

# Editeur de niveaux Prog & Play

## Documentation Utilisateur

Benjamin BONTEMPS

6 juin 2016

### Table des matières

<b>1</b>	<b>Lancement de l'éditeur</b>	<b>2</b>
<b>2</b>	<b>Launcher propre au mod sélectionné</b>	<b>3</b>
2.1	Création d'un nouveau niveau . . . . .	3
2.2	Modification d'un niveau existant . . . . .	3
2.3	Éditeur de scénario . . . . .	3
<b>3</b>	<b>Éditeur de niveaux</b>	<b>4</b>
3.1	Fichier . . . . .	4
3.2	Unités . . . . .	4
3.3	Zones . . . . .	6
3.4	Forces . . . . .	7
3.5	Évènements . . . . .	8
3.5.1	Évènements, conditions et actions . . . . .	8
3.5.2	Variables . . . . .	9
3.5.3	Paramètres particuliers . . . . .	10
3.6	Paramètres de la carte . . . . .	10
<b>4</b>	<b>Lancement du jeu créé</b>	<b>11</b>

# 1 Lancement de l'éditeur

Pour démarrer l'éditeur de niveaux de Prog & Play, il faut tout d'abord s'assurer que tous les fichiers nécessaires à son bon fonctionnement sont présents :

- L'archive *PPLE\_Launcher.sdz* doit être présente dans le répertoire *<Spring>/mods/* (82.5.1) ou *<Spring>/games/* (98+). Cette archive constitue le mod "Launcher", qui permet donc de lancer l'éditeur dans de bonnes conditions.
- Le dossier *pp\_editor* doit être présent à la racine de Spring. Il contient des fichiers nécessaires au lancement de l'éditeur (dans *editor\_files*), des fichiers nécessaires à la création de l'archive finale (dans *game\_files*) et les niveaux et scénarios créés (respectivement dans *missions* et *scenarios*). Il est à noter que les dossiers *editor\_files* et *game\_files* ne sont pas nécessaires dans la version 82.5.1.
- Au moins un mod pour Spring (*Kernel Panic 4.1* par exemple) doit être présent dans le répertoire *<Spring>/mods/* (82.5.1) ou *<Spring>/games/* (98+). Celui-ci servira de base pour créer les niveaux.

Une fois que ceci est fait, il suffit de lancer Spring, puis de lancer le mod *Prog & Play Level Editor Launcher 1.0*. (Il vous sera demandé de choisir une carte et une IA. Ces dernières n'ayant aucune influence sur la suite, n'importe lesquelles feront l'affaire.)

Enfin, il vous sera demandé de sélectionner un mod parmi ceux présents dans le dossier *<Spring>/mods/* (82.5.1) ou *<Spring>/games/* (98+) et qui ne sont pas en lien avec Prog & Play. Ce mod sert à définir le jeu de base sur lequel seront créés les niveaux pour Prog & Play (définition des unités et de leur comportement). Une fois ce mod sélectionné, Spring redémarrera sur l'éditeur à proprement parler.

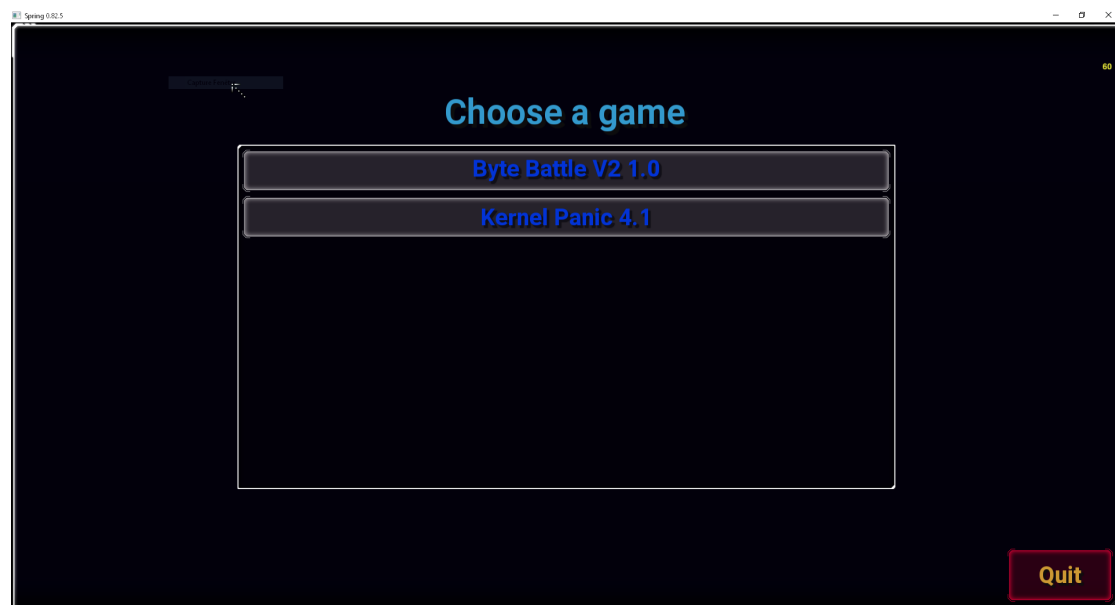


FIGURE 1 – Choix du mod de base pour l'éditeur.

