

UNIVERSITÉ DE MONTPELLIER

HMIN319 - JEUX SÉRIEUX

GAME DESIGN DOCUMENT

VERSION 0.1

---

# Everything is Fine

---

*Auteurs :*  
Edouard BREUILLE  
Florian CARNAC  
Issam FATAH  
Rémi COLIN DE  
VERDIERE  
Célia ROUQUAIROL



20 décembre 2017



# Table des matières

<b>1</b>	<b>Game overview</b>	<b>1</b>
1.1	Titre . . . . .	1
1.2	Concept . . . . .	1
1.3	Genre . . . . .	1
1.4	Look and Feel . . . . .	1
1.5	Références . . . . .	2
1.5.1	Les mécaniques de jeu . . . . .	2
1.5.2	Graphismes . . . . .	4
1.5.3	L’ambiance sonore . . . . .	5
<b>2</b>	<b>Game Context</b>	<b>6</b>
2.1	Public visé . . . . .	6
2.2	Game Appartus . . . . .	6
<b>3</b>	<b>Game scenes Storyboard</b>	<b>7</b>
3.1	Les directives du pompier Bonoeil . . . . .	7
3.2	Une première expérience . . . . .	8
<b>4</b>	<b>Game Distribution</b>	<b>13</b>
<b>5</b>	<b>Gameplay</b>	<b>14</b>
5.1	Acteurs . . . . .	14
5.1.1	Personnages . . . . .	14
5.1.2	Objets . . . . .	14
5.2	Actions . . . . .	14
<b>6</b>	<b>Game interface</b>	<b>16</b>
6.1	Camera . . . . .	16
6.2	Controls . . . . .	16
6.3	HUD . . . . .	17
6.3.1	La phase de construction . . . . .	17
6.3.2	La phase de simulation . . . . .	18
6.4	Feedback . . . . .	19
6.5	Audio, music, sound effects . . . . .	20
6.6	Help system . . . . .	20

<b>7</b>	<b>Game Mechanics</b>	<b>21</b>
7.1	Game OCR . . . . .	21
7.2	Game Rules . . . . .	22
7.3	Game Flow . . . . .	22
7.4	Game AI . . . . .	22
7.5	Replaying and saving . . . . .	23
<b>8</b>	<b>Game Model</b>	<b>24</b>
8.1	AI . . . . .	24
8.2	Interface . . . . .	26
<b>9</b>	<b>Game Story</b>	<b>28</b>
9.1	Story and narrative . . . . .	28
9.2	Game world . . . . .	28
9.3	Characters traits . . . . .	29
9.4	Game levels . . . . .	29
<b>10</b>	<b>Game Aesthetics</b>	<b>30</b>

# 1 Game overview

## 1.1 Titre

*"Everything is Fine"*

Contrebalancer la gravité des situations mises en place en rassurant le joueur : tout va bien !

## 1.2 Concept

Ce jeu a pour but de sensibiliser ses utilisateurs aux normes incendie déployées dans les **E**tablissements **R**ecevant du **P**ublic.

*"Les établissements recevant du public (ERP) sont des bâtiments dans lesquels des personnes extérieures sont admises. Peu importe que l'accès soit payant ou gratuit, libre, restreint ou sur invitation."*  
service-public.fr

## 1.3 Genre

Le projet se découpe en deux parties distinctes :

- Une phase de gestion/puzzle : en respectant un budget limité, le joueur doit placer au mieux des éléments qui aideront les visiteurs d'un bâtiment à évacuer en cas d'incendie.
- Une phase de simulation : alors que le-dit bâtiment est la proie des flammes, toutes les personnes présentes vont donc devoir évacuer. Le joueur aura-t-il correctement signalé la sortie et les outils pour maîtriser le feu ?

## 1.4 Look and Feel

Cette expérience est à double tranchants. A la manière des Sims, le joueur pourra très bien s'attacher à toutes ces vies qui sont entre ses mains et donc s'appliquer et tout mettre en oeuvre pour les sauver. Il en ressentira alors une grande joie s'il réussit mais également de la peine si certaines personnes venaient à rester bloquées dans le bâtiment. Ce sera alors pour lui l'occasion de recommencer et chercher à s'améliorer. D'un tout autre côté, toujours en reprenant les Sims, certains joueurs développeront leur mauvais côté "Go-dLike" en induisant les visiteurs en erreur, leur bloquant volontairement le

passage, leur recommandant de se regrouper à la plus reculée des salles du plus haut étage de l'établissement. Au contraire du premier type de joueur, celui-ci ressentira de la satisfaction à piéger des personnes et pourra ainsi répéter les niveaux pour condamner le plus de gens possible.

Quoi qu'il en soit, dans chacun de ces cas, la connaissance et la bonne compréhension des normes incendie est nécessaire : qu'on aiguille les personnes vers la sortie ou sur le toit, on apprend qu'un panneaux de signalisation indique le chemin à suivre.

## 1.5 Références

### 1.5.1 Les mécaniques de jeu

Comme évoqué plus haut, on pensera bien évidemment aux Sims (figure 1) : armé d'un panneau de contrôle comprenant tous les objets possibles à placer dans un univers virtuel, le joueur, limité par un budget influe sur le monde et donc le comportement des gens qui y existent.



FIGURE 1 – Sims 4 : Build mode

Un autre jeu dont les mécaniques nous ont inspirées : PolyBridge (figure 2).

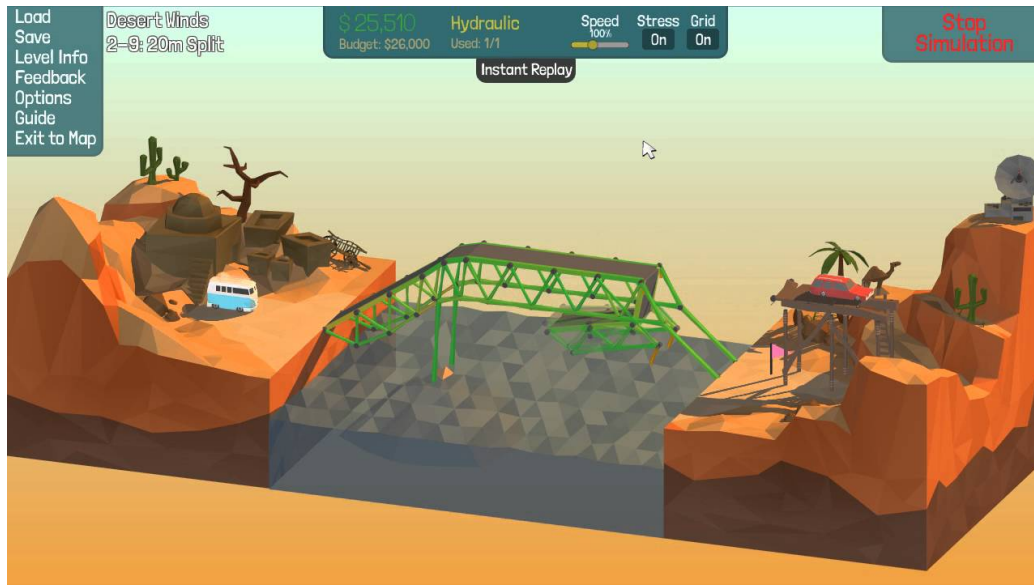


FIGURE 2 – PolyBridge : phase de simulation

Ici nous allons retrouver un élément important : la séparation entre la phase de jeu et la phase de simulation. Dans un premier temps le joueur doit construire un pont pour permettre aux voitures de traverser un fleuve. Une fois qu'il est satisfait de sa construction, il peut alors la tester en laissant les véhicules avancer. Si ils atteignent tous l'autre côté, alors le niveau est considéré comme réussi. En adaptant le vocabulaire et remplaçant les acteurs pour se placer dans le cadre de l'évacuation lors d'incendie, nous retrouvons toutes les mécaniques de notre jeu.

