Cahier des Charges

# Fonctionnalités de base, à implémenter

-          Représentation 3D d’une structure, plus ou moins grande (pièce, étage, appartement, partie d’entreprise, etc…)

-          Navigation de l’utilisateur à l’intérieur de la structure (se déplacer, se téléporter à un autre endroit via raccourcis, monter escaliers/ascenseurs etc)

-          Interactions simples avec objets : le clic donne des informations sur l’objet

-          Le bâtiment est structuré, chaque objet a un certain nombre d’informations.

-          Création/modification/suppression d’objets appartenant à cette base de données du bâtiment

-          La création d’objets se fait via un importeur. Import d’objets déjà présents dans la bibliothèque d’objets, ou objets personnalisés.

# Fonctionnalités secondaires

-          Edition de structure : tracé d’un plan 2D, construction automatique d’une structure 3D à partir de ce plan

-     L’utilisateur peut déplacer les objets à sa convenance

# Déplacement dans la structure

- Utilisation du clavier et de la souris

- haut/bas/droite/gauche -> zsdq/fleches

- trackball souris pour les rotations caméra, avec point/viseur qui symbolise là où se porte le regard de l’utilisateur.

- utilisation de boutons ou d’une mini-map ou des 2 pour accéder à d’autres pièces (téléport)

# Modes de vue

- FPS (mode immersif):  pour se déplacer et obtenir des informations sur les objets

- aménagement: en vue de dessus pour l’édition des objets et le déplacement

- Construction: mode plan de construction

Switch entre les différents modes : middle clic affiche une roue, avec les différents modes de vue autour. Déplacement de la souris sur l’un de ces modes (avec middle clic toujours enfoncé) switche vers ce mode. Relâchement du middle-clic sans avoir déplacé la souris : annule l’action (et reste sur le mode de vue actuel)

Possibilité de changer de mode de vue via icones/boutons.

# Interactions

- clic gauche sur un objet pour obtenir des informations

- clic droit sur un objet pour accéder à une liste d’actions possibles …

- barre espace pour accéder à un menu d’action (accès au catalogue des modèles…)

## Interactions en sélection d’objet

/\*

2 modes de vue, disponibles à tout moment:

- FPS

- mode carte vue de dessus de l’étage courant

- (possibilité d’afficher plusieurs étages en vue plan dans la même fenêtre ??)

Différents modes de vues, non accessibles directement

- highlight de tous les objets d’un type (vue éloignée, murs et autres objets avec une certaine transparence)

- déplacement d’objets (vue différente que la vue FPS ?), soit dans le cadre d’une édition d’objet déjà existant, soit dans le cadre d’une preview d’objet non acheté.

\*/

# Fenêtre des informations “de base”

- titre de la fourniture/objets tel qu’il apparaît dans le catalogue du magasin

- fournisseur

- numéro d’identification unique du matériel dans la base de donnée de l’entreprise

- petite image preview du modèle

- quantité de ce modèle dans l’entreprise

- date d’achat

- un bouton pour faire une nouvelle commande de ce type de modèle

# Fenêtre des informations “étendues”

- fournisseur

- date d’achat

- type d’achat ( investissement immédiat, location avec/sans  option d’achat)

- date de fin de contrat (jamais ou une date..)

- type d’amortissement

- amortissement courant

- prix unitaire (déduit du prix de groupe le cas échéant)

# Bases de données

Accessibles depuis l’application Unity.

Diverses possibilités :

* Utilisation de SQLite. Désavantage : compliqué avec C#, plus adapté au JS.
* Utilisation de Linq
* Utilisation de fichiers XML (solution adaptée dans le cas d’un standalone)

Pour l’utilisation de fichiers XML :

* Au chargement de l’application, on ouvre fichiers xml, et désérialisation pour remplir nos objets (List<Objet>).
* Au moment de sauvegarder, sérialisation des objets voulus et remplissage des fichiers xml
* Requêtes : créer ses propres fonctions de recherches sur nos collections (List<>), ou utiliser Linq.

Modèle 3D

* Description
* Objet 3D (.fbx ou autre)
* Aperçu photo

Pièce

Objet

Bâtiment

* Id
* Description

Contient (1…n)

Contient (0…n)

GameObject

Est représenté par (1…1)

Chaque GameObject sur la scène devra être clairement lié à un objet dans notre BDD. De sorte que si on veut mettre en highlight tous les objets de type « alarme », alors une requête sera utile.