

SUPPORTS

Nous allons utiliser 2 logiciels libres, multi plate-formes, mais qui sont aussi utilisables en ligne directement depuis le site web :

- ➔ SweetHome3D <http://www.sweethome3d.com/fr/>
- ➔ Scratch2 <http://scratch.mit.edu>

Ainsi pour la domotique gratuite, il est possible de faire créer aux élèves leur propre environnement et ainsi leur demander d'imaginer leur propre scénario de domotique. L'intérêt est de pouvoir **solliciter l'imaginaire individuel, personnaliser les degrés de difficulté et d'ainsi motiver/impliquer davantage** grâce à l'utilisation de ses propres créations.

Cahier des charges simple : créer votre maison idéale.

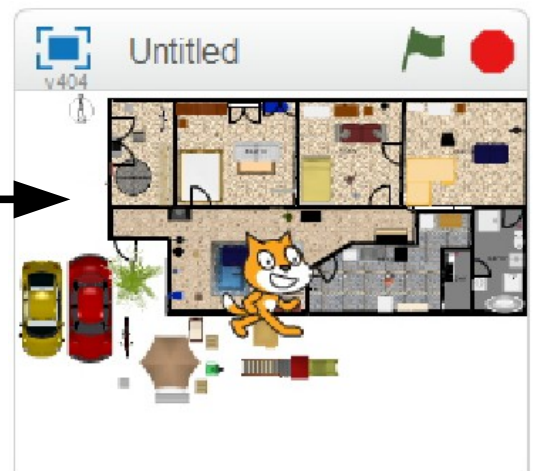
Exemple réalisé en autonomie par une élève de 5ème SEGPA



DÉMARCHE VISÉE

Nous allons utiliser cet habitat dans un programme qui déclenchera une alarme quand un personnage rentre dans une pièce, ce qui nous intéresse est donc la vue de dessus :

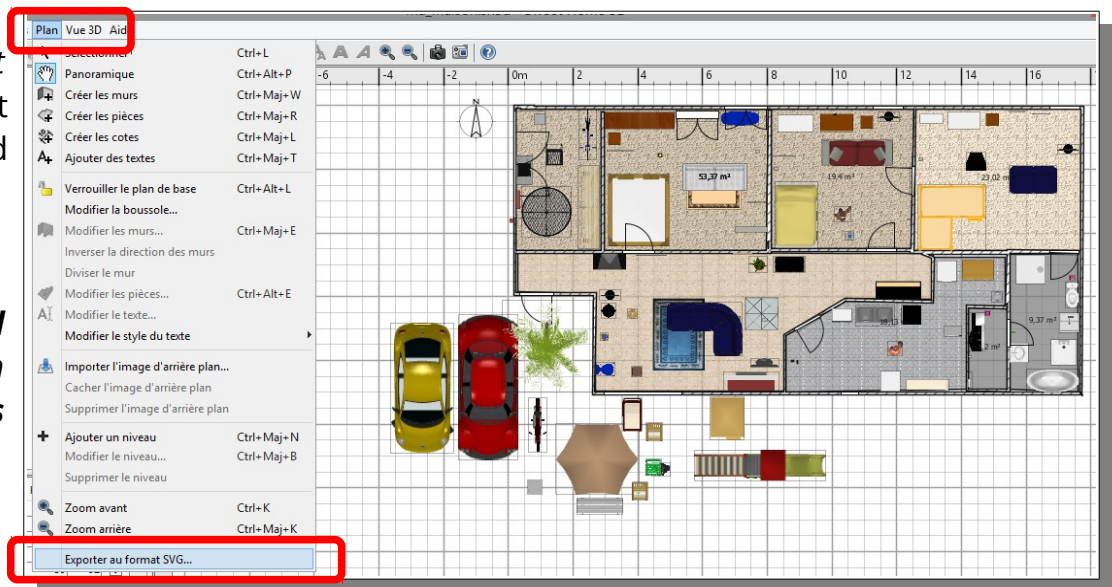
le plan.