## 2 Launcher propre au mod sélectionné

Le launcher se divise en 3 menus distincts : Création d'un nouveau niveau, Modification d'un niveau existant et Éditeur de scénario.

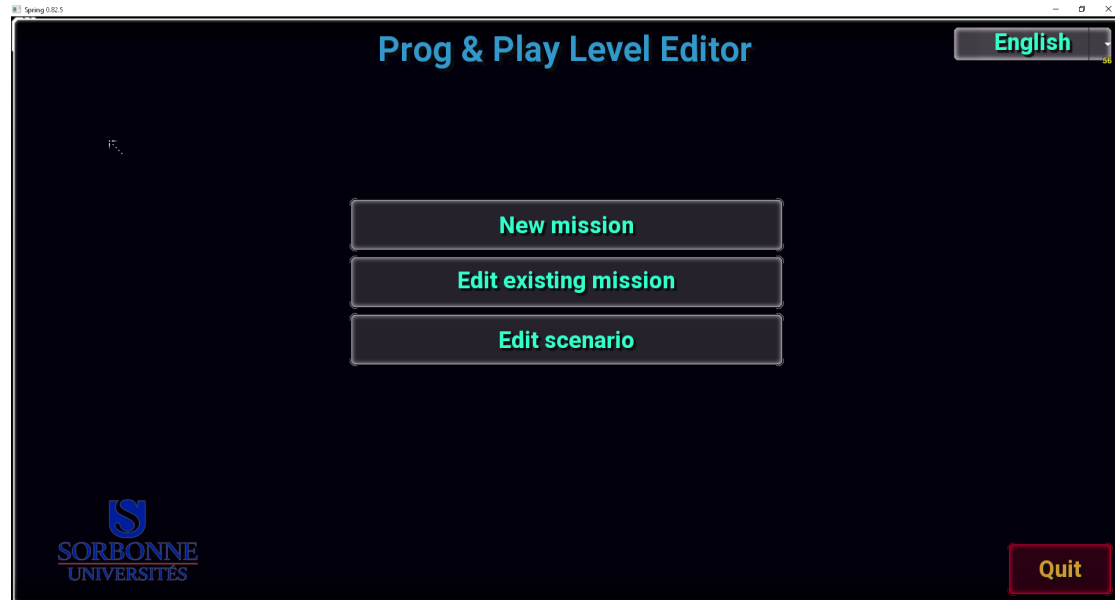


FIGURE 2 – Launcher de l'éditeur.

### 2.1 Création d'un nouveau niveau

Ce menu permet de choisir la carte sur laquelle on veut créer un nouveau niveau. L'ensemble des cartes disponibles correspond aux cartes présentes dans le répertoire `<Spring>/maps`.

### 2.2 Modification d'un niveau existant

Ce menu permet de choisir une mission préalablement créée via l'éditeur. L'ensemble des missions proposées correspond aux missions présentes dans le répertoire `<Spring>/pp_editor/missions`. Les missions recensées sont celles ayant été créées pour le mod de base préalablement choisi.

### 2.3 Éditeur de scénario

L'éditeur de scénario permet de scénariser les missions créées avec l'éditeur de niveaux. Chaque niveau est représenté par une fenêtre contenant le nom du niveau, un état d'entrée et un ou plusieurs états de sortie en fonction de ce qui a été défini dans l'éditeur. Les états de sortie peuvent être connectés à des états d'entrée. Il est possible de sauvegarder et de charger des scénarios (ces derniers se trouvant dans le répertoire `<Spring>/pp_editor/scenarios`). Enfin, la génération de l'archive finale du jeu créé avec les différentes missions et la scénarisation se fait dans ce menu.

