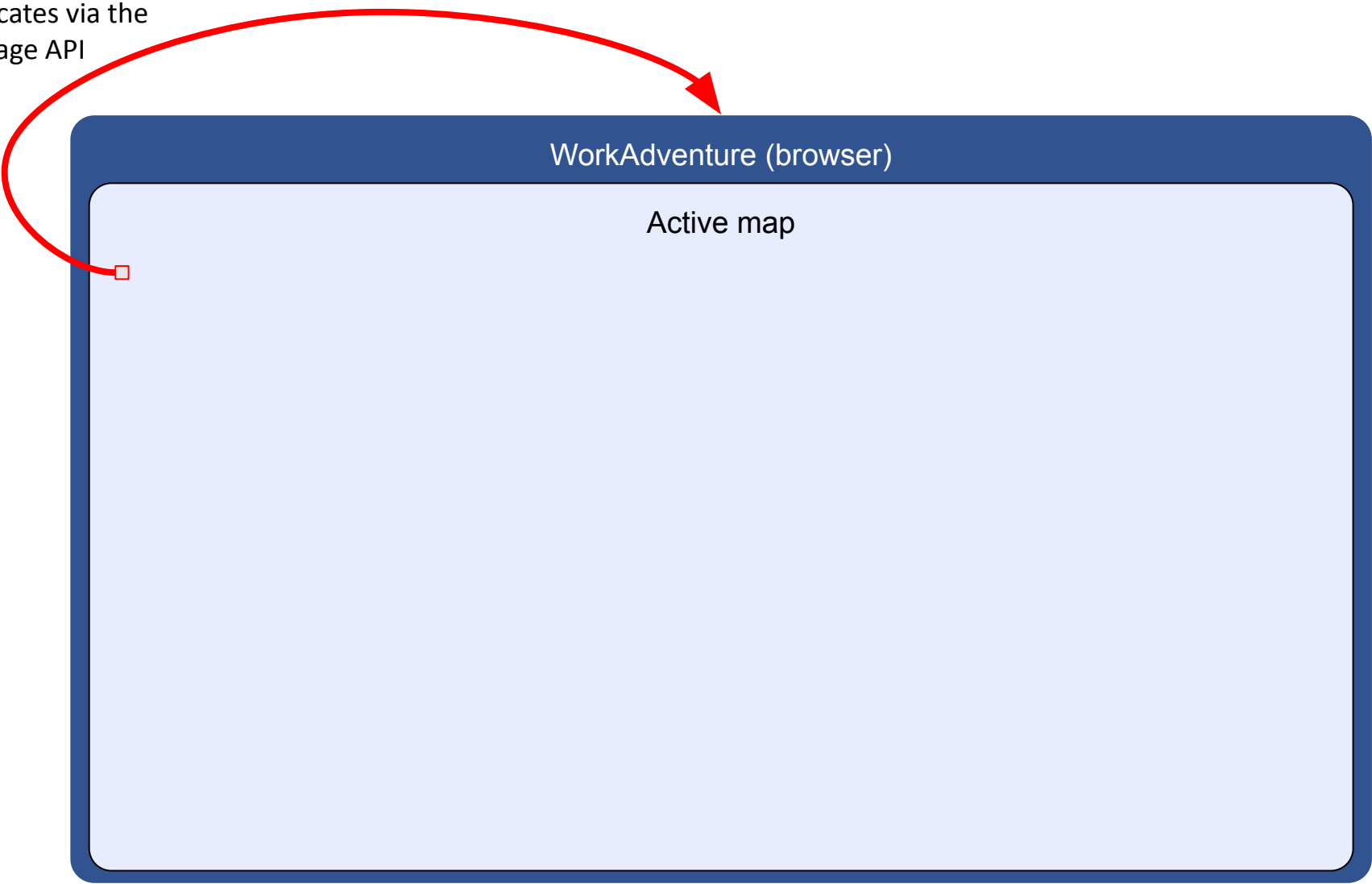
**iframes** to the rescue!



