

**DOSSIER-PROJET ISN 2015**

# **STARWALKER**

**RÉDIGÉ PAR HEXABEAST**

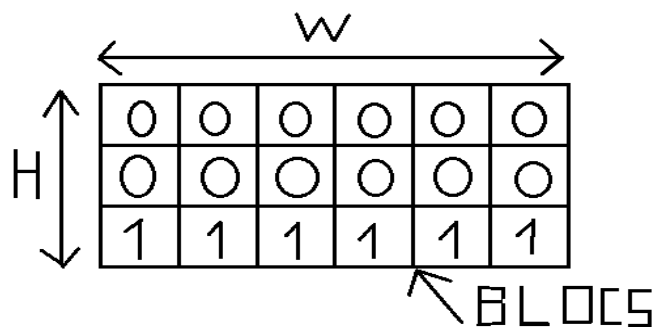
## V – Explications techniques :

Le code du jeu est réparti dans 13 modules différents, et n'est constitué d'aucune forme de programmation orientée objet, malgré le fait que le Python soit un langage de programmation orienté objet, car cela n'est pas au programme et son utilisation a été fortement déconseillé par le professeur d'ISN.

Voici une brève présentation de chacun des modules :

Le module « afficheur » permet de gérer l'affichage des textures et des textes à l'écran par rapport aux coordonnées qui lui sont donnés par le module « camera », qui se recentre constamment en direction du joueur tout en faisant attention de ne pas dépasser les limites du niveau. Ces deux modules réunis permettent donc de simuler le déplacement dans un monde dépassant les limites de l'écran, ce que l'on appelle communément le « scrolling ».

Le module « carte » est relativement complexe. Il dispose de trois variables principales : Une liste de listes d'entiers nommée « blocs », grille des blocs qui constituent le niveau, et deux entiers, la hauteur h et la largeur w de cette grille :

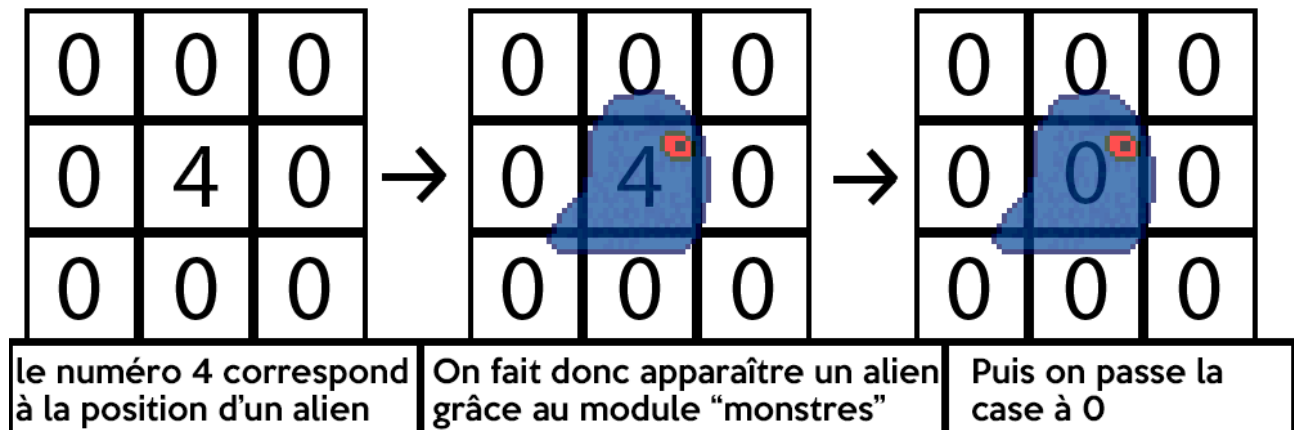


Dans cet exemple,  $w = 6$ ,  $h = 3$ ,  
et blocs =  
[[0,0,0,0,0,0],[0,0,0,0,0,0],[1,1,1,1,1,1]]