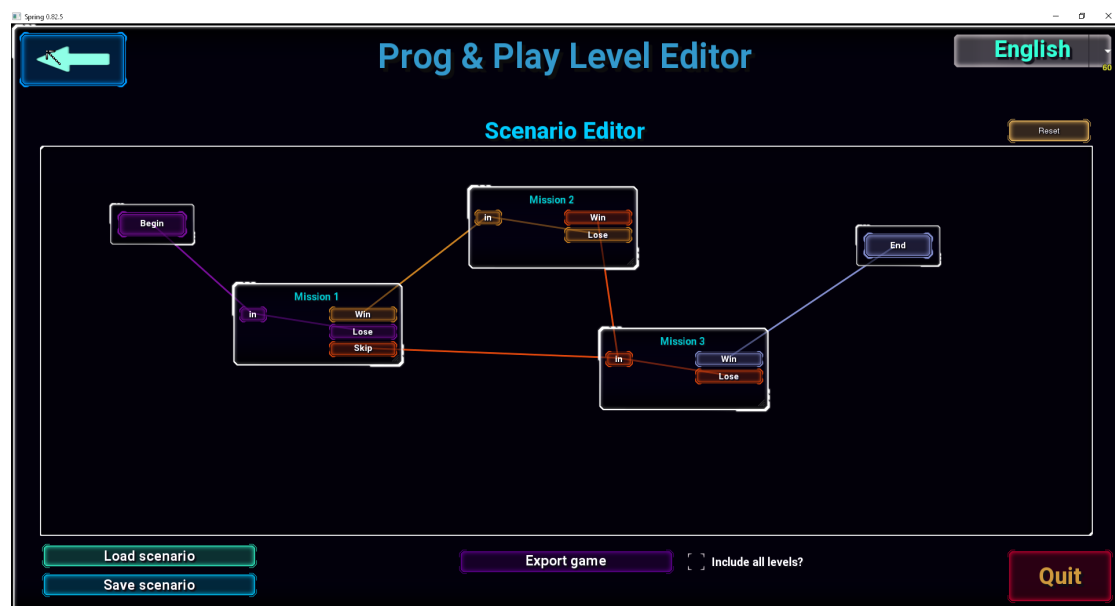


FIGURE 3 – Exemple d'édition d'un scénario à 3 missions.

### 3 Éditeur de niveaux

L'éditeur se décompose en plusieurs menus distincts. Les parties suivantes s'attachent à décrire le fonctionnement et l'intérêt de chacun des menus.

#### 3.1 Fichier

Les fonctions de ce menu sont assez classiques :

- **Nouveau** - Permet de créer un nouveau niveau sur une carte sélectionnée parmi une liste déroulante.
- **Ouvrir** - Permet de modifier un niveau ayant déjà été créé. Seuls les niveaux ayant été créés pour le mod de base sélectionné dans le launcher apparaitront dans la liste déroulante.
- **Enregistrer** - Permet de sauvegarder le niveau dans le fichier *<Spring>/pp\_editor/missions/<Nom du niveau>.editor*.
- **Menu principal** - Permet de retourner au launcher propre au mod sélectionné (voir 2).
- **Quitter** - Permet de quitter l'éditeur et Spring.

#### 3.2 Unités

Ce menu permet d'ajouter, de déplacer et de supprimer des unités, ainsi que de modifier certains de leurs attributs. Il est également possible de les ajouter à des groupes d'unités pour une utilisation ultérieure dans les événements.

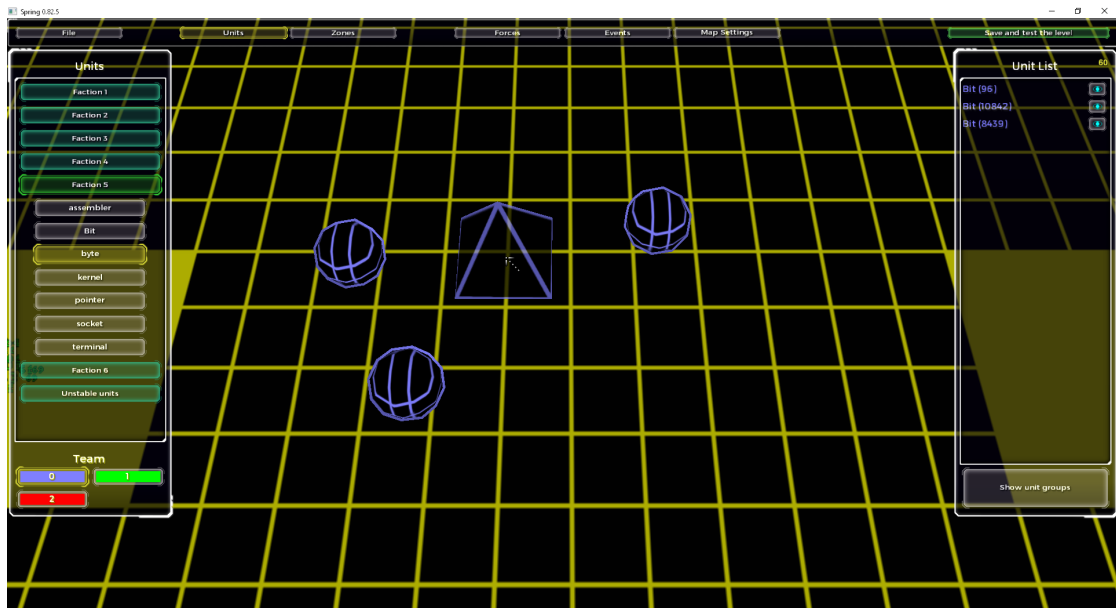


FIGURE 4 – Interface de gestion des unités.

Le menu de gauche permet de sélectionner un type d'unité et une équipe, et il est ensuite possible d'instancier une unité du type sélectionné qui appartient à l'équipe sélectionnée. Les types d'unité sont classés par faction. Une fois que des unités se trouvent sur le terrain, il est possible de les sélectionner pour :

- les déplacer par cliquer-glisser ou en utilisant les flèches directionnelles.
- les pivoter par cliquer-glisser en laissant enfoncer CTRL ou ALT.
- les supprimer en appuyant sur SUPPR.
- modifier certains de leurs attributs (appartenance à une équipe, nombre de points de vie initiaux, activation ou désactivation des soins automatiques) en accédant au menu contextuel via un clic droit.
- les ajouter ou les retirer d'un groupe d'unités.

Les groupes d'unités peuvent être gérés plus précisément en utilisant l'interface accessible par le menu de droite. Ces groupes sont purement logiques et servent à effectuer des actions ou à tester des conditions sur plusieurs unités en même temps.

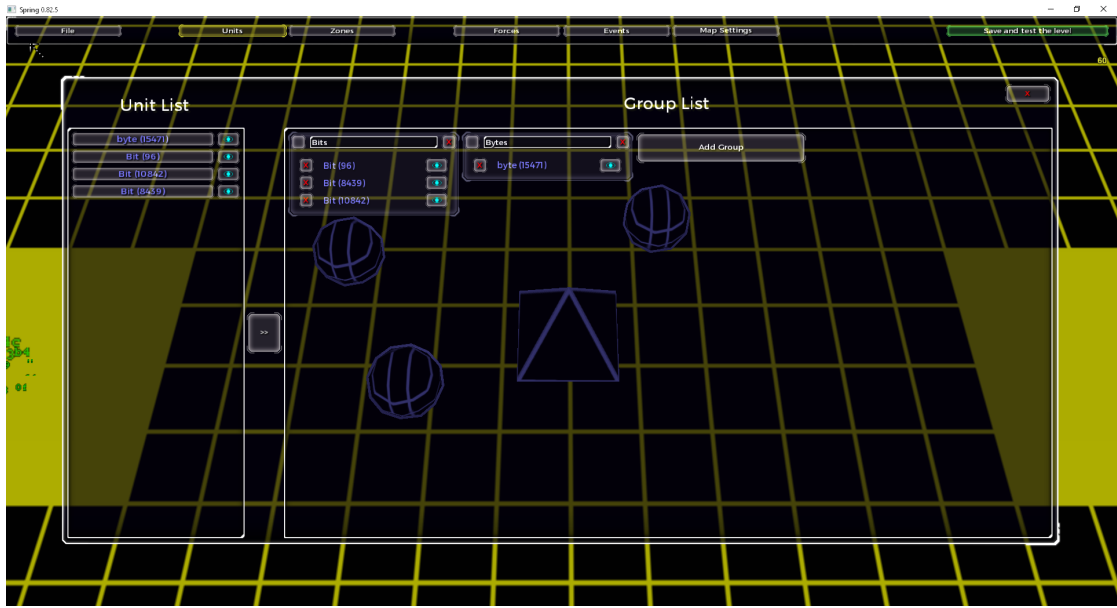


FIGURE 5 – Interface de gestion des groupes d'unités.

### 3.3 Zones

Ce menu permet de définir des zones logiques pour une utilisation ultérieure dans les événements.

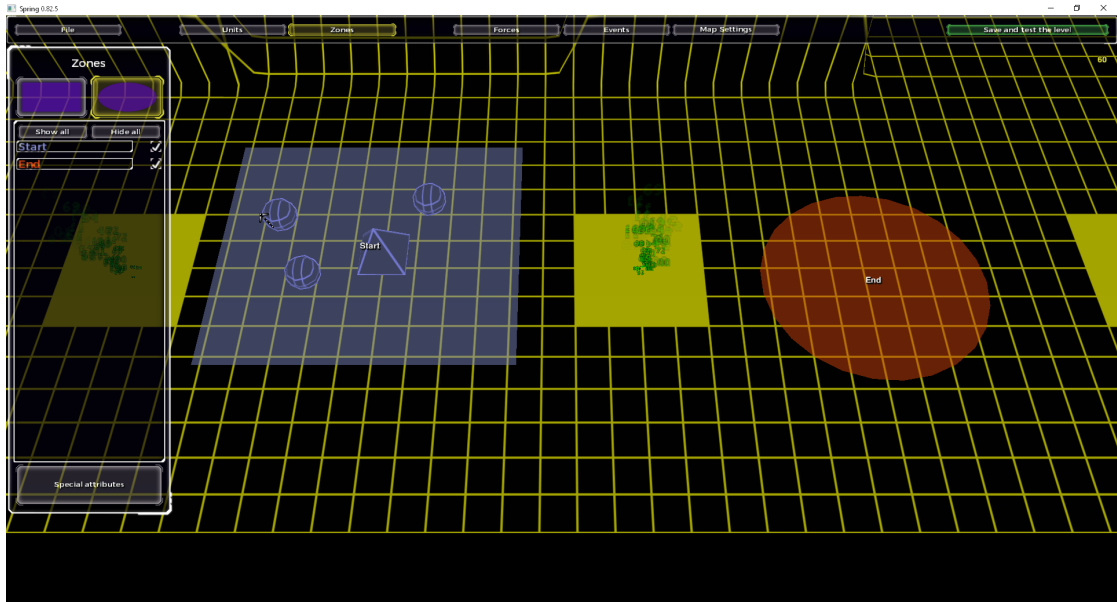


FIGURE 6 – Interface de gestion des zones logiques.

Les zones peuvent être rectangulaires ou elliptiques en fonction des besoins. De façon similaire aux unités, il est possible de les tracer, de les déplacer, de les redimensionner, de les renommer et de les supprimer. Chaque zone possède également des attributs spéciaux qui permettent d’afficher un marqueur portant le nom de la zone pendant l’exécution du jeu ou d’afficher en permanence le centre de la zone dans le champ de la caméra (dans le cas d’une caméra automatique).

### 3.4 Forces

Ce menu permet de gérer le statut des différentes équipes et les alliances.

Chaque équipe peut être activée ou désactivée et contrôlée par un joueur ou par l’ordinateur selon les besoins. Il est également possible de changer le nom de l’équipe ainsi que sa couleur pendant l’exécution du jeu (la couleur dans l’éditeur restera la même). Dans le cas d’une équipe contrôlée par l’ordinateur, il est possible de spécifier le nom d’une intelligence artificielle pour la contrôler (optionnel).

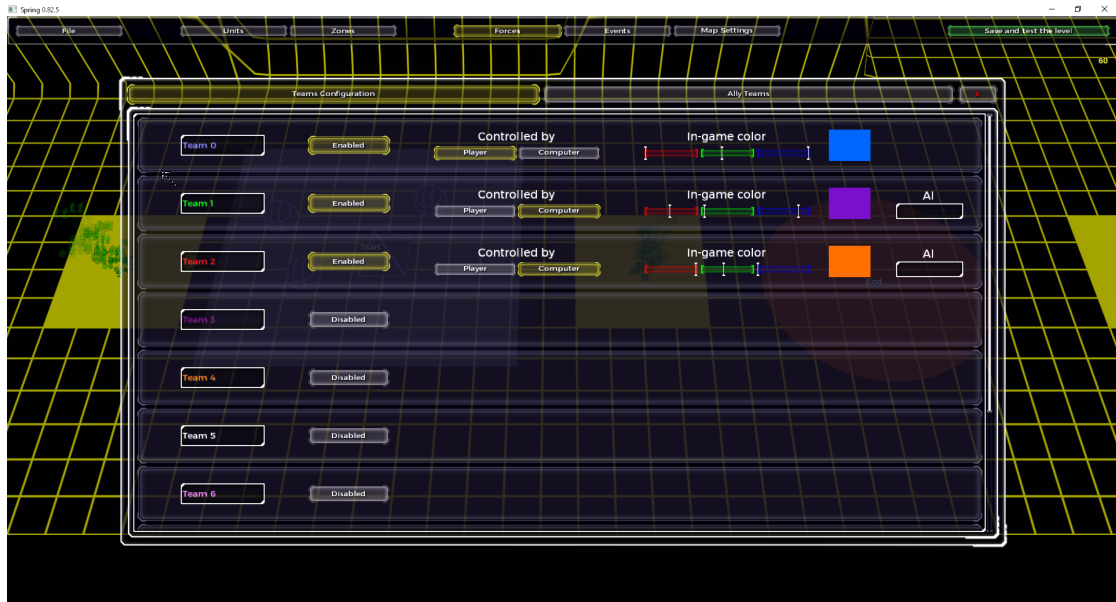


FIGURE 7 – Interface de configuration des équipes.

Les équipes peuvent être alliées entre elles. Il faut néanmoins noter que la notion d’alliance n’est pas réciproque : l’équipe 1 peut considérer l’équipe 2 comme son alliée tandis que l’équipe 2 considère l’équipe 1 comme ennemie.

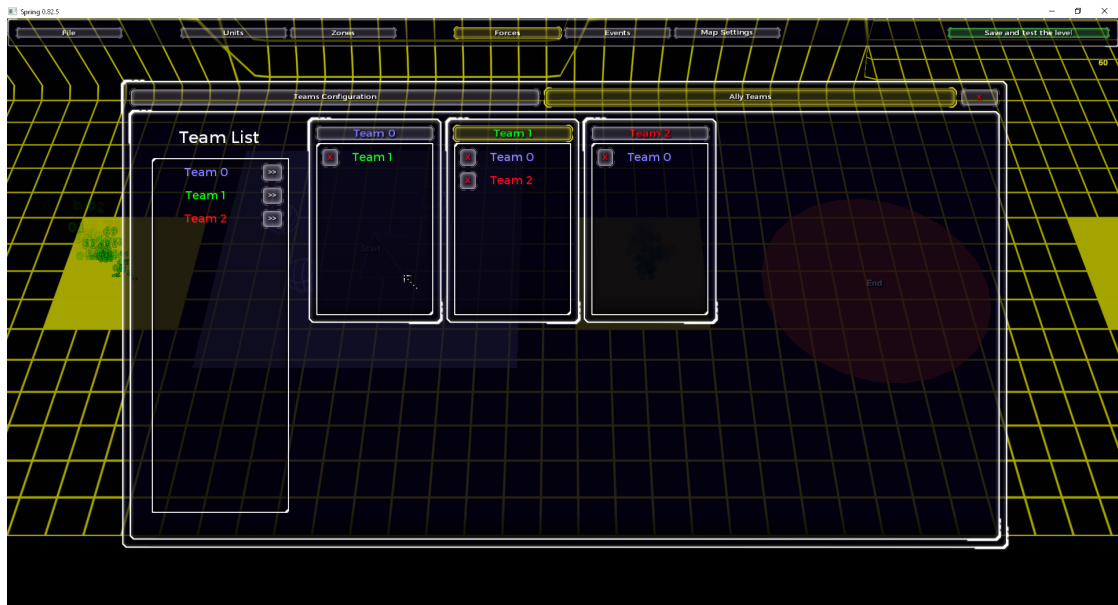


FIGURE 8 – Interface de gestion des alliances.

### 3.5 Évènements

Ce menu permet de gérer les différents évènements s'effectuant au cours de la mission.

#### 3.5.1 Évènements, conditions et actions

Un évènement correspond à un couple ensemble de conditions / ensemble d'actions. Il est possible d'ajouter des conditions et des actions puis de les paramétrer en utilisant les boutons et fenêtres appropriés.

Ensuite, il est possible de paramétrer l'évènement en définissant un déclencheur qui correspond à une expression logique entre toutes les conditions. Par défaut, le déclencheur est un ET logique global.<sup>1</sup> Lorsque le déclencheur renvoie *"true"*, les actions s'effectuent l'une après l'autre, dans l'ordre où elles ont été définies. Cet ordre est paramétrable dans la même fenêtre que la définition du déclencheur.

Il est également possible de changer les paramètres de répétition d'un évènement, ainsi que d'ajouter un commentaire qui pourra servir lors de la modification ultérieure du niveau.

1. Par exemple, pour un évènement possédant 3 conditions C1, C2 et C3, le déclencheur par défaut sera (C1 ET C2 ET C3).



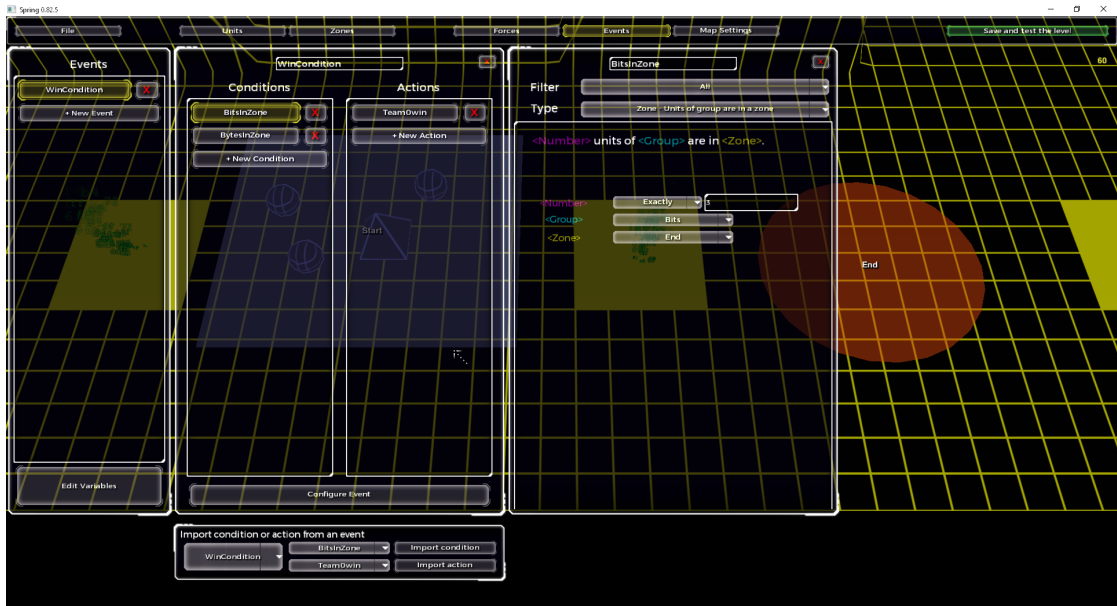


FIGURE 9 – Exemple d'édition de la condition *BitsInZone* de l'événement *WinCondition*.

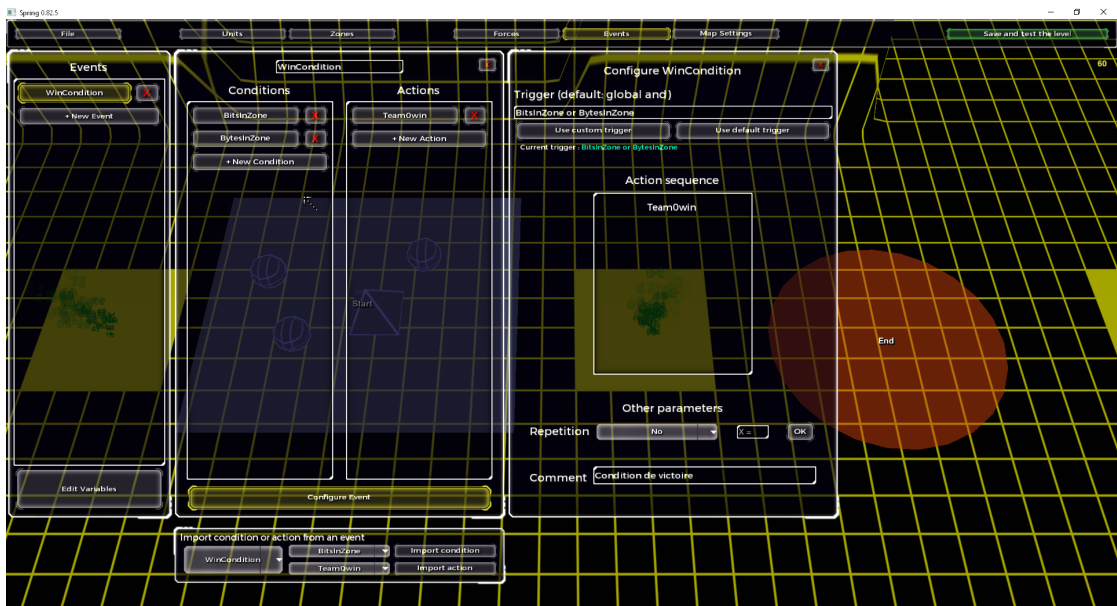


FIGURE 10 – Interface de gestion d'un événement.

### 3.5.2 Variables

L'éditeur d'événements permet de définir des variables numériques ou booléennes. On accède à l'interface d'édition des variables en utilisant le bouton situé en dessous de la liste des événements. Ces variables permettent d'avoir plus de contrôle sur la façon dont s'enchaînent les

différents événements. Leurs valeurs peuvent être modifiées en utilisant des actions spécifiques aux variables.

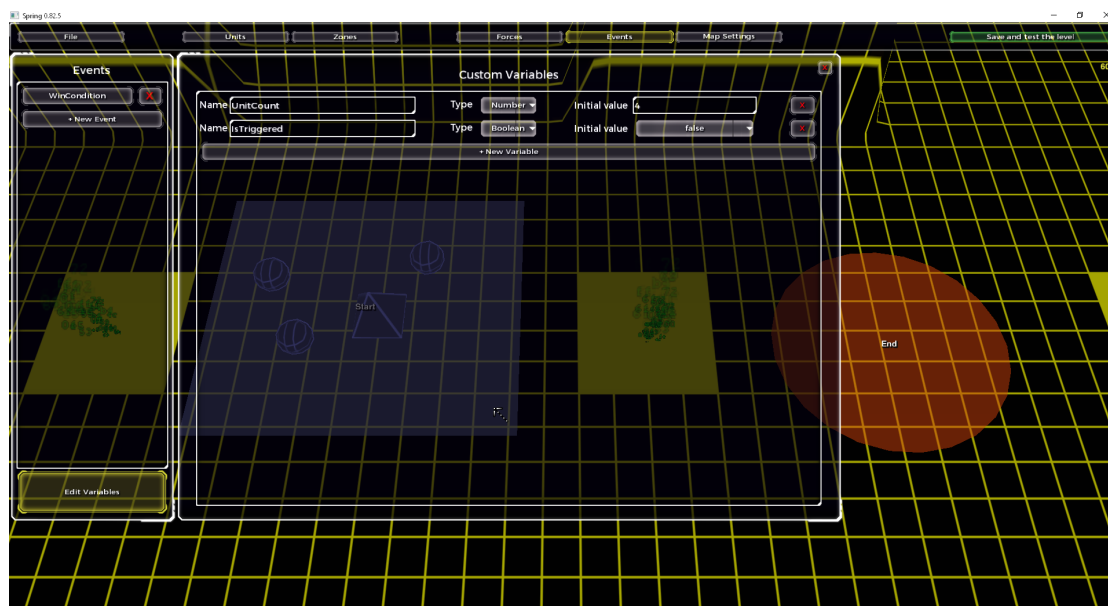


FIGURE 11 – Interface de gestion des variables.

### 3.5.3 Paramètres particuliers

Certains paramètres définissables dans les conditions et dans les actions requièrent des explications supplémentaires :

- **Unité** - Une unité peut soit être une unité déjà présente sur le terrain, soit une ou plusieurs unités créées par le dernier appel à une action de création d'unité(s). Une fois qu'une unité a été sélectionnée, il est possible de zoomer dessus en cliquant sur le bouton avec l'œil.
- **Groupe d'unités** - Un groupe peut correspondre soit à un groupe défini dans l'interface des groupes d'unités, soit à l'ensemble des unités créées par l'ensemble des appels à une action de création d'unité(s).
- **Position** - Une position peut être soit un couple (x, y), soit une position aléatoire à l'intérieur d'une zone. Une fois qu'une position a été sélectionnée, il est possible de zoomer dessus en cliquant sur le bouton avec l'œil.
- **Script** - Les paramètres de ce type s'adressent à des utilisateurs avancés. Le champ texte permet d'écrire du code en Lua permettant de définir totalement une condition<sup>2</sup> ou une action personnalisée.

## 3.6 Paramètres de la carte

Ce menu permet de choisir un nom pour le niveau, d'écrire un briefing qui sera affiché au début de la mission, et de spécifier certains paramètres.

2. Le script défini dans le cas d'une condition doit nécessairement renvoyer une valeur booléenne.

Le briefing peut contenir du texte coloré. Il faut pour cela choisir une couleur grâce aux sliders, sélectionner le texte à colorer, puis appuyer sur le bouton correspondant.

Il est également possible d'activer ou de désactiver :

- la caméra automatique (en jeu, cette caméra contient en permanence dans son champ de vision les unités visibles par le joueur ainsi que les zones définies comme toujours présentes dans le champ de vision)
- le soin automatique des unités
- la souris
- la minimap
- certains widgets propres au mod sélectionné ou à Prog & Play

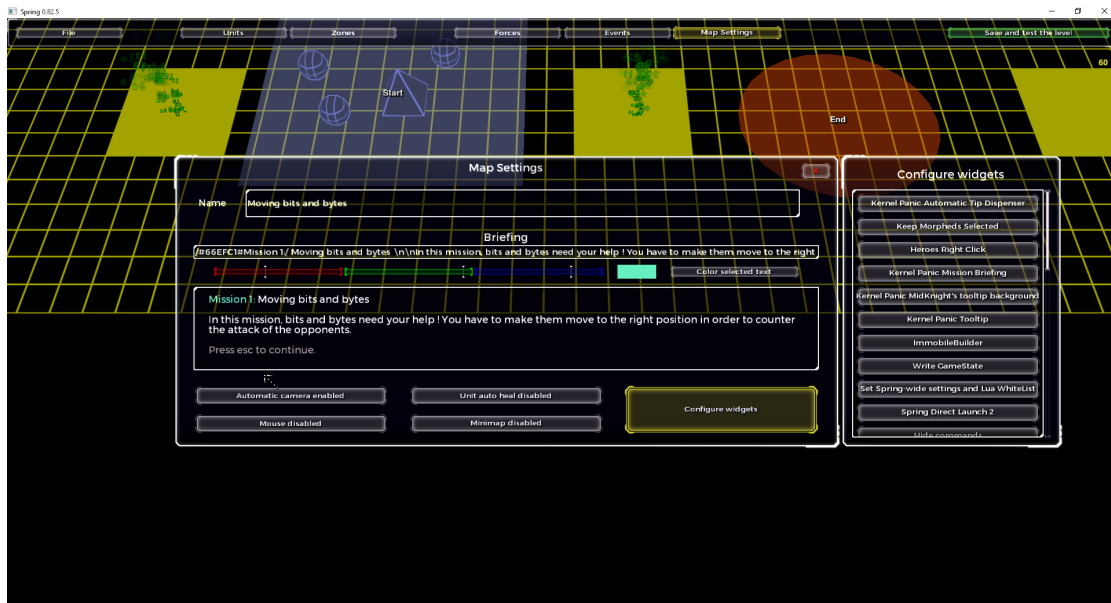


FIGURE 12 – Interface de modification des paramètres de la mission.

## 4 Lancement du jeu créé

Une fois que les niveaux ont été créés, que le scénario a été défini et que le jeu a été packagé, il devient jouable. Il suffit alors de le placer dans le répertoire `<Spring>/mods/` (82.5.1) ou `<Spring>/games/` (98+) accompagné du jeu duquel il dépend (Kernel Panic 4.1 par exemple). Ensuite, il suffit de lancer Spring et de sélectionner le mod correspondant.

Pour jouer en utilisant l'API Prog & Play (et contrôler les unités par des lignes de code), assurez-vous de bien posséder une version de Spring intégrant Prog & Play<sup>3</sup>.

3. Disponible sur <https://www.irit.fr/ProgAndPlay/>.