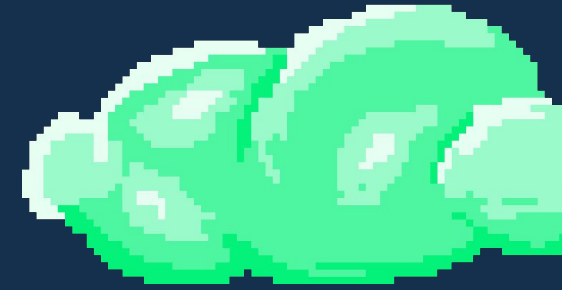


- 1px \* 1px iframe
- contains the code of your map

Communicates via the  
postMessage API

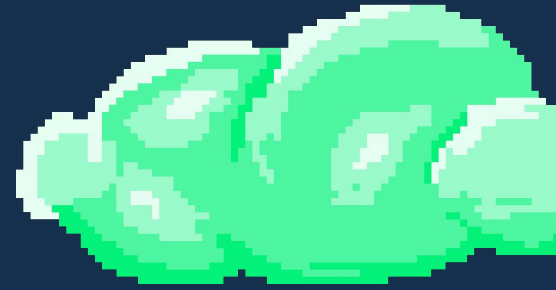


# Scripting API



La complexité des appels cross-iframe avec “`window.postMessage`” est cachée derrière une librairie qui fournit une API simple.

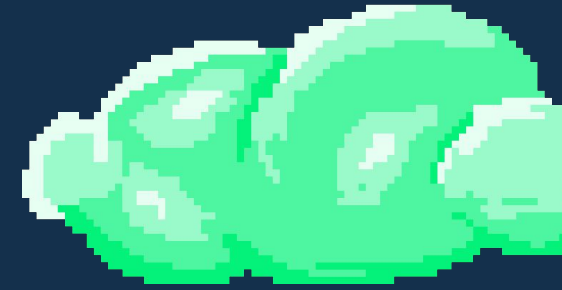
# Scripting API



L'API fonctionne dans le navigateur en  
Javascript



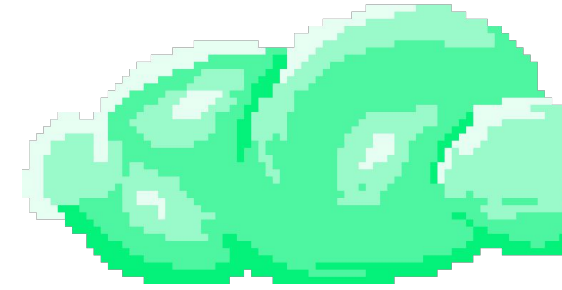
# Scripting API



Donc on peut scripter des cartes en

# Typescript

# Scripting API



## Sample script

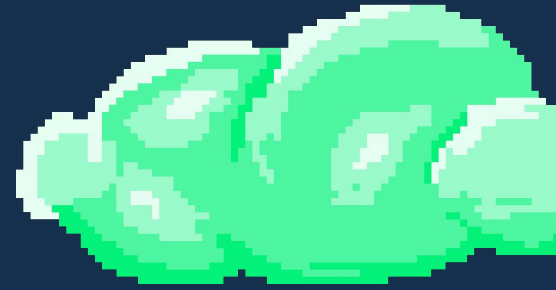
```
WA.room.onEnterLayer ('clock', () => {  
  const today = new Date();  
  const time = today.getHours () + ":" + today.getMinutes ();  
  WA.ui.openPopup ("clockPopup", "It's " + time, []);  
})
```