IMPORTER UN PLAN SWEETHOME3D DANS SCRATCH

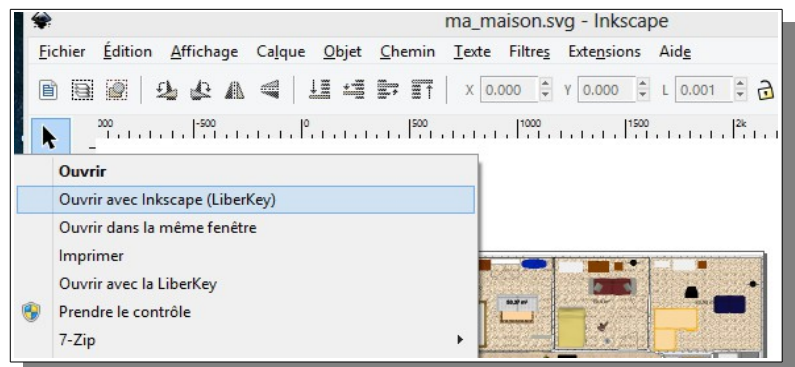
Scratch2, contrairement à Scratch1, permet d'utiliser le standard d'image vectoriel SVG.

Rappel : ce standard d'image tolère le zoom à l'infini sans pixellisation.

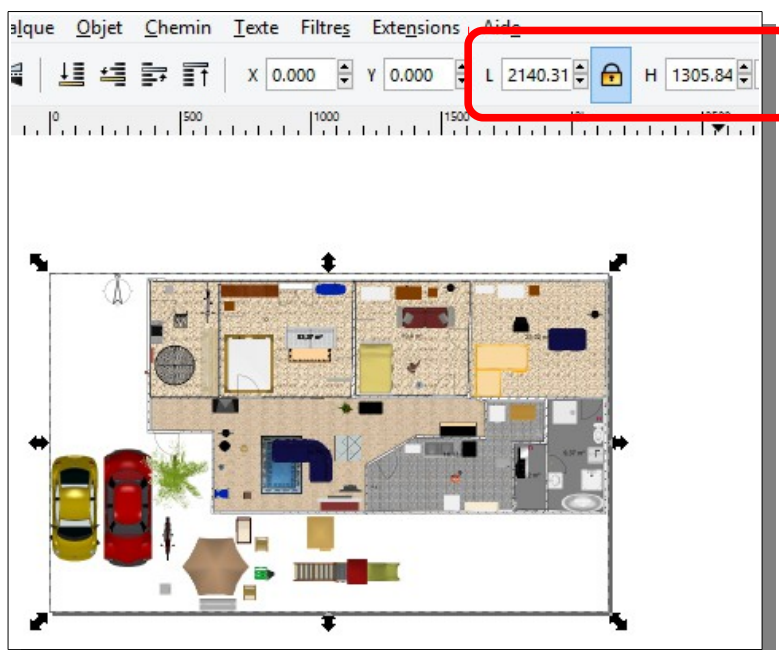


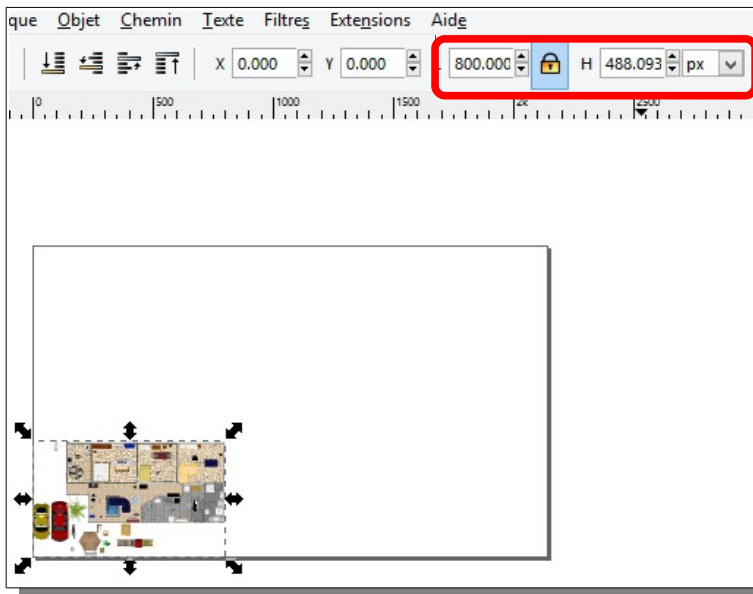
Si vous essayez d'importer directement ce plan dans Scratch2, vous vous apercevrez très vite du problème : *la taille* !