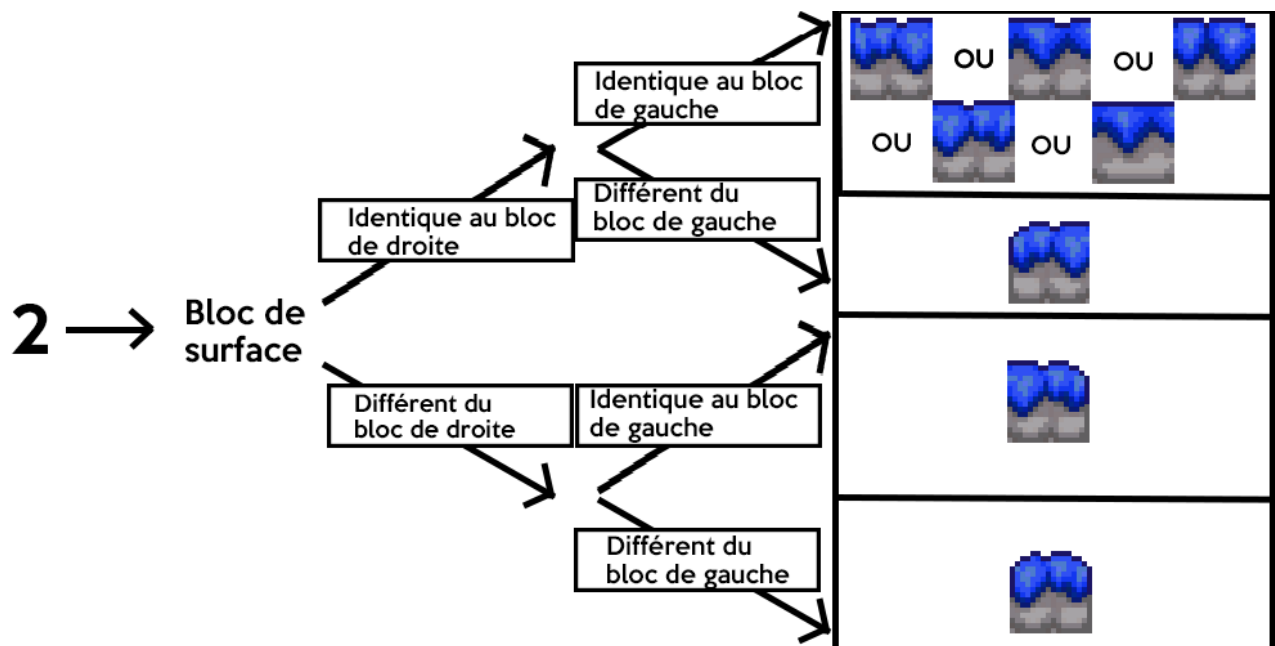
En réalité, les variables w et h ne sont pas nécessaires car elles pourraient être obtenues en utilisant la fonction len() sur la grille, mais elles permettent une meilleure lisibilité du code.

Chaque entier de la grille correspond à un type de bloc : 0 pour le vide, 34 pour un certain type de terre, les nombres de 25 à 29 pour les différentes formes de roches...

La fonction « Charger » de ce module lit le fichier qui lui est donné en argument, constitué de lignes de nombres séparés par des virgules, et place chacun de ces nombres dans la grille à la bonne place, en effectuant un traitement préalable si besoin, par exemple en faisant apparaître un monstre grâce au module « monstres », ou un bonus à l'aide du module « bonus », et en repassant la case correspondante au nombre 0, correspondant au vide :



Le traitement permet aussi d'ajouter le facteur aléatoire au terrain en choisissant aléatoirement une variation du bloc demandé ou de lui assigner une variation spécifique selon s'il est entouré de blocs du même type ou non :



La fonction « draw » du même module permet de déterminer et d'afficher les blocs visibles à l'écran selon leur position dans la grille et leur nature. La fonction « collision », quand à elle, renvoie si le bloc aux coordonnées données en argument est solide ou non.

Le module « bonus » permet l'apparition de bonus, qui flottent dans l'air avec un mouvement vertical, et qui disparaissent au contact du héros, émettant un son et donnant au joueur la possibilité de les utiliser.

Le module « joueur » gère les déplacements, le comportement et l'affichage du héros, en fonction des entrées clavier et souris détectées à l'aide du module « inputs ». Pour appliquer un comportement physique réaliste au joueur, les lois de Newton sont utilisées, donnant au joueur une accélération constante vers le bas, qui permet un retour réaliste vers le sol suite à un saut.