# Live coding



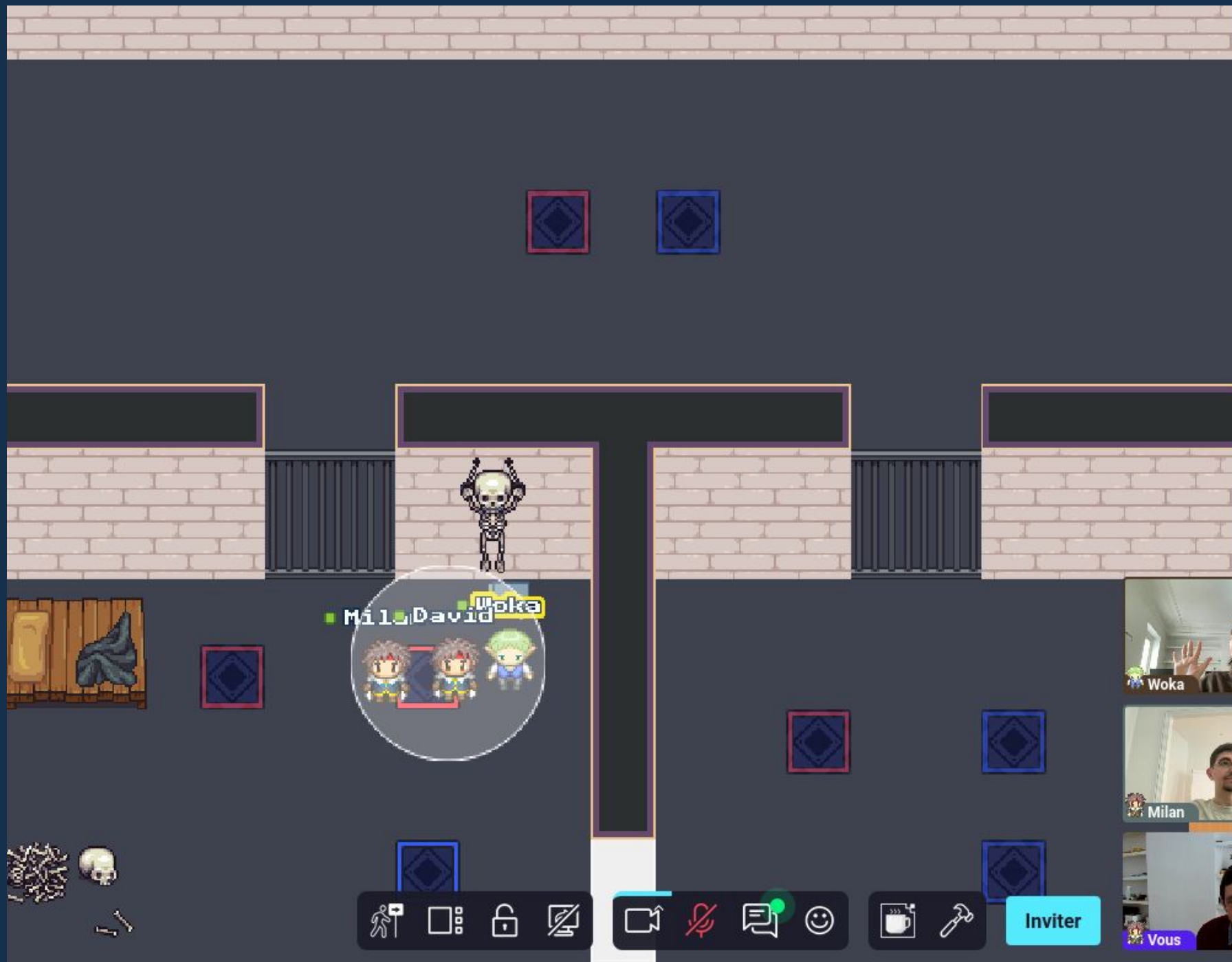
# Notre escape game

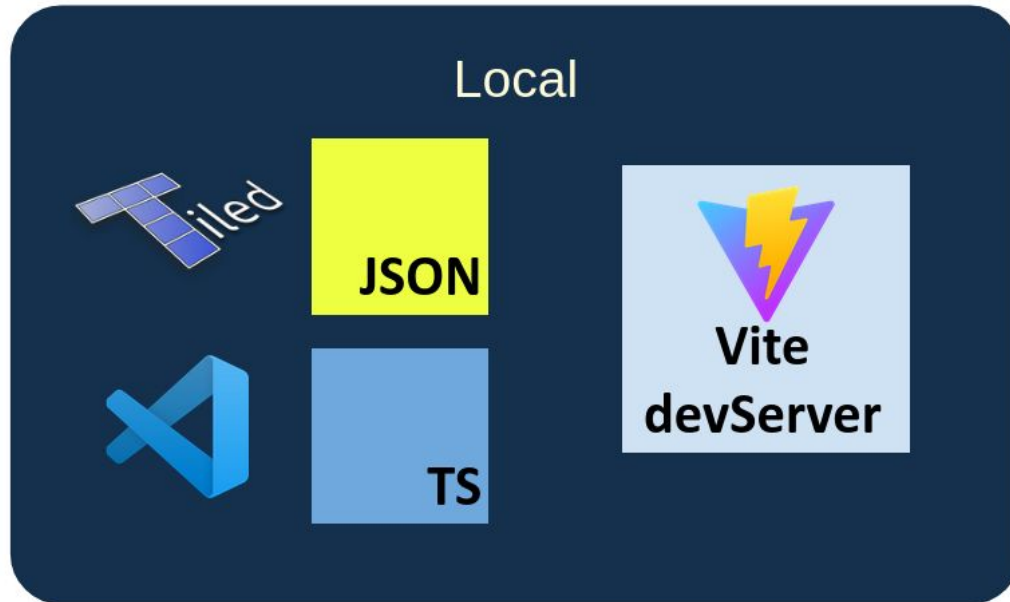


Des tuiles de couleur (rouge / bleu)

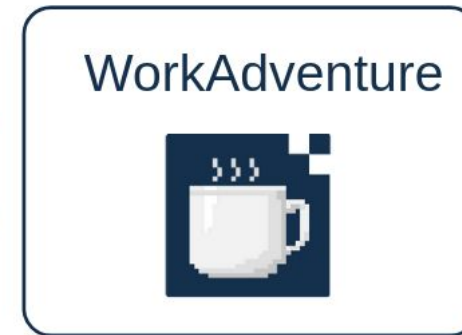
Il faut marcher sur 2 tuiles de la même couleur pour ouvrir une porte







Vite compile carte et scripts et expose  
