



La petite Aventure de Bloup

ANTINEA GONTARD
ADELINE DELBECQ

LEVELS

PLAY

Sommaire



1. Rappel du contexte Page 3

2. But du jeu Page 3

3. Comment jouer Page 3

4. Comment l'installer Page 4

5. Aspects techniques

5.1 Base de données Page 4

5.2 Défis techniques

5.2.1 PathFinding Page 5

**5.2.2 Génération de niveaux
aléatoires** Page 6

PLAY

LEVELS



1. Rappel du contexte

Dans le cadre de notre formation au sein du campus Ynov, l'établissement demande de fournir un projet englobant plusieurs défis techniques. Pour notre projet de fin d'année nous devons développer un jeu vidéo comprenant :

- Une architecture en services
- Utilisation d'au moins 3 designs patterns
- Une base de données
- L'utilisation d'un ORM
- Au moins un défi technique

En effet ce jeu vidéo est un jeu vidéo de type Plateforme 2D sur PC.

2. But du jeu

Vous incarnez un petit personnage: **Bloup**. Celui-ci doit traverser les nombreux obstacles sur son chemin pour arriver jusqu'au boss qu'il doit vaincre pour finir le niveau et pouvoir passer au suivant. Où qu'il aille, le boss le suivra et tentera par tous les moyens d'arrêter l'aventure de notre jeune héro.

3. Comment jouer?

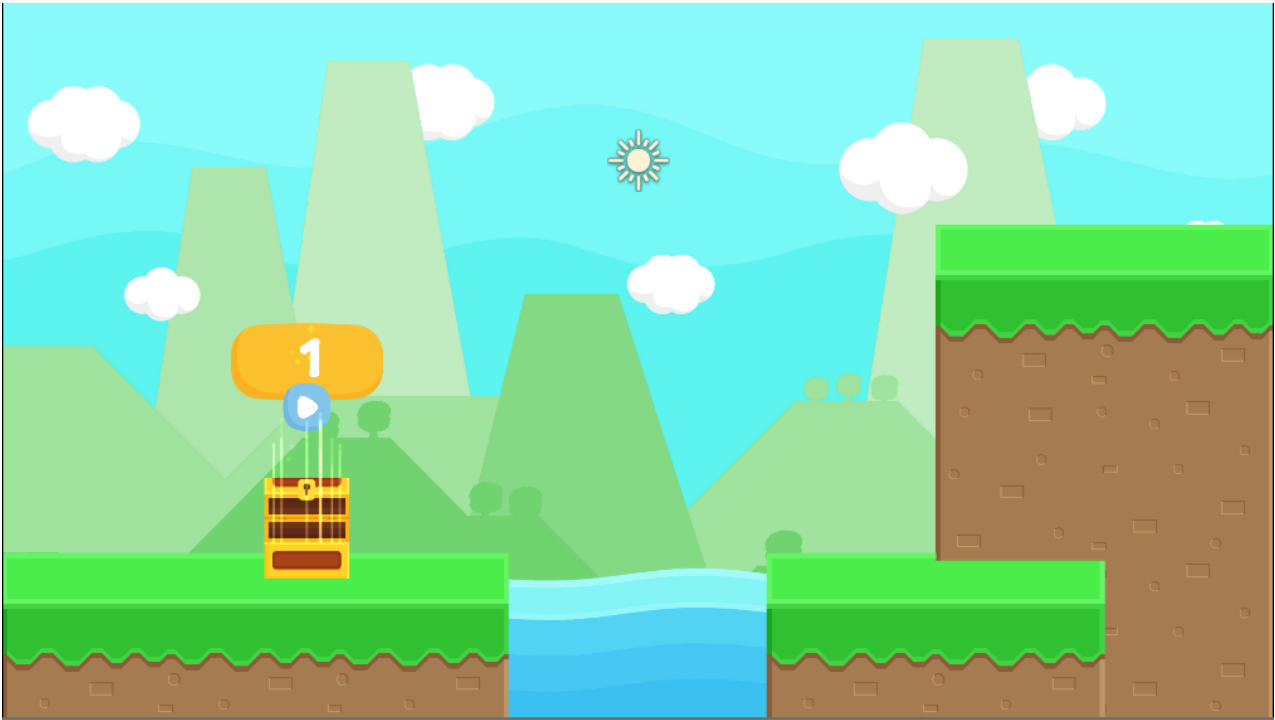
Au début du jeu vous aurez un menu comme ci-dessous qui s'affichera:



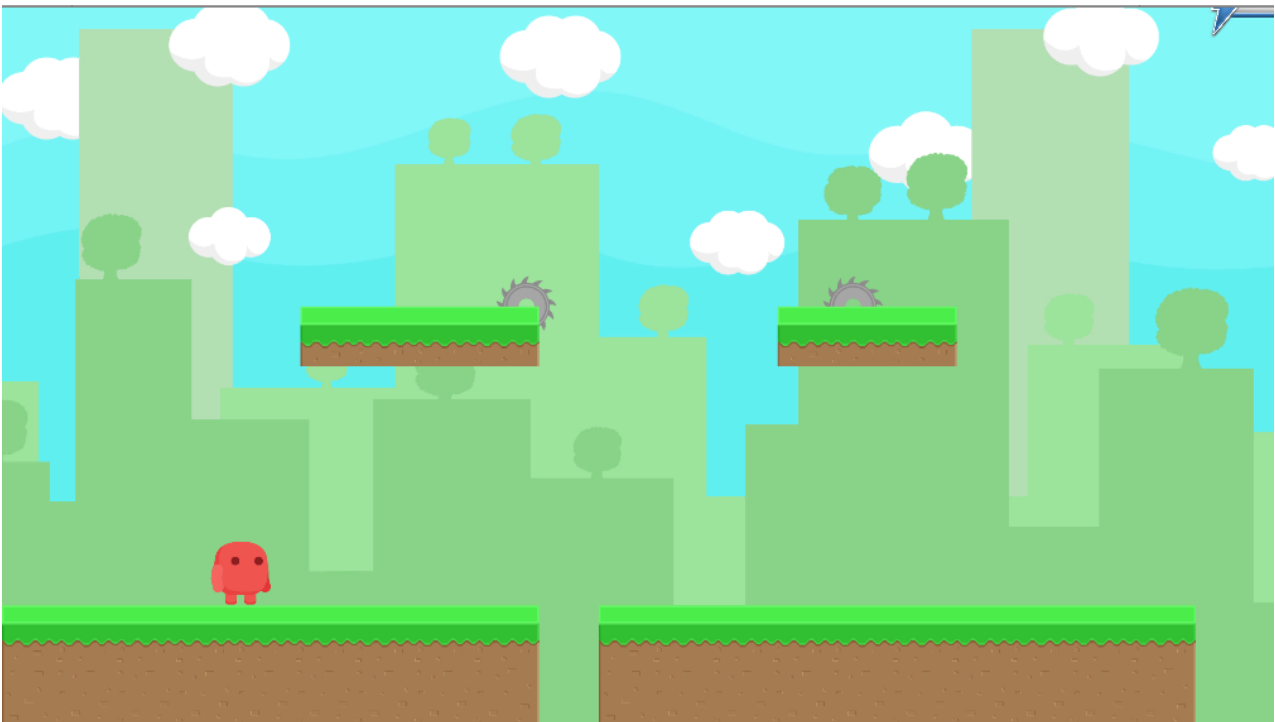


Pour jouer, cliquez sur le bouton **Play**.

Vous arriverez sur le menu de choix de niveau comme ci-dessous, pour commencer le niveau cliquez sur le bouton play bleu:



Le niveau est généré aléatoirement selon des paramètres donnés en début de niveau:

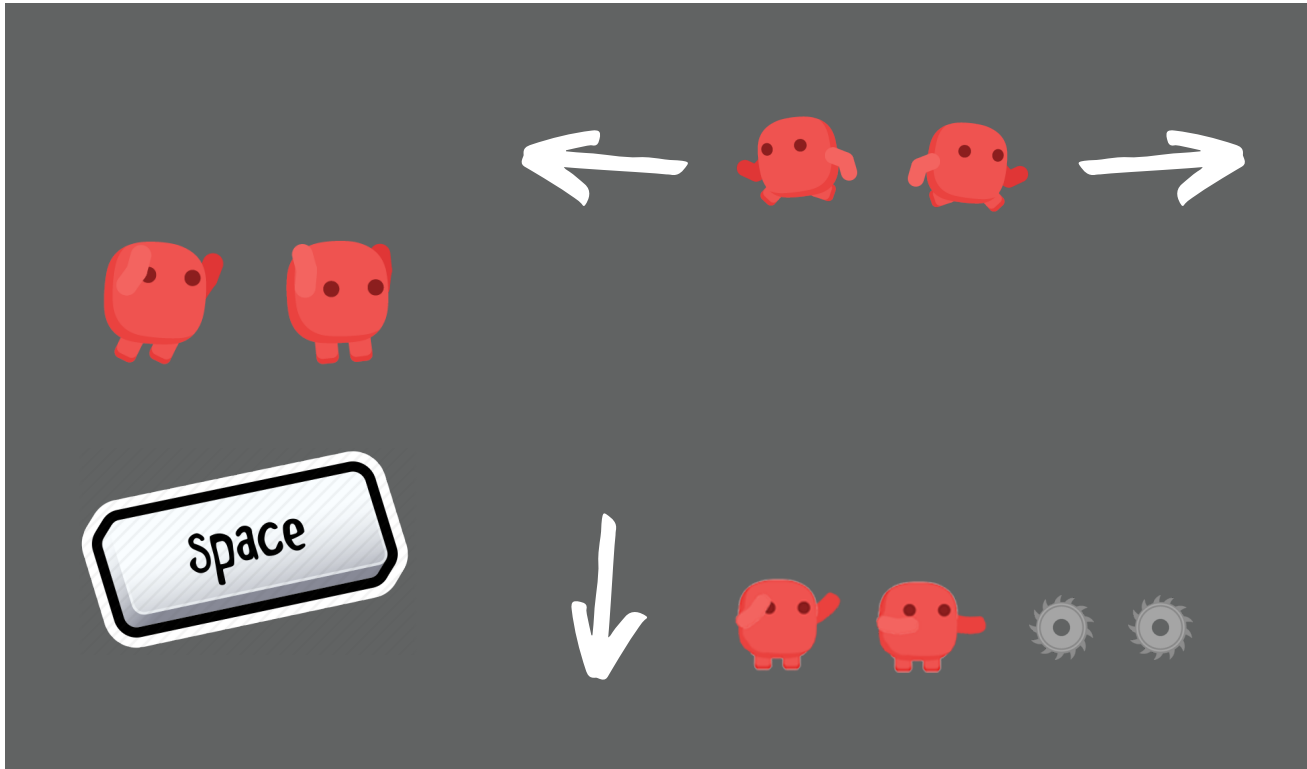




Les touches directionnelles permettent de se déplacer de chaque côté:

La touche espace permet de sauter.

La touche directionnelle du bas permet d'attaquer. L'attaque du bloup consiste à lancer des scies devant lui



4. Comment l'installer

En téléchargeant le jeu vous trouverez joint un exécutable. Pour jouer il suffit de lancer cet exécutable et de profiter du jeu.

5. Aspects techniques

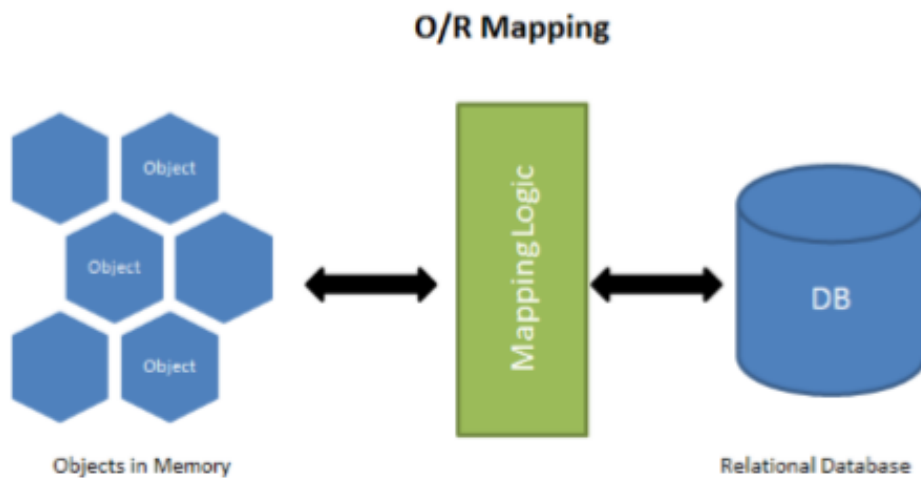
5.1 : Base de données

Pour sauvegarder les données des niveaux ainsi que du joueur et du boss nous avons mis en place une base de données **SQLite**.

En effet cette base de donnée locale nous paraissait idéale pour ce jeu puisqu'il va la sauvegarder sur le client.



L'ORM utilisé(unity) nous permet de traduire les données contenues en base en objet C# exploitables facilement.