Le module « monstres » fonctionne sur le même principe, mais gère une liste de monstres qui disposent d'une intelligence artificielle basique au lieu d'un seul personnage dont les déplacements sont déterminés par les entrées clavier. Les monstres tuent le joueur à son contact. Le module « boulesdefeu » est très similaire, à la différence près que l'intelligence artificielle se résume à rebondir sur les obstacles, toujours à l'aide des lois de Newton, que chaque boule de feu a une durée de vie limitée, et que les boules de feu ne tuent pas le joueur mais les monstres.

Le module « musique » permet le chargement des sons et de la musique, tandis-que le module « images » se charge des textures.

Le module « cache » sert, comme son nom l'indique, à « cacher » la transition entre les niveaux, en rendant l'écran noir avec un fondu et en affichant le nom du prochain niveau.

Le module « interface » affiche le nombre restant de chaque bonus en haut à gauche de l'écran et les textes d'aide dans le tutoriel.

Enfin, le module « main » permet la création de la fenêtre ainsi que l'initialisation et la coordination de tous les autres modules. Il contient la boucle de rendu du jeu et c'est lui qui doit être exécuté pour que le jeu fonctionne.

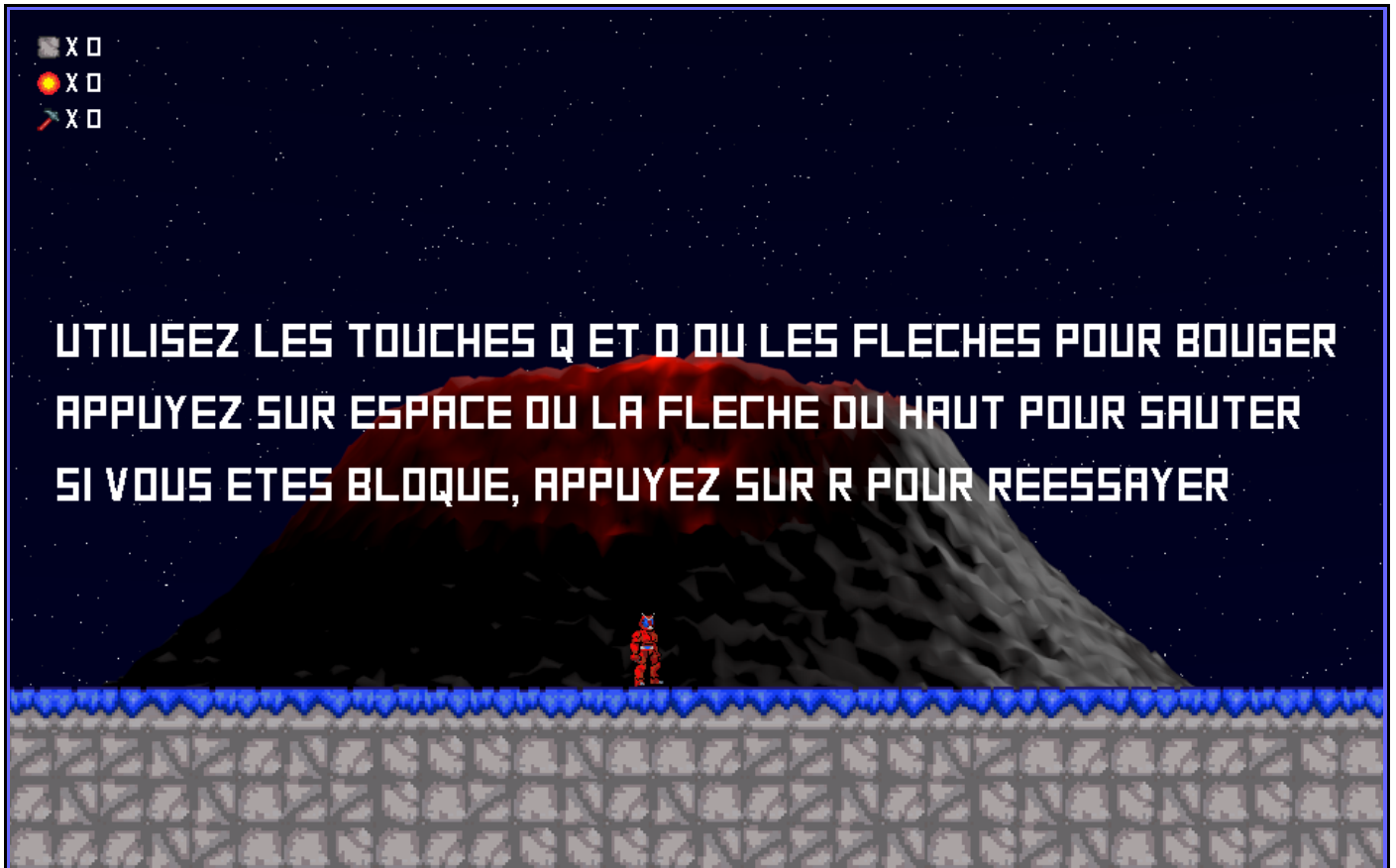
## **VI – Technologies utilisés :**

- Le langage de programmation Python dans sa version 3
- La bibliothèque Pygame pour la gestion de l'affichage et des sons
- Les logiciels Gimp, Blender et Adobe Photoshop pour la création des textures
- Le logiciel PxTone pour la musique du jeu
- Le logiciel bfxr pour les sons
- L'éditeur de texte Notepad++ pour la création des différents niveaux