un server web de test

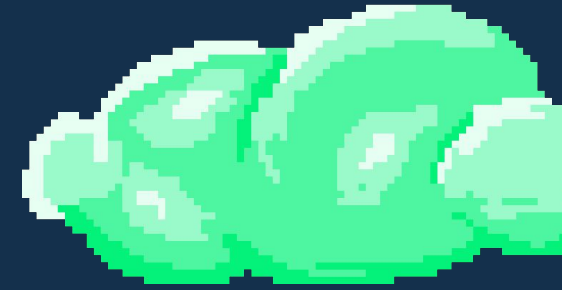


WorkAdventure charge  
la carte en local

Go  
!



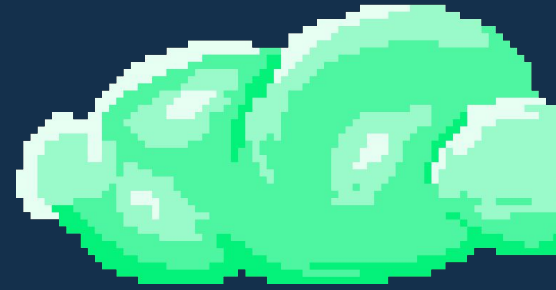
# Scripting API



- Les changements sont effectués dans le navigateur, en **local**
- Les changements **ne sont pas partagés** avec d'autres utilisateurs