Voici un aperçu des tables utilisées et leurs information:

- Boss
 - Nom
 - Points de vie
 - Points d'endurance
 - Multiplicateur de dégâts infligés
- Player
 - Nom
 - Points de vie
 - Multiplicateur de dégâts infligés
- Niveaux
 - Numéro,
 - Nombre d'ennemis
 - Nombre de Bonus
 - Taille

5.2 : Défis techniques

5.2.1 : Pathfinding

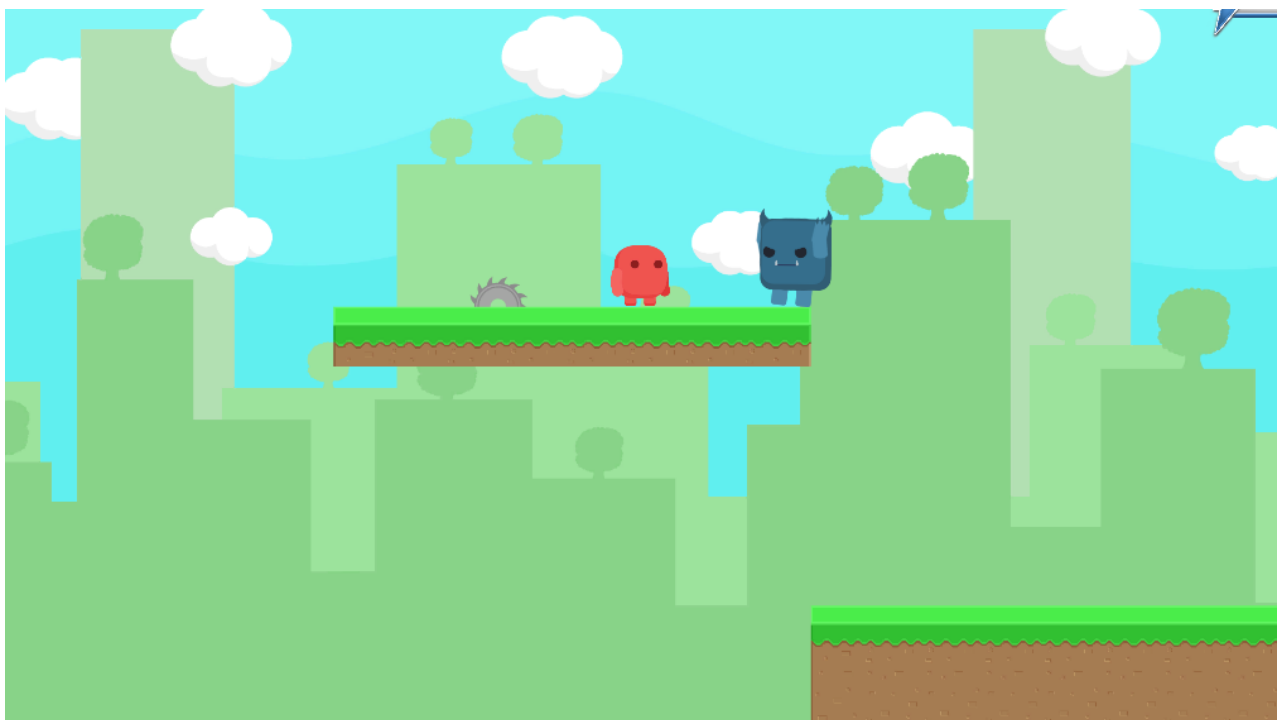
Nous avons mis en place plusieurs défi techniques. En effet, le boss de fin de niveaux est une intelligence artificielle qui cherchera toujours le chemin jusqu'au joueur du moment que celui-ci ait atteint la fin du niveaux: En effet il essaiera de barrer la route au joueur et d'aller jusqu'à lui en empruntant le chemin le plus court. L'algorithme utilisé pour le boss s'inspire du l'algorithme A*. En effet une grille est déployée sur toute la longueur du niveau.



Cette grille est fractionnée en case. Ces cases possèdent des liens entre elles. Par exemple, une case sur une plateforme aura un lien de saut avec une case d'une autre plateforme afin qu'il puisse sauter.

Deux cases entre elles d'une même plateforme peuvent avoir des liens pour pouvoir marcher jusqu'à l'autre case.

Ainsi l'algorithme va alors boucler entre ces cases et ces liens pour pouvoir récupérer la distance la plus courte jusqu'à la cible. L'algorithme fonctionne selon un système de coût qui est calculé avec la distance entre une case de la grille et une autre et avec le coût du lien entre une case et l'autre. Un lien pour marcher n'aura pas le même coût qu'un lien pour sauter.



5.2.2 : Génération de niveaux aléatoires

Chaque niveau est généré aléatoirement selon certains paramètres. Tout est calculé pour poser des plateformes à distance de saut du joueur pour qu'il puisse parcourir le niveau jusqu'à la fin. Il prend en base de donnée des paramètres de distance de saut, la taille de la map, et le nombre de pièges dans chaque niveau afin de générer le niveau selon ces critères. Le joueur apparaît au début du niveau et le boss apparaît à la fin du niveau.