### 1.5.2 Graphismes

Simplistes mais néanmoins efficace dans sa représentation du monde et de ses agents, Meet my Company (figure 3) est l'une de nos inspirations pour les graphismes de notre jeu.



FIGURE 3 – Meet my Company

Dans ce serious game où le but est de se familiariser avec les différents membres d'une entreprise, nous allons retrouver un lieu comprenant des salles/bureaux, des couloirs et une entrée/sortie. Les murs ne diminuent pas la perception de l'environnement même s'ils ne disparaissent pas, il est toujours possible de trouver les différents agents. Chaque élément est clairement identifiable et il est donc facile d'interagir avec.



### 1.5.3 L’ambiance sonore

Le jeu va se découper en deux parties :

- Le mode de construction, où le joueur apprendra les normes de sécurités et devra les appliquer.
- La simulation en elle même, où les agents devront utiliser les voies de secours correctement indiquées (ou non) par le joueur.

L’ambiance sonore devra accompagner ces deux situations. Ainsi, lors de la première phase, le joueur sera dans une ambiance calme et douce, lui laissant le loisir de réfléchir et optimiser son budget en équipant au mieux l’établissement du niveau. On pensera notamment aux musiques des Sims ou de Poly bridge : nous reprendrons les codes de l’ambient, que l’on nomme vulgairement la musique d’ascenseur.

*La musique ambient doit être capable d’accommoder tous les niveaux d’intérêt sans forcer l’auditeur à écouter ; elle doit être discrète et intéressante.*

Brian Eno

Dans la second phase, le joueur doit ressentir l’état émotionnel des agents qu’il doit sauver, il faut instaurer une ambiance stressante.

La musique va alors progressivement se modifier :

- Le tempo accélérera en fonction du temps : plus celui-ci défile, plus les personnages sont en danger ;
- Plus d’instruments seront présents pour créer une ambiance plus pesante ;
- Quelques notes dissonantes feront leur apparition lors d’évènements tragiques, comme la perte d’un agent par exemple.

Elle est liée au niveau de panique générale et servira de feedback sonore au joueur.

## 2 Game Context

### 2.1 Public visé

Le jeu a pour ambition d'être "tout public". En effet, ayant un objectif de sensibilisation, nous ne voulons pas le réserver à une catégorie particulière : il sera donc abordable par tous. Chacun pourra en tirer une expérience différente : sauver ou piéger les personnages, recommencer les niveaux pour obtenir la note maximale, découvrir les niveaux, autant de possibilités d'explorer le jeu.

### 2.2 Game Appartus

Pour le développement, nous utiliserons Unity ainsi que plusieurs plugins :

- ProBuilder pour la création facilitée et à grande échelle des niveaux et la gestion de la transparence des murs selon la caméra pour permettre au joueur de bien visualiser son environnement ;
- LMMS et FMOD pour créer, intégrer et modifier la musique et la faire varier selon l'état du jeu.
- Fire Propagation, un asset pour la gestion du feu et de son développement.

Pour une diffusion PC, la gestion des contrôles pourra se faire au clavier ou à la souris.

Pour une diffusion mobile, les contrôles seront affichés à l'écran et rendus tactiles.

Dans les deux cas, une connexion internet sera nécessaire uniquement pour le téléchargement du jeu.

## 3 Game scenes Storyboard

### 3.1 Les directives du pompier Bonoeil

Vous êtes la nouvelle recrue de l'entreprise *Everything is Fine*, société d'aménagement d'établissements recevant du public pour le respect des normes incendie. Plutôt que de vous former durant une semaine dans une salle sombre de leur bâtiment, vos employeurs décident que vous apprendrez mieux sur le terrain. Pour veiller à ce que vous maîtrisiez bien les tenants et aboutissants des protocoles d'évacuation, vous serez accompagné du pompier Bonoeil, vétéran aguerri de son métier, qui pourra vous conseiller et vous épauler lors de vos dures épreuves.

Votre objectif : prouver que vous êtes un employé modèle en assurant au maximum la sécurité des personnes qui se retrouvent en situation d'évacuation.

Pour cela, vous devrez aménager correctement différents établissements avec les panneaux adéquats : issues de secours, plans du bâtiment, portes coupe-feux... et bien d'autres. Mais attention ! Votre entreprise vous alloue un budget limité qu'il faudra savoir gérer pour être efficace.

Le pompier Bonoeil sera également votre évaluateur : il veillera au bon respect des règles. À vous de l'impressionner pour prouver votre valeur !

### 3.2 Une première expérience

Pour bien illustrer le comportement d'une personne jouant à Everything is Fine, nous prendrons l'exemple de Liliane, joueuse imaginaire, qui va naviguer au travers des différentes scènes du jeu.

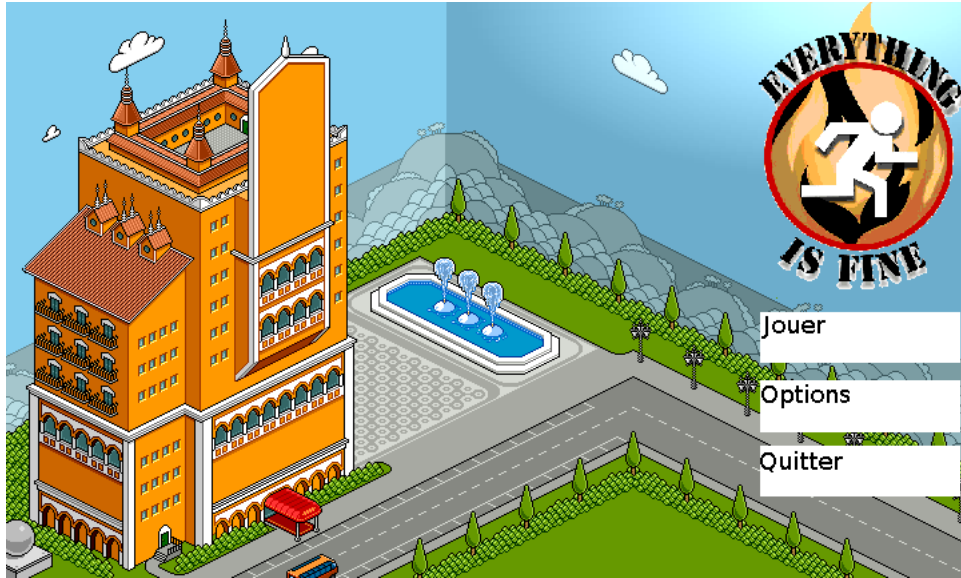


FIGURE 4 – Menu principal

Au premier plan du menu, Liliane distinguera rapidement les boutons lui permettant de naviguer dans le jeu :

- "Jouer", pour accéder à sa partie ;
- "Options" figure 5 ;
- "Quitter", pour quitter le jeu.