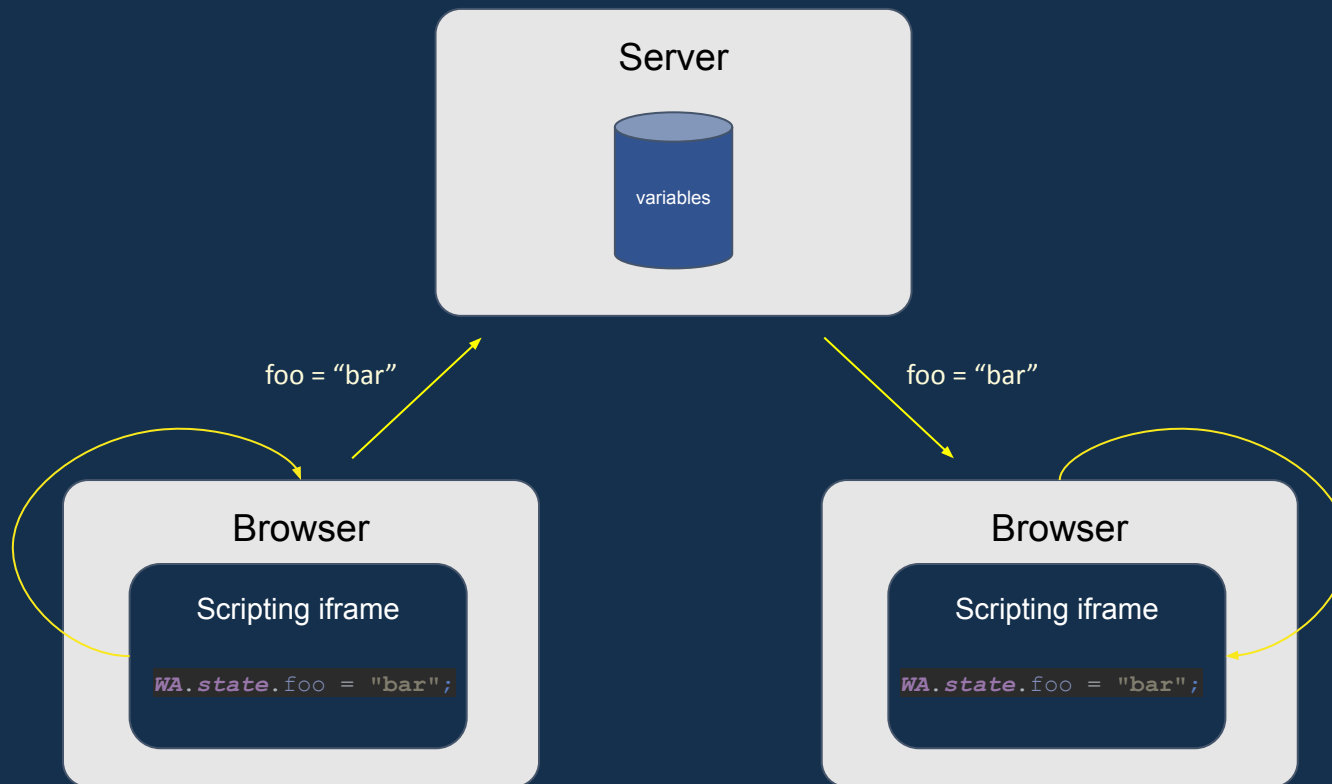
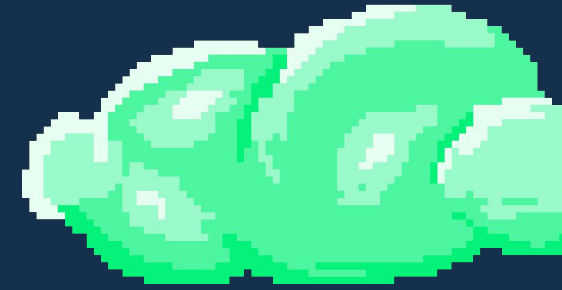