Pour s'en apercevoir il suffit d'ouvrir le fichier exporté par SweetHome3D dans un logiciel qui est capable de traiter le SVG : **Inkscape** (si vous avez l'EduClé cela ne vous posera aucun problème avec un simple clic droit sur le fichier SVG →).



Vous pouvez voir les dimensions en pixels en haut à droite de l'écran :

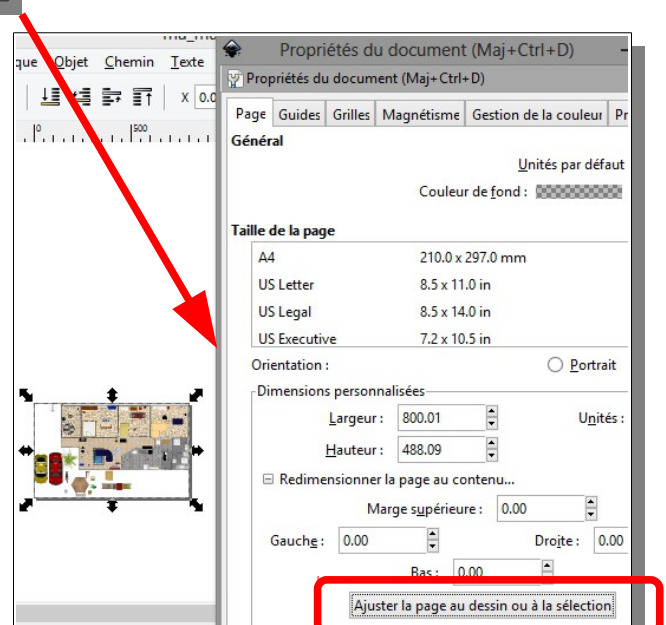
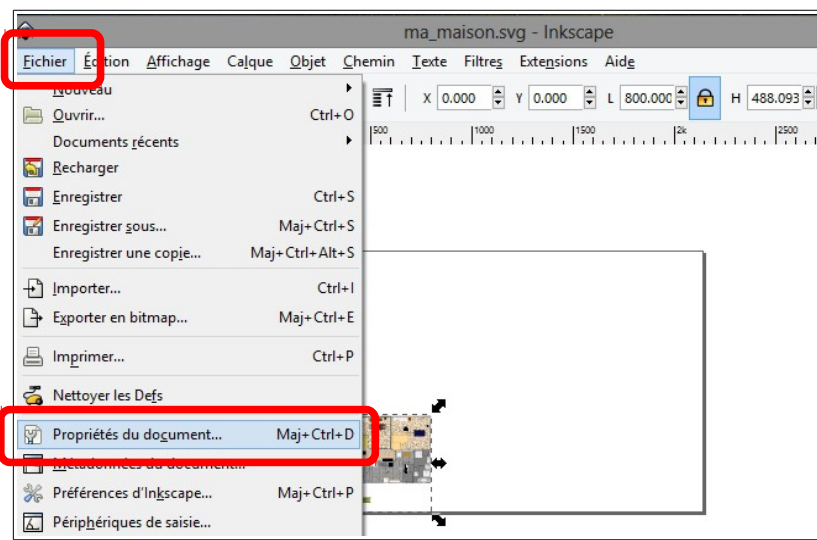




Il suffit d'imposer la plus grande des deux dimensions comme ne dépassant pas 800 pixels, ou moins (*rappel : ça ne changera rien lors des zoom : on fonctionne en vectoriel !*), en prenant soin de **verrouiller le ratio Longueur/Hauteur** grâce à l'icône du cadenas.

La zone de travail par défaut dans Scratch est de **480x360 pixels**.