A l'arrière plan, elle aura un aperçu du jeu puisqu'il y figurera un niveau animé : les agents se déplaceront dans le bâtiment comme lors d'une véritable phase de simulation. Le logo sera également présent, il fera office de titre et d'identité visuelle du jeu.

Comme c'est la première fois que Liliane utilise le jeu, elle désire d'abord savoir ce qu'il est possible de modifier dans les options, et clique donc sur le boutons associé. On change alors de scène.

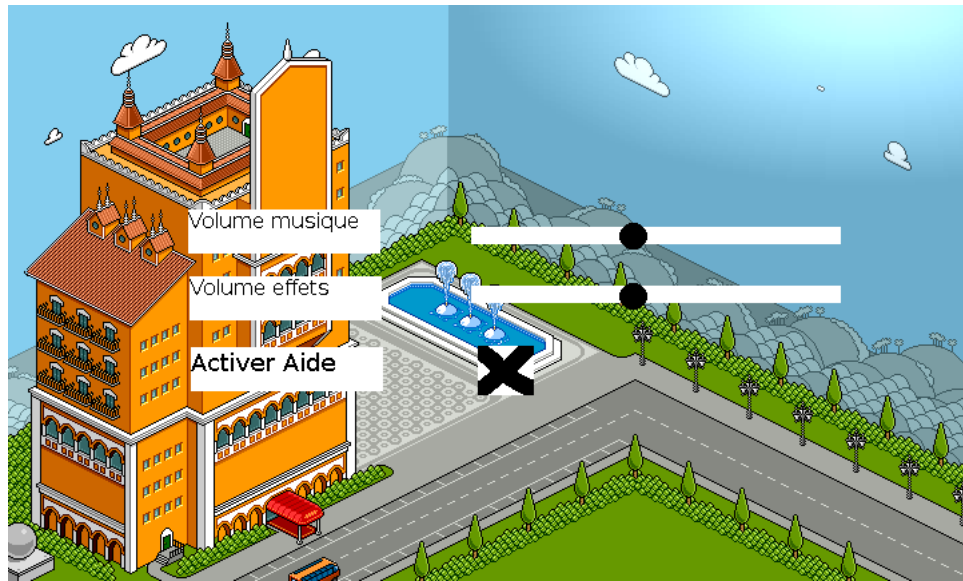


FIGURE 5 – Les options

Liliane prend ici conscience qu'il n'y a que trois aspects modifiables :

- La musique, qui sert l'ambiance du jeu et peut être réglée au bon vouloir de l'utilisateur voire même coupée ;
- Les bruitages, feedbacks des actions du joueur et de l'état des agents ;
- L'aide en jeu, qui, si elle est activée, se proposera sous forme du pompier Bonoeil afin de prodiguer des conseils sur le placement des objets à disposition du joueur.

Elle décide de diminuer la musique qui lui paraît un peu forte, et de désactiver les bruitages en se disant qu'elle pourra tout de même bénéficier des informations visuelles de ses actions. Comme c'est sa première partie, elle active également l'aide afin d'avoir accès à l'avis d'un expert.

Une fois tout cela réglé, elle clique donc sur le bouton "Retour", qui la ramène au menu.

Revenue à l'écran principal, Liliane veut se lancer dans l'aventure ! Elle sélectionne alors "Jouer" et accède à la sélection des niveaux.

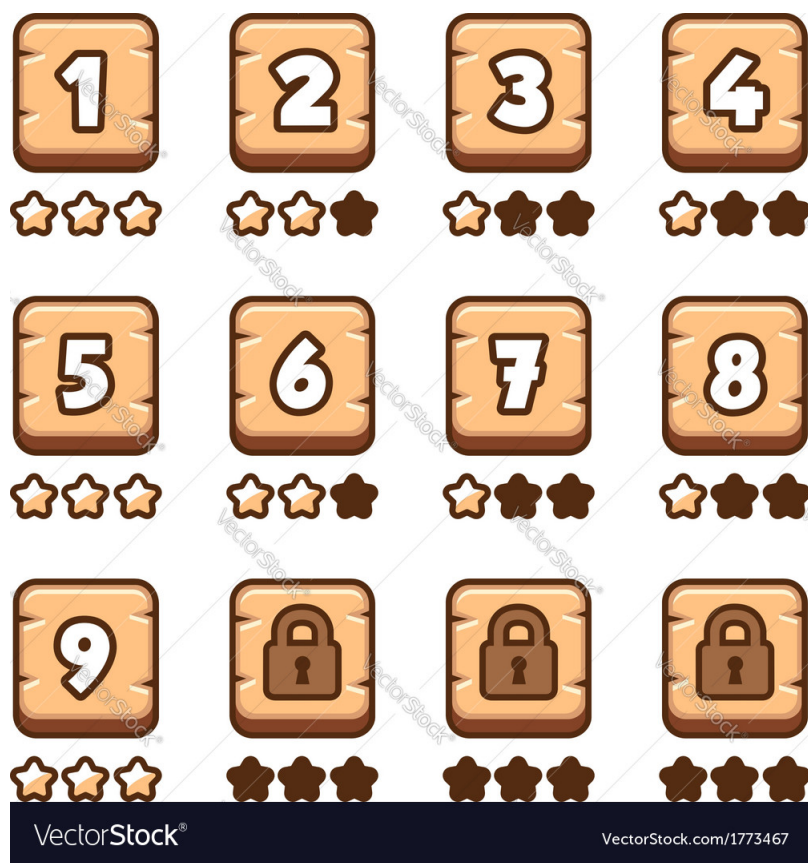


FIGURE 6 – Sélection des niveaux

Puisque c'est la première qu'elle joue, Liliane se verra proposé le niveau de tutoriel où elle aura tout le loisir de se familiariser avec les différentes mécaniques du jeu et comprendre le déroulement d'un niveau. Elle a également le choix de décliner car il est possible qu'elle connaisse déjà les bases. Comme ce n'est pas le cas, elle accepte de suivre ces explications. Elle sera guidée tout au long de sa première expérience par le pompier Bonoeil.

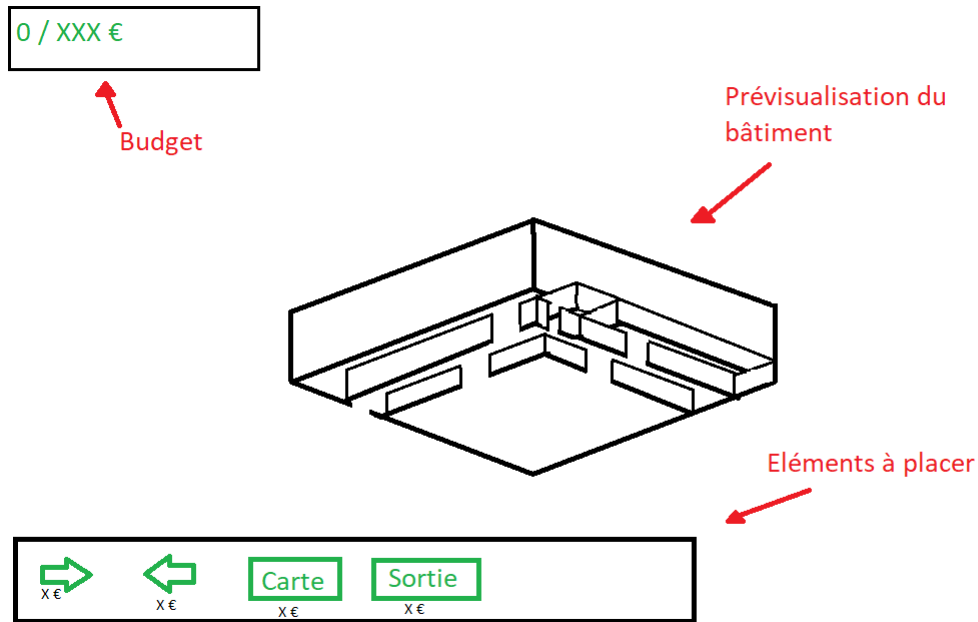


FIGURE 7 – Tutoriel

Liliane prend alors connaissance des différents éléments de l'interface :

- Le budget qu'elle devra gérer ;
- Les différents éléments qu'elle pourra placer sur le bâtiment ;
- Le bâtiment lui-même.

Une fois qu'elle pense avoir bien respecté les normes, Liliane lance la simulation. Bien que dans cette phase elle soit passive, elle peut tout de même suivre les agents se déplacer et interagir avec les éléments qu'elle a réparti sur tout le bâtiment. Elle surveille également le compteur car le temps est important lors de l'évacuation : plus vite les personnages seront sortis, plus vite ils seront en sécurité. Lorsque tous les agents sont sortis ou que les pompiers arrivent pour maîtriser l'incendie, il est temps pour Liliane de savoir si elle a bien répondu aux exigences de Bonoeil. Celui-ci la note donc sur les critères suivants :

- Le budget : il ne doit pas être dépassé ;
- Le nombre de personnes en sécurité à la fin de la simulation ;
- Le temps d'évacuation.



FIGURE 8 – Tableau des scores

Pour chaque condition respectée, Liliane obtiendra une étoile et il lui en faut donc trois pour réussir entièrement l'épreuve.

Elle peut recommencer si elle le désire ou retourner à la sélection de niveaux. Ayant obtenu la meilleure note du premier coup, elle se sent confiante et laisse derrière elle son succès du tutoriel pour commencer le jeu à proprement dit. Après quelques niveaux supplémentaires complétés avec brio, Liliane décide de faire une pause. De l'écran de sélection elle passe donc à celui du menu principal et sélectionne le bouton "Quitter". Sa progression est sauvegardée automatiquement et revenir à un niveau déjà complété ne lui fait pas perdre son score si elle obtient une moins bonne note. Elle a donc tout le loisir de montrer Everything is Fine à ses amis sans risquer de modifier sa progression.



## 4 Game Distribution

Que ce soit pour une distribution PC ou une distribution mobile, nous désirons que notre jeu soit accessible à tous. Pour cela, nous avons décidé de fournir directement l'exécutable des versions. Dans un souci de continuité, nous retrouverons donc le jeu décliné pour Mac, Windows et Android sur notre dépôt Github, où nous avons développé le projet durant toute la durée du premier semestre.

## 5 Gameplay

Le gameplay tournera principalement autour de l'aménagement des bâtiments grâce à l'ajout d'objet, à la manière des Sims.

### 5.1 Acteurs

#### 5.1.1 Personnages

Le principal personnage avec lequel le joueur pourra interagir est le pompier Bonoeil, à qui il pourra demander des conseils ou en bénéficier ponctuellement s'il active l'aide.

Il aura également la possibilité de sélectionner un employé pour le former aux protocoles d'évacuation.

#### 5.1.2 Objets

Les principales actions du joueur se dérouleront autour de la manipulation d'objets, puisque celui-ci devra équiper les bâtiments des différents niveaux. Il pourra ainsi créer et déposer :

- des panneaux de signalisation ;
- des portes coupe-feu ;
- des points de rassemblement ;
- des sorties de secours ;
- des plan d'évacuation.

Le joueur va pouvoir interagir principalement avec les bâtiments de de la scène en y ajoutant (à l'instar des Sims) des objets, en améliorant la qualité des matériaux. Ce sont ces interactions qui constitueront la majeure partie du gameplay et qui permettront de résoudre le niveau.

### 5.2 Actions

**Atteindre** les objectifs fixés par le pompier Bonoeil :

- Sauver un maximum de personnes ;
- Respecter le budget ;
- Limiter le temps d'évacuation.

**Déplacer** la caméra grâce à la souris et les flèches directionnelles, la signalétique sur le bâtiment.

**Choisir** quels objets seront intégrés à l'établissement et où. Choisir également des employés pour les former aux protocoles d'évacuation.

**Gérer** le budget pour ne pas le dépasser et si possible l'optimiser pour obtenir une étoile supplémentaire.

**Détruire** les éléments préalablement placés et qui ne sont plus désirables.

**Créer** les-dits éléments pour les installer dans le niveau.

**Demander** de l'aide au pompier Bonoeil.

## **6 Game interface**

### **6.1 Camera**

Le joueur a besoin d'une vue d'ensemble du bâtiment qu'il doit équiper. Il pourra visualiser un étage à la fois grâce à une vue isométrique qui lui permettra alors d'accéder notamment aux murs pour y intégrer des panneaux indiquant les issues de secours, par exemple. Pour naviguer entre les niveaux, il lui suffira d'utiliser les flèches directionnelles et la caméra s'adaptera au nouvel étage à visualiser. L'angle de vue peut également être modifier afin de faire apparaître des endroits non accessibles au-paravent.

### **6.2 Controls**

Le jeu se veut simple et intuitif pour que le joueur puisse se concentrer sur son objectif : sauver des vies, virtuelles certes, mais vies tout de même. Lors de la phase de construction, le placement des objets sera représenté visuellement par un cliqué/déposé.

En ce qui concerne le déplacement de la caméra, comme dit plus haut, il s'effectuera grâce au contact de la souris avec les bords de l'écran ou à l'aide des flèches directionnelles.

## 6.3 HUD

### 6.3.1 La phase de construction

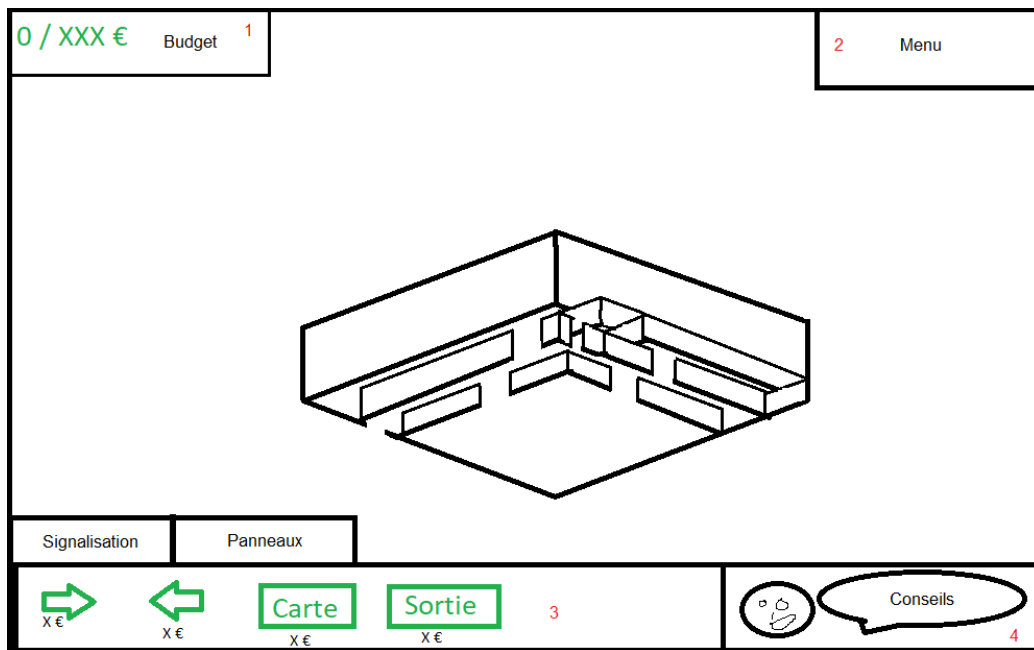


FIGURE 9 – Interface utilisateur en phase de construction

- (1) Budget : celui qui est alloué et celui qui est dépensé ;
- (2) Retour à la sélection des niveaux, passer à la phase suivante, quitter le jeu ;
- (3) Différentes catégories et les objets à placer correspondants ;
- (4) Pompier Bonoeil prodiguant ses inestimables conseils avec parcimonie.

L'interface est répartie sur les côtés pour ne pas gêner le joueur et lui permettre d'interagir facilement avec le décor qu'il doit aménager. Elle évolue donc à la fin de la phase de construction pour mieux s'adapter.

### 6.3.2 La phase de simulation

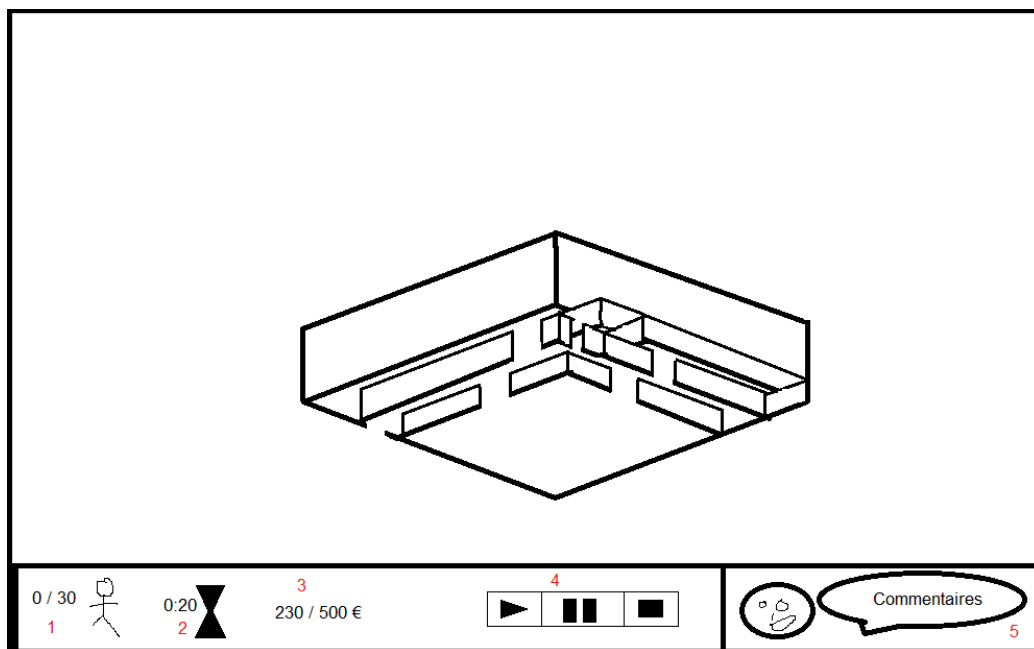


FIGURE 10 – Interface utilisateur en phase de simulation

- (1) Nombre de personnes en sécurité et le nombre total de personnes dans la simulation ;
- (2) Temps écoulé durant l'évacuation ;
- (3) Rappel du budget dépensé et celui total alloué ;
- (4) Lancement, pause, arrêt de la simulation ;
- (5) Pompier Bonoeil émettant des commentaires sur ce qui se passe à l'écran.

Cette interface réduite permet à l'utilisateur d'avoir en vue les données concernant la simulation tout en l'observant directement. Même s'il ne peut pas agir sur les agents, il est capable d'utiliser la caméra pour se déplacer et obtenir l'angle de vue le plus propice pour profiter au mieux de l'action en cours. Il peut également manipuler l'écoulement du temps, voire arrêter totalement cette phase s'il la considère comme caduque. Pendant ce temps, le pompier Bonoeil gratifiera le joueur de son avis sur la situation.

## 6.4 Feedback

### Déplacer la caméra

- Lorsque la souris s'approche d'un des bords de l'écran, elle se déplace et le joueur voit alors le monde "bouger". S'il utilise une des flèches directionnelles, l'angle de la caméra change et le joueur peut alors observer le monde d'un point de vue différent.
- Si le bord du monde est atteint, la caméra ne se déplace plus, même si le joueur essaye de déplacer le curseur. L'action principale se déroule dans le bâtiment, il n'y a donc pas de nécessité d'aller plus loin. Un feedback auditif n'est pas utile dans ce cas puisque le joueur sait qu'il doit se limiter à l'établissement qu'il doit équiper.

### Choisir des objets ou des employés à former

- Le joueur peut sélectionner l'objet de son choix à intégrer au bâtiment. Il peut également former un employé aux protocoles d'évacuation. Dans tous les cas un simple clic suffit. L'icône correspondante apparaît alors comme "enfoncée". La distinction de l'action se fera lors de la **création**.
- Aucun son particulier n'est à prévoir pour cette action, l'aspect visuel se retrouve déjà dans plusieurs autres jeux et est considéré comme "classique", cela pourrait donc surcharger inutilement l'attention du joueur.

### Gestion du budget

- Lors d'un achat, la somme en question est débitée du solde total et "s'envole" au-dessus du montant restant tout en disparaissant peu à peu afin de laisser une trace visuelle de son action au joueur.
- Lorsque le budget est dépensé, aucun bruit ne se déclenche pour ne pas faire doublon avec le bruit de construction de la création d'objets.

### Dépassement du budget

- Si le budget est dépassé, il s'affiche désormais en rouge pour signifier au joueur qu'il n'a pas respecté cette contrainte.
- Un son de cloche se déclenche alors pour alerter le joueur.

### Destruction d'un élément dans le bâtiment

- Si le joueur décide de supprimer un élément qu'il a préalablement placé, celui-ci est alors enlevé du bâtiment.
- Un son sec et grave se fait entendre pour signifier une action négative mais pas pénalisante.

### Objets à placer dans le bâtiment

- L'objet suit le curseur de la souris et se positionne avec un halo vert sur le mur/sol si c'est un endroit valide.
- Un son de construction de validation retentit si un objet est placé et sera pris en compte lors de la simulation.

#### **Formation d'un employé**

- La formation est symbolisée par un gilet jaune. Le joueur peut le sélectionner comme un objet classique, à la seule différence qu'il doit le placer sur un employé. Tous les employés formés porteront alors un gilet jaune pour signifier au joueur qu'ils auront un comportement plus adapté à l'évacuation.
- Un agent, enjoué par sa nouvelle formation, émettra un "wahou" pour montrer au joueur que son action a été réalisée.

#### **L'aide du pompier**

- Comme vu sur la figure 9, les conseils de Bonoeil apparaîtront en bas à droite. Le texte s'affiche lorsque le joueur le décide.
- Le pompier s'exprimera à la manière des personnages de Banjo & Kazooie.

#### **La simulation**

Bien que le joueur n'effectue aucune action dans la diégèse lors de cette phase, il bénéficiera tout de même de feedbacks pour comprendre la scène qui se déroule devant lui.

- Tous les agents vont se déplacer dans le bâtiment. Chacun possèdera un halo de couleur selon ses émotions : colère, panique...
- Ils émettront également des sons pour partager leurs sentiments : cris, pleurs...

## **6.5 Audio, music, sound effects**

## **6.6 Help system**

Comme cela a été vu sur lors du storyboard, c'est le joueur qui décide dans les menus du jeu s'il active ou pas l'aide du pompier Bonoeil. Le système d'aide est donc explicite.



## 7 Game Mechanics

### 7.1 Game OCR

#### Réussir un niveau

- Objectif : Respecter les trois contraintes suivantes :
  - Ne pas dépasser le budget ;
  - Sauver toutes les personnes présentes dans le bâtiment ;
  - Minimiser le temps d'évacuation.
- Challenge : Comprendre le fonctionnement des objets à disposition du joueur.
- Récompense : Une étoile par contrainte respectée.

Puisque c'est un jeu de sensibilisation, il est normal que la boucle principale repose sur les notions que l'ont souhaite inculquer au joueur.

#### Ne pas dépasser le budget

- Objectif : Dépenser une somme moins importante que le budget alloué.
- Challenge : Optimiser le nombre d'objets et leur placement.
- Récompense : Une étoile et la satisfaction de ne pas être "passé dans le rouge".

Le rouge étant perçu comme négatif, le joueur aura tendance à ne pas vouloir dépenser plus que ce qu'il lui est donné et sera donc réjoui s'il y arrive.

#### Sauver toutes les personnes présentes

- Objectif : Permettre à tout le monde de pouvoir évacuer.
- Challenge : Repérer les différents types de personnes et anticiper leur comportement pour faciliter leur évacuation. Par exemple : signaler l'ascenseur pour une personne handicapée.
- Récompense : Une étoile et la satisfaction de ne pas avoir laissé mourir des gens.

Cela peut avoir l'effet inverse, comme dit plus haut, et le joueur, même s'il n'obtiendra pas d'étoiles, peut s'amuser à voir les agents bloqués dans le bâtiment.

#### Évacuer dans les temps

- Objectif : Terminer l'évacuation avant le temps légal.
- Challenge : Prendre connaissance du niveau pour identifier le meilleur chemin à suivre.

— Récompense : Une étoile et la satisfaction d’avoir respecté une contrainte. Cette boucle est peut être la plus dure à accomplir puisqu’elle n’est pas prévisible. Le joueur ne sait pas combien de temps vont prendre les agents pour évacuer. C’est cependant une donnée importante car elle entre en jeu lors des exercices d’évacuation effectués dans les ERP selon leur superficie.

### **Placer un objet**

- Objectif : Créer un objet et le placer dans le bâtiment.
- Challenge : Savoir où placer l’objet.
- Récompense : L’objet sera pris en compte lors de la simulation.

Même si cette boucle ne contient pas de challenge à proprement parler (ce ne doit pas être une épreuve pour le joueur), il est tout de même nécessaire de la préciser puisque c’est celle qui reviendra le plus souvent dans le cycle du jeu.

## **7.2 Game Rules**

En ce qui concerne les éléments à placer, le joueur n’a pas besoin de gérer une quelconque physique. Les objets se positionnent correctement automatiquement (sur les murs ou sur le sol) car cela ne revêt pas un aspect pertinent pour le jeu.

Pour la phase de simulation, les agents possèdent un corps tangible et ne peuvent donc pas se traverser, de même pour les murs ou les portes. S’ils passent à travers une fenêtre ou se font renverser par une voiture, ils tomberont à terre. Lorsque le feu propage, s’il touche un agent, celui-ci s’enflamme.

## **7.3 Game Flow**

Il n’y a pas de réelle continuité spatiale dans le jeu : les différents niveaux ne sont pas liés.

L’évolution dans le temps peut être caractérisée par la difficulté croissante, synonyme de la progression du joueur dans l’entreprise.

## **7.4 Game AI**

Basés sur les travaux de Jéméry PATRIX, les agents auront un comportement sur plusieurs niveaux détaillés dans la prochaine partie. Leur principale

fonction sera de s'adapter à la signalétique mise en place par le joueur afin de mesurer la justesse de ses choix.

## 7.5 Replaying and saving

La rejouabilité se trouve d'une part dans l'obtention des étoiles : si le joueur ne les a pas eues du premier coup, il devra alors recommencer le niveau. Comme évoqué précédemment dans le *Look and Feel*, la dualité de l'expérience peut inciter à équiper un bâtiment d'une toute nouvelle manière. La sauvegarde est automatique à chaque fin de niveau et un score moins bon que le précédent n'annule pas celui-ci.

## 8 Game Model

### 8.1 AI

Le comportement des agents repose sur le modèle Belief - Desire - Intention (figure 11).

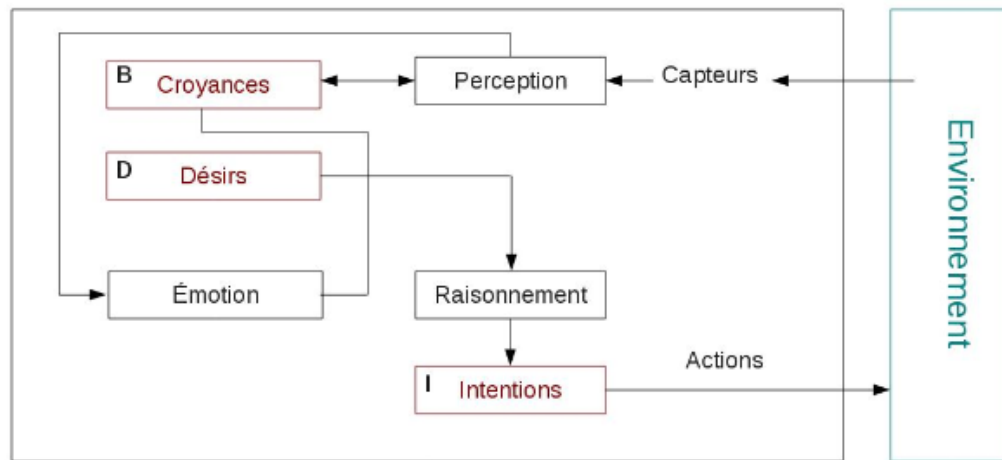


FIGURE 11 – Modèle BDI

- **Perception** : Tout ce que l'agent voit de son environnement (autres agents, murs, flammes, panneaux) ;
- **Croyances** : Ce que l'agent *pense* savoir du monde, ces données ne sont pas vraies ou fausses tant qu'elles ne sont pas vérifiées ;
- **Désirs** : Les actions ou états du monde que l'agent souhaite voir réalisés ;
- **Émotions** : Le caractère de l'agent et ses sentiments vis à vis de ce qui l'entoure (notamment la peur à la vue des flammes dans notre cas) ;
- **Raisonnement** : Au vu de tout ce qui a été présenté, l'agent va prendre une décision de ce qui est le mieux à faire pour lui ;
- **Intentions** : Le plan à suivre pour accomplir l'action prise lors de l'étape précédente.

Est donc modélisé ici le niveau micro de la simulation.

Le niveau meso, lui, est représenté par du flocking (figure 12) entre les agents d'un même groupe de connaissance (des amis venus visiter l'ERP ensemble, une famille, ect...)

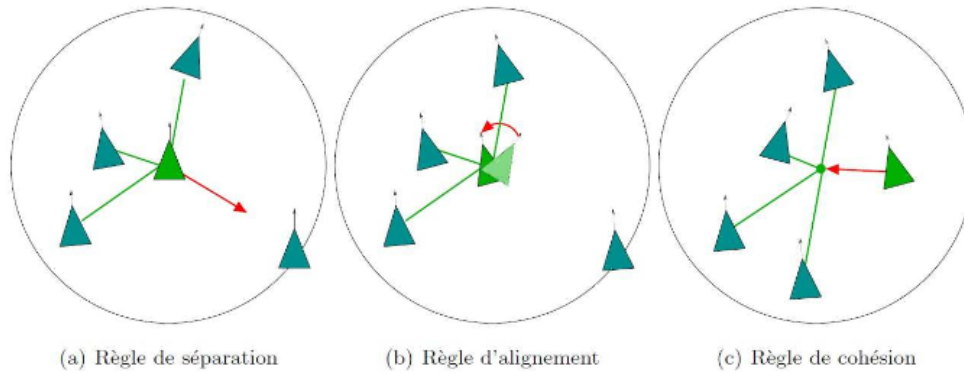


FIGURE 12 – Les 3 règles du flocking

Tout comme le modèle BDI, le flocking est un ensemble de règles, et plus précisément dans ce cas, de vecteurs, au nombre de 3 :

- **Separate** : Chaque agent ayant une zone de confort qu'il souhaite respectée, il décidera de s'éloigner des agents qui sont trop près de lui. Ce vecteur est la moyenne de tous les vecteurs qui pointent la direction opposée à chaque agent proche.
- **Align** : De la même manière, sauf en cas de fort engagement dans sa tâche, un agent ne se déplacera pas à contre courant. Ce vecteur représente la moyenne des directions suivies par les autres agents proches afin qu'il puisse suivre le flot de son entourage.
- **Cohere** : Le contraire de **Separate**, si un agent se sent trop éloigné de ses pairs il aura tendance à se rapprocher. Ce vecteur est la somme des vecteurs pointant chaque agent dont on souhaite se rapprocher.

La moyenne de ceux-ci, après pondération de chacun par des coefficients, modifiera le comportement de l'agent pour qu'il ai tendance à suivre le mouvement ambiant.

## 8.2 Interface

Chaque élément cliquable de l'interface est un objet bouton d'Unity. Pour ensuite attribuer une action à un bouton il faut lui ajouter un script (ici le langage est C#).

Chaque bouton de l'interface permettant d'ajouter un objet sur le bâtiment possède en paramètre un prefab de l'objet auquel il correspond. Pour ajouter des éléments sur le bâtiment l'utilisateur va cliquer sur le bouton dans l'interface qui correspond à l'objet qu'il faut ajouter. Lors du clic sur ce bouton un booléen est mis à "true", celui-ci indique que le joueur a cliqué sur le bouton. Ensuite on vérifie à chaque frame (fonction "update()" d'Unity) si un clic gauche est effectué par l'utilisateur. Si c'est le cas et que le booléen du script est à "true", alors il faut poser un objet à l'endroit où l'utilisateur a cliqué. Pour ce faire le script du bouton va ensuite appeler un objet "GameManager" qui possède les fonctions permettant de s'occuper de l'ajout de l'objet sur le bâtiment.

Pour ajouter un objet sur le bâtiment le GameManager va utiliser des "RayCast". Il va tirer un rayon de la souris dans la direction de la caméra. Il va ensuite récupérer la position du premier mur qu'il touche ainsi que sa normale pour poser l'objet à la bonne position et dans la bonne rotation. Une fois que l'objet est posé le GameManager s'occupe de modifier le budget ainsi que son affichage à l'écran.

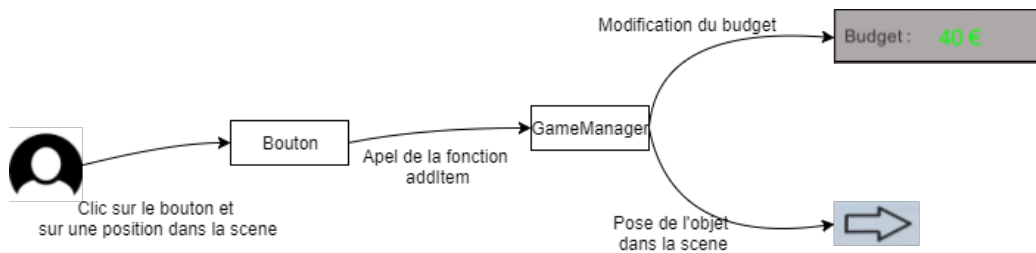


FIGURE 13 – Fonctionnement de l'ajout d'éléments

Pour la suppression d'éléments dans la scène on fait un peu le fonctionnement inverse sauf que l'on ne passe pas par les boutons. Le GameManager va vérifier à chaque frame si l'utilisateur effectue un clic droit, si c'est le cas il va alors tirer un rayon dans la direction de la souris. Si ce rayon touche un objet placé par le joueur alors il va le supprimer et modifier le budget et son affichage.

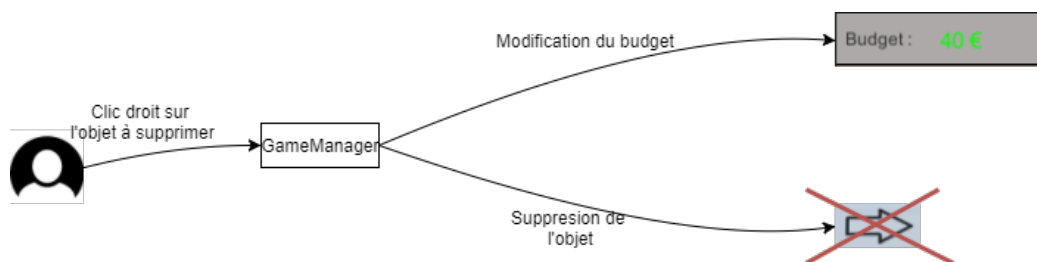


FIGURE 14 – Fonctionnement de la suppression d'éléments

## 9 Game Story

Pensé à la manière des jeux mobiles, comme Cut the Rope par exemple, il existe un léger contexte pour justifier les actions demandées au joueur sans pour autant que celles-ci impactent une réelle histoire.