# Scripting API



Besoin de partager un “état” avec d’autres utilisateurs

# Scripting - Variables

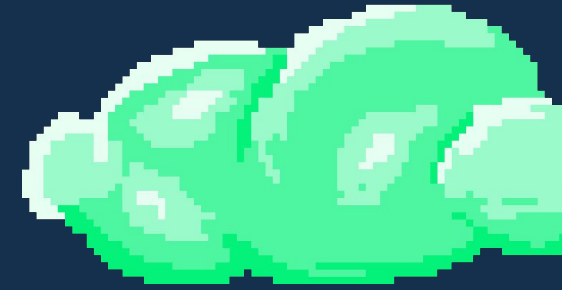


## Bonus:

You can listen to variables changes.

```
WA.state.onVariableChange (
  'foo',
  (value) => {
    console.log('Hooray!')
  }
)
```

# Scripting - Variables

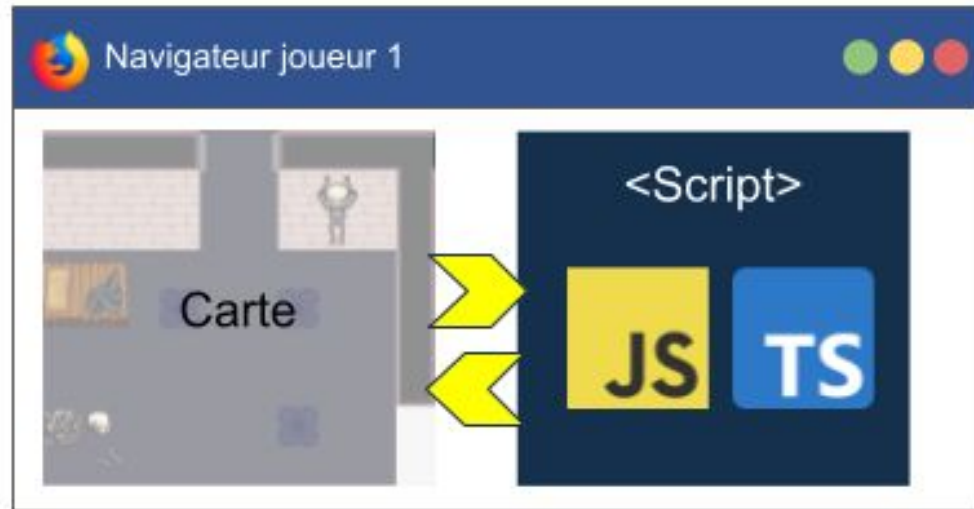


2 types de variables

Room variables

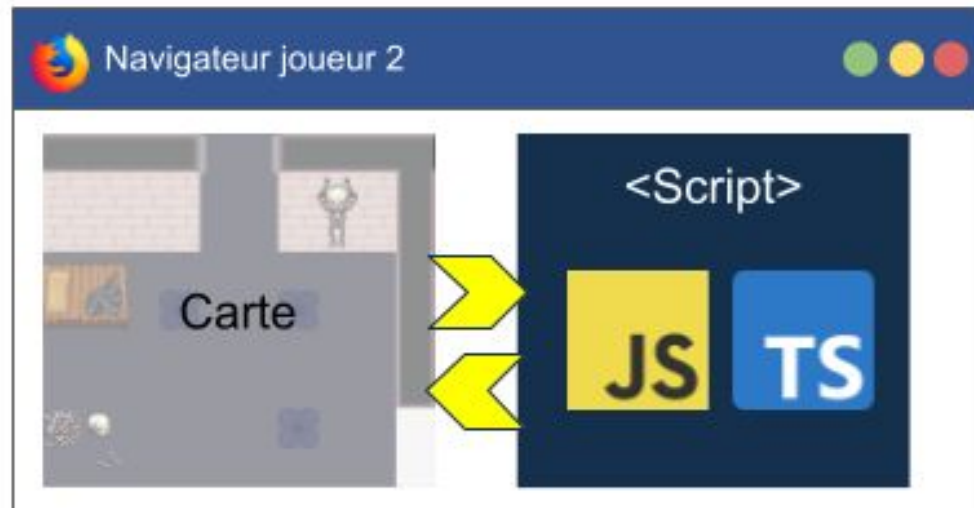
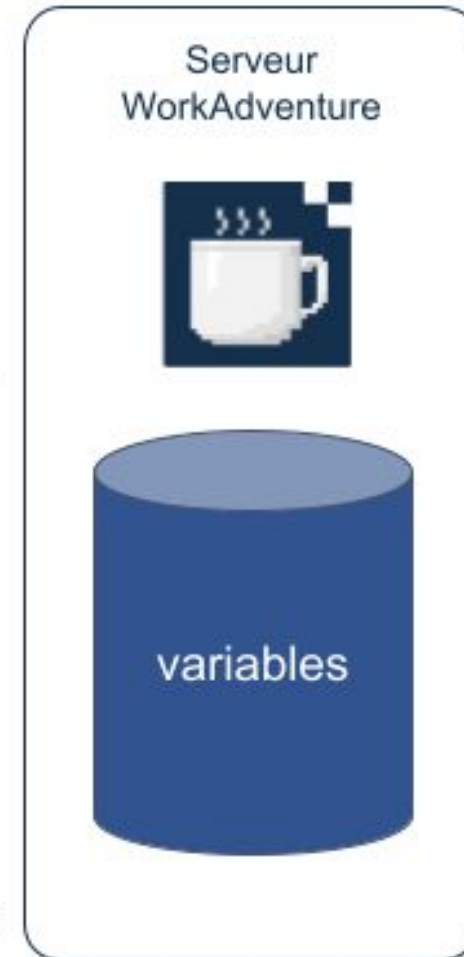
Player variables

- 1 Le joueur 1 marche sur une dalle rouge.  
La variable du joueur est modifiée



- 2 Le serveur reçoit, stocke et propage les variables.

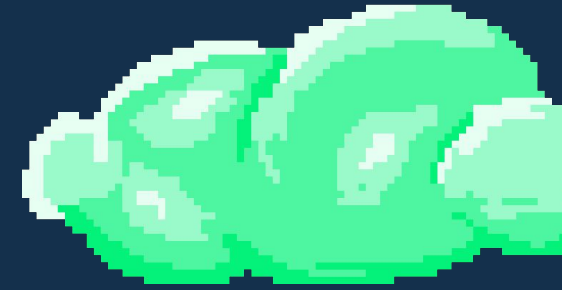
Variable modifiée  
`buttonPressed=red`



`player1.state.  