Mais la zone de travail va prendre trop de place dans Scratch, donc il faut elle aussi la redimensionner :



Nous disposons donc d'un plan que nous allons pouvoir importer en fond d'écran de notre logiciel **Scratch**.

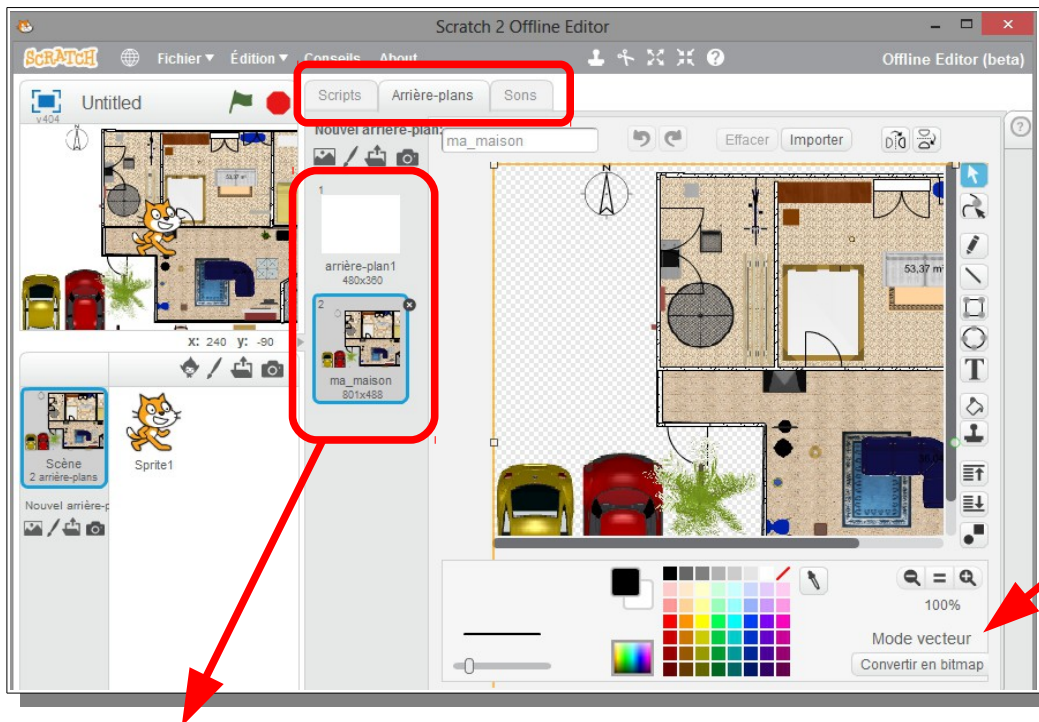
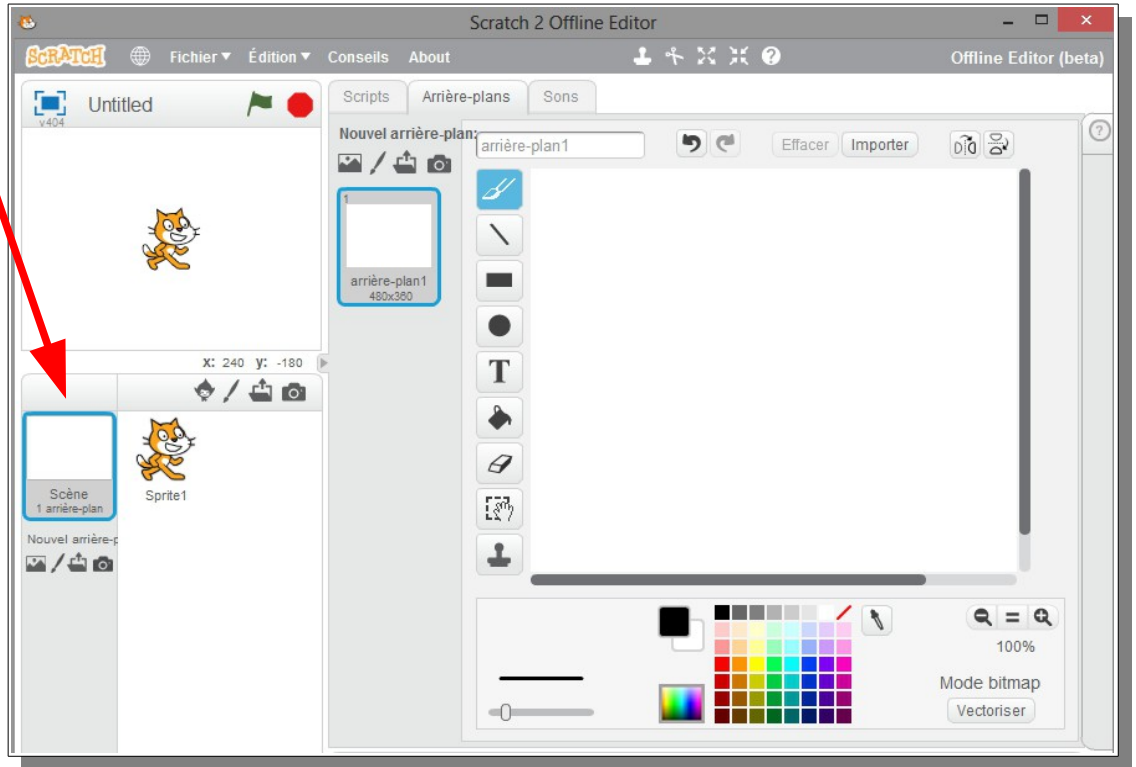


Pour cela rien de plus simple : vous cliquez sur l'arrière plan dans la liste des éléments à gauche,

puis vous choisissez « Importer depuis un fichier » :



Suivant les choix de dimensions fixées dans **Inkscape**, votre image peut être plus grande que la zone par défaut :

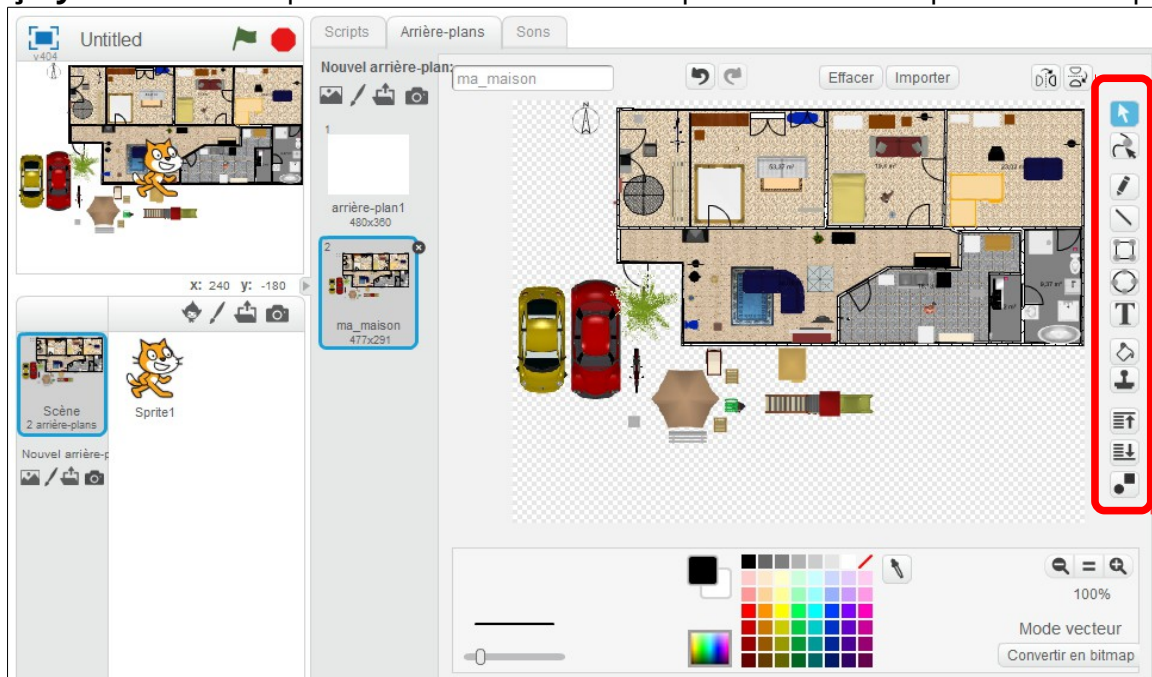


Il suffit simplement de la redimensionner en prenant soin de rester en « Mode vecteur ».

On peut voir qu'on dispose maintenant de 2 arrières plans, vous pouvez supprimer celui qui ne vous intéresse pas par un simple 'clic-droit → supprimer'.

L'intérêt est de pouvoir changer d'arrière plan à n'importe quel moment du programme : changer de pièce, changer de lieu, le même lieu de 2 couleurs différentes pour simuler quand la lumière est allumée ou non, etc.

Ça y est ! Nous disposons de la zone dans laquelle Scratch va pouvoir se déplacer :



Mais il faudra aussi réduire le personnage pour le mettre à l'échelle :



sélectionner le personnage puis redimensionner dans la zone de droite, dans l'onglet 'Costumes'.

Comme on peut le voir ici, il sera possible de disposer de **scripts** (programmes) spécifiques/différents pour chaque « Lutin » ainsi que pour l'arrière plan.

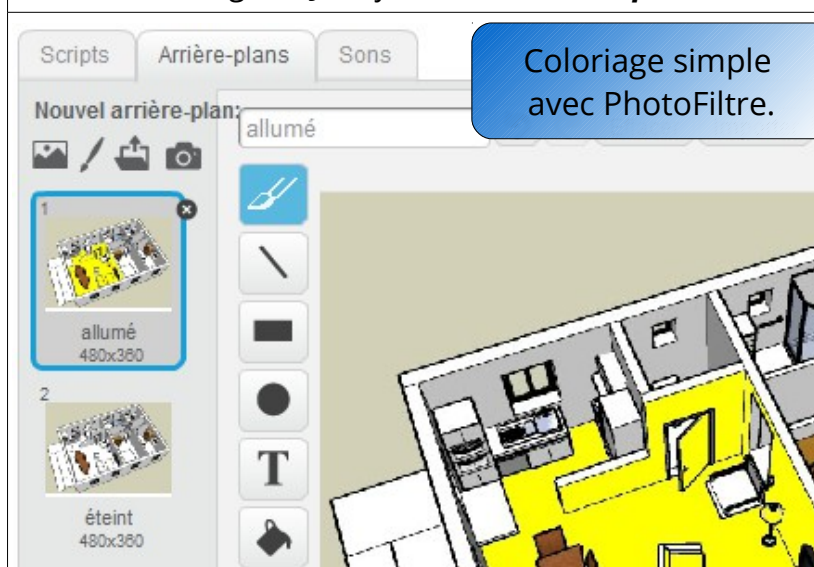
→ chaque script peut s'exécuter en **parallèle** aux autres, de façon **autonome** mais aussi **interagir avec les autres** comme nous le verrons par la suite.

Scripts Costumes Sons

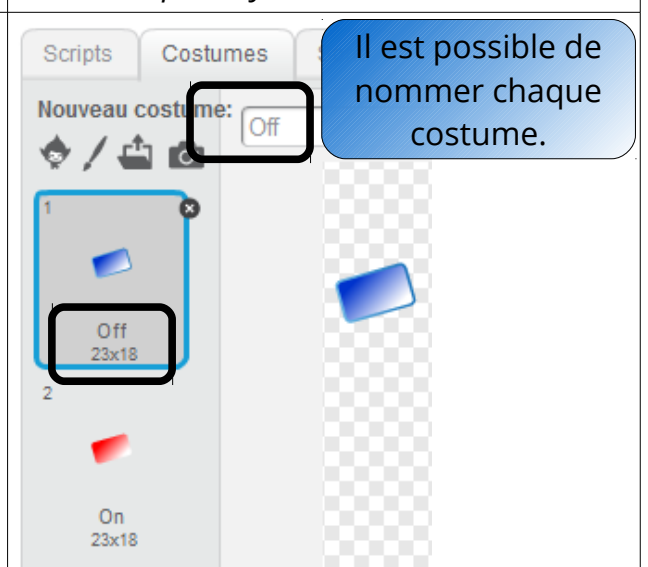
PROGRAMMONS UN ÉVÉNEMENT

Nous allons profiter des 'costumes' pour éclairer une pièce quand on appuie sur un interrupteur.

Exemple avec une capture d'un logement, une simple image en JPG, fait dans **SketchUp** :



On prépare un interrupteur avec 2 couleurs pour symboliser l'état :



On démarre toujours un programme en initialisant l'environnement.

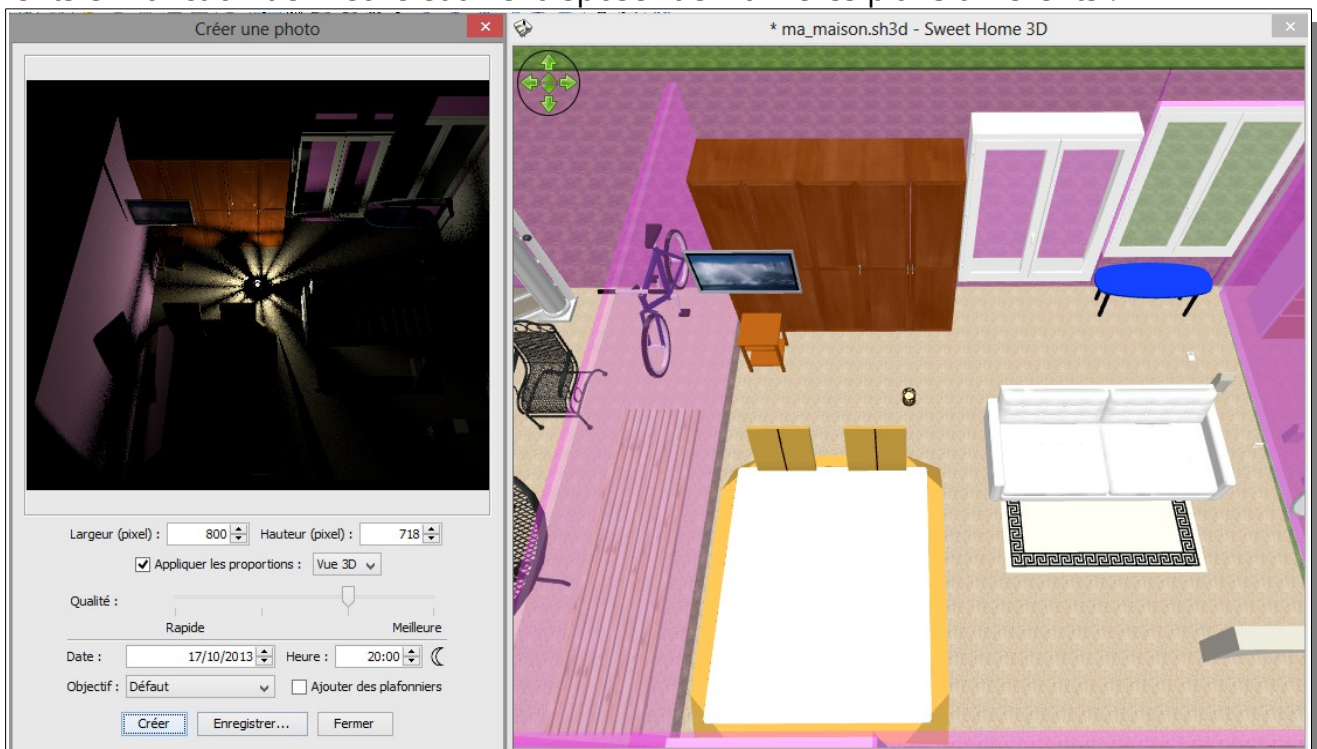


Ensuite on teste si la souris est sur l'interrupteur **ET** si on clique :