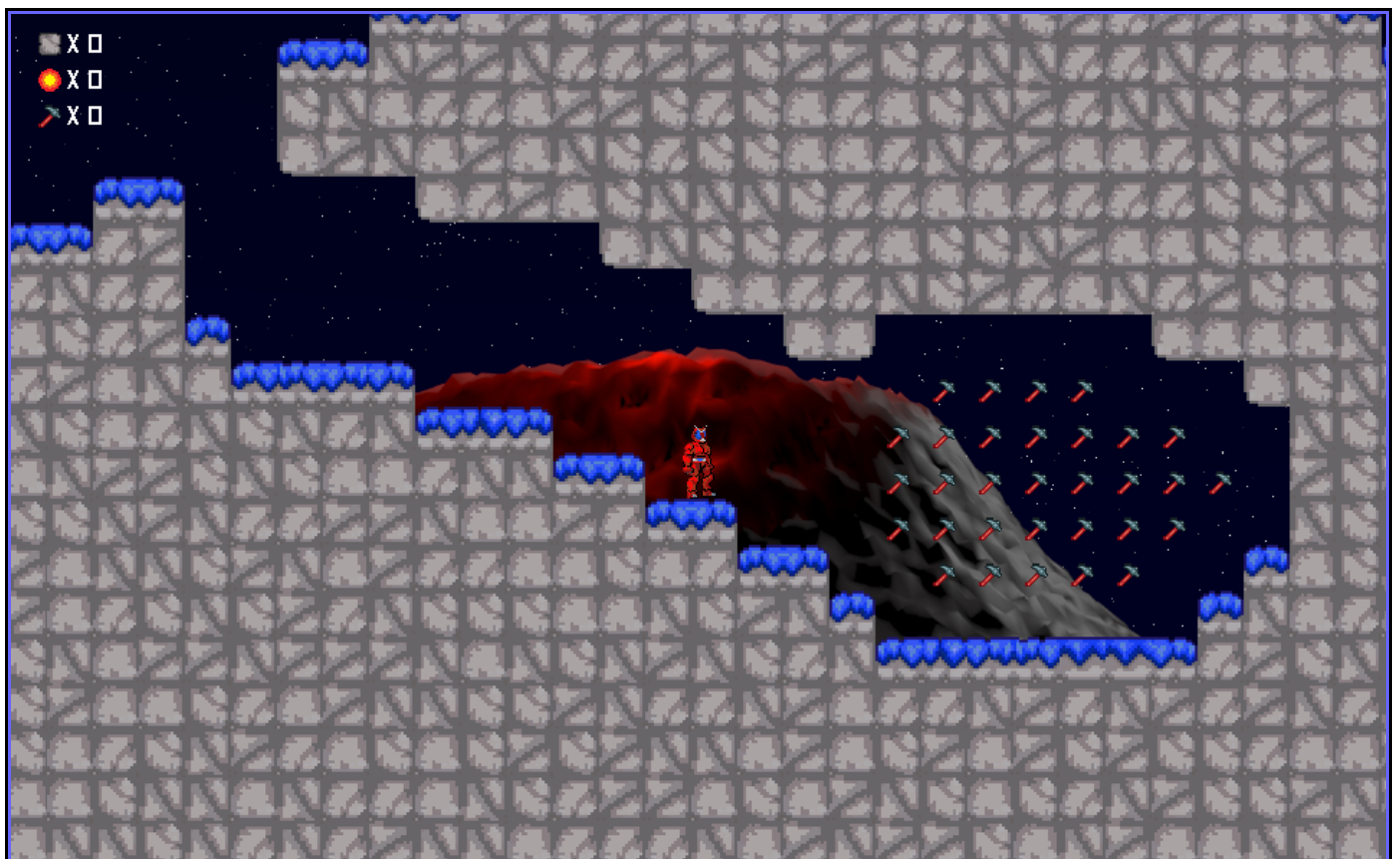
## Création du niveau 1 dans Notepad++



## Captures