buttonPressed=red`

- 3 La carte est mise à jour à partir des nouvelles valeurs des variables.

# Et demain?



Garder WorkAdventure ouvert pour en faire le  
bac à sable ultime...

...

Encourager la communauté à créer un  
écosystème de librairies

...

... et laisser la communauté nous surprendre!

“me”

David Négrier  
david@workadventu.re



# WorkAdventure

Your workplace. Better.



@moufmouf



@david\_negrier



joinind.in/user/moufmouf

Slides:

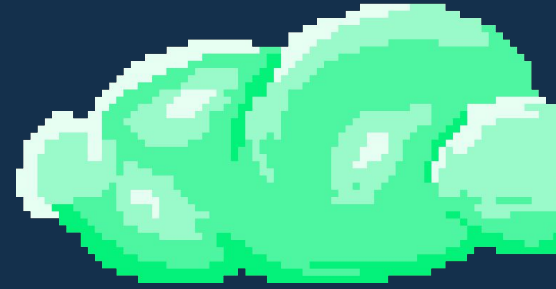
<https://bit.ly/3Xpilcd>



<https://workadventu.re/demo>



# Resources



Documentation:

<https://workadventu.re/map-building/scripting.md>

Map:

<https://github.com/moufmouf/programmez-escape-game>





# WorkAdventure

Your workplace. Better.