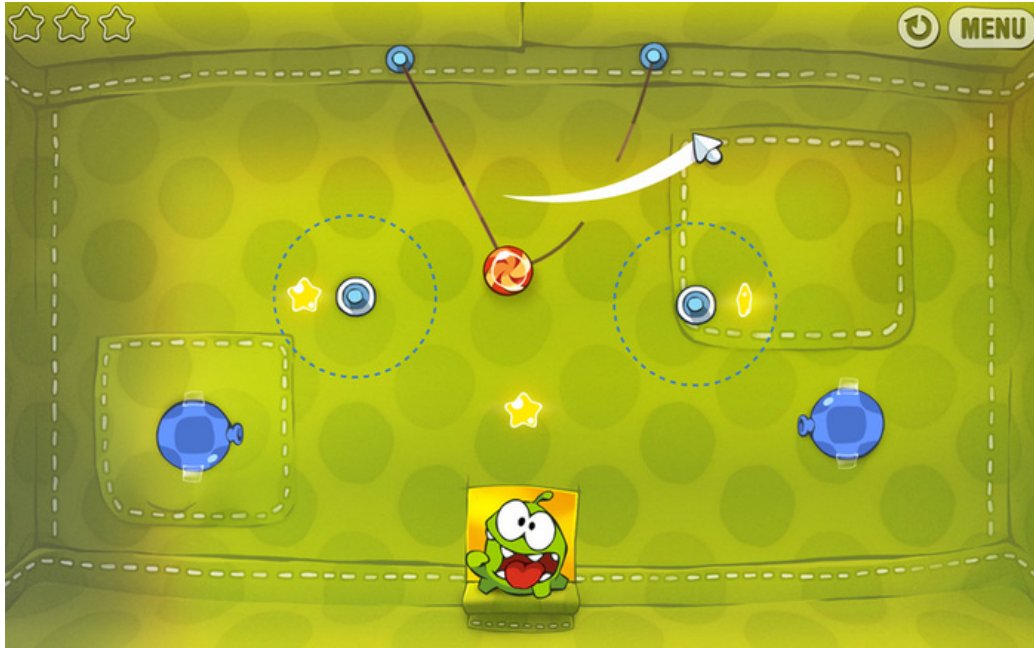


FIGURE 15 – Un niveau de Cut the rope

Dans ce jeu (fig 15), le but est de nourrir le petit monstre vert à chaque niveau, sans pour autant faire avancer une quelconque narration.

### 9.1 Story and narrative

Lors du tutoriel, le pompier Bonoeil détaillera le contexte au travers des instructions qu'il va donner au joueur. Cela justifiera alors les éléments de gameplay qui seront ancrés dans la réalité du jeu.

### 9.2 Game world

L'univers de Everything is Fine est sensiblement le même que le nôtre. Il n'existe pas de règles physiques particulières. Le changement majeur réside



dans les personnes qui peuplent ce monde parallèle : ils sont un peu simple d'esprit et ne savent pas trop prendre des initiatives : ils peuvent très bien se trouver à côté de la sortie d'un bâtiment et pourtant suivre de mauvaises indications qui les feraient alors s'éloigner de leur but premier !

### **9.3 Characters traits**

Bien qu'il existe plusieurs types d'agent (enfants, adultes), et qu'ils possèdent des comportements et caractères propres, le seul personnage avec qui le joueur pourra interagir, c'est le pompier Bonoeil.

Fort de ses années d'expérience en tant que pompier, Bonoeil a décidé de mettre son savoir-faire au service de l'entreprise Everything is Fine afin de former les nouveaux arrivants. Lorsque le joueur commence une partie, il fait donc sa rencontre puisqu'il va l'encadrer lors de sa première mission et tout au long du jeu. Ce personnage servira de conseiller et d'évaluateur.

### **9.4 Game levels**

Chaque niveau représente un ERP, il peut donc s'agir de musées, de lycées ou encore de restaurants. Le principe est sensiblement le même pour tous : plusieurs pièces et divers couloirs, comment réussir à indiquer à tout le monde le meilleur chemin à suivre ?

## 10 Game Aesthetics

Les décors du jeu (bâtiments, arbres, voitures...) sont représentés en "low-poly", il ne possèdent donc pas beaucoup de détails, mais permettent tout de même de se repérer et d'identifier rapidement leur fonction.

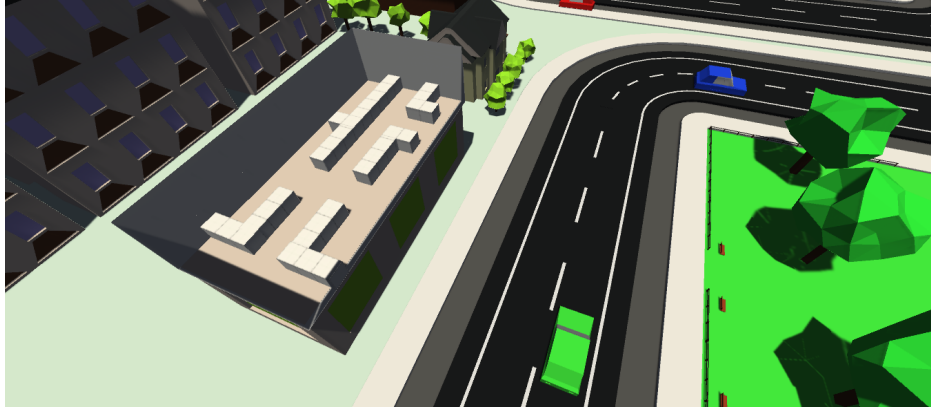


FIGURE 16 – Apparence lowpoly du jeu

Les agents quant à eux sont modélisés de manière simple avec des voxels. De même que pour le reste de l'univers, le joueur peut déterminer d'un coup d'œil si l'ERP dont il a la charge est occupé par des employés, des visiteurs adultes, des enfants ou encore des personnes âgées. Cela lui donne des informations complémentaires sur leur comportement puisqu'ils n'agissent pas tous de la même façon. De plus, cela permet également de minimiser la tension éprouvée par le joueur puisque l'absence de côté réaliste n'accuse pas celui-ci d'avoir laissé mourir de "vraies" personnes.

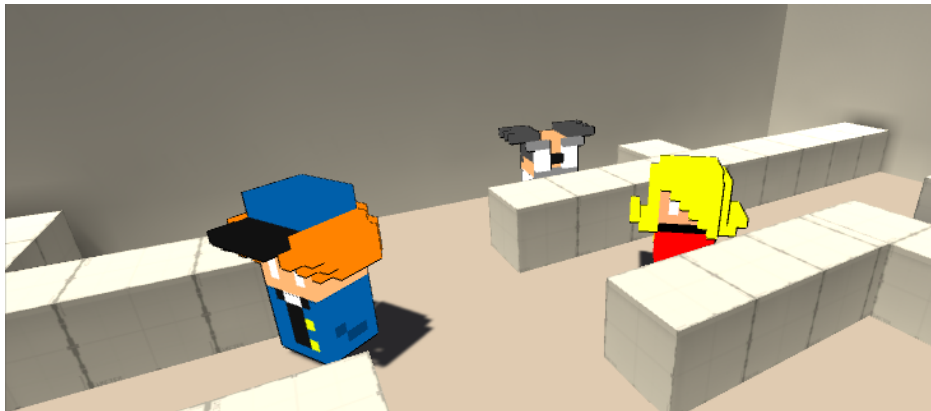


FIGURE 17 – Apparence des agents