d'écran du jeu :

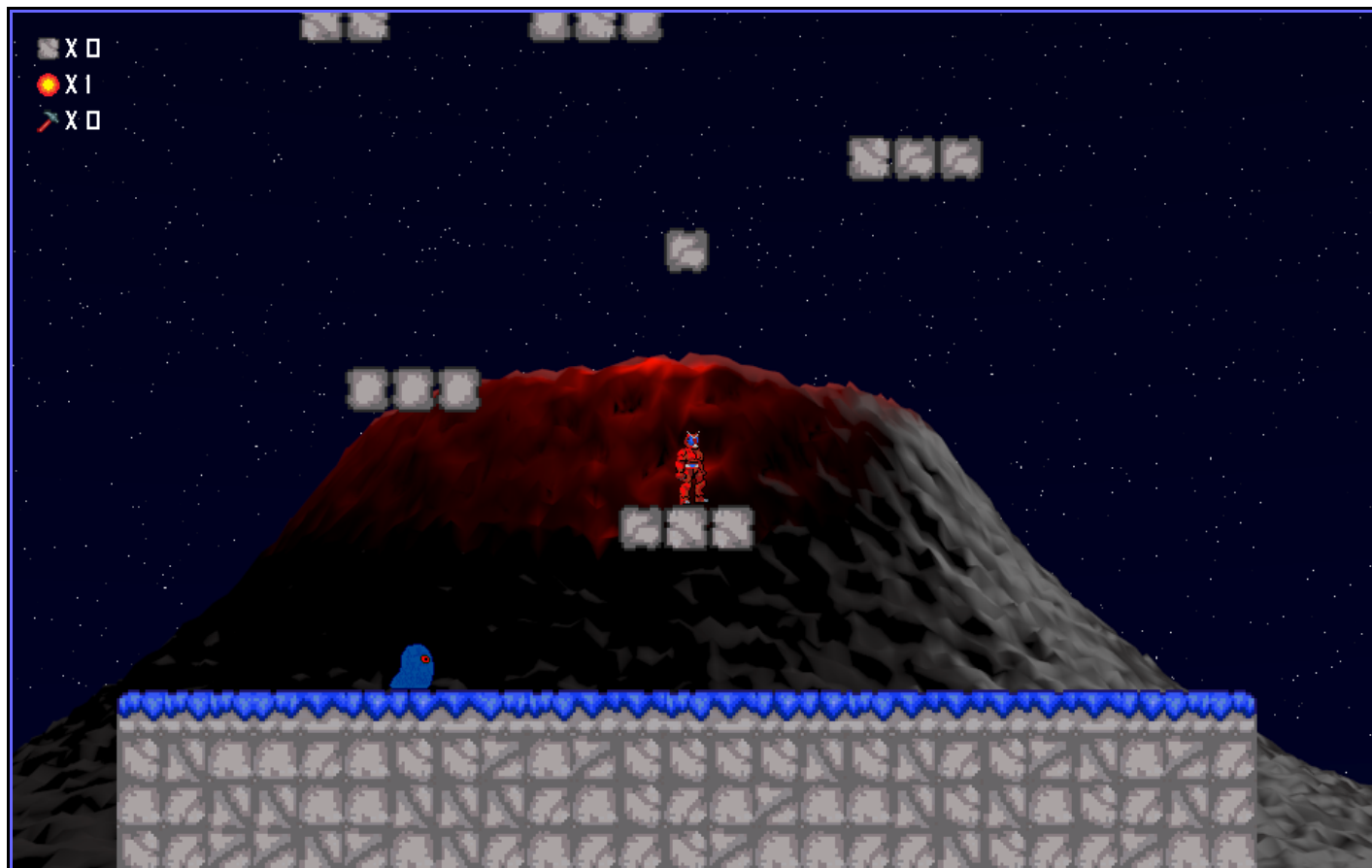


### Tutoriel

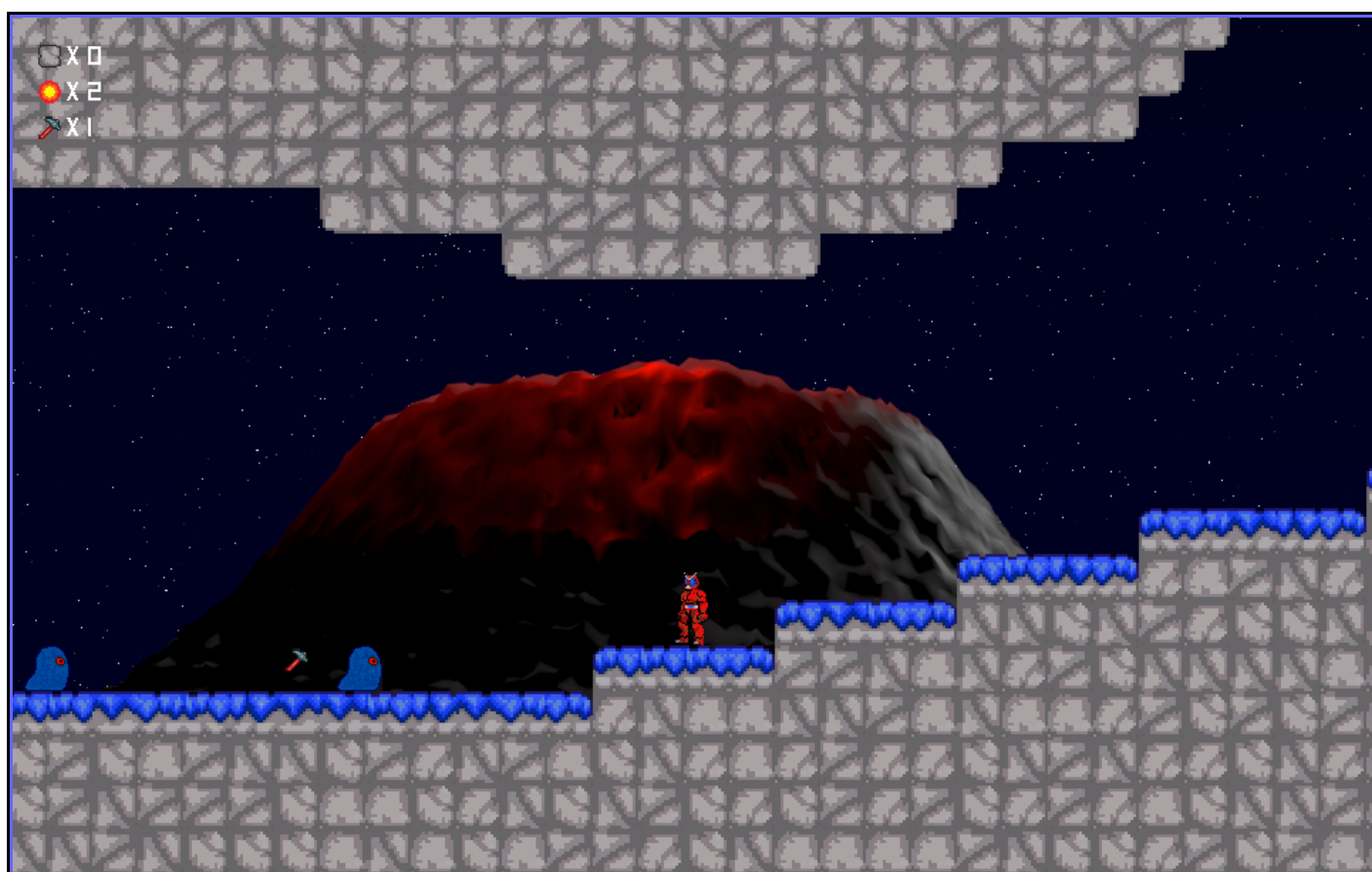


### Niveau 1





Niveau 2



Niveau 3



## Images du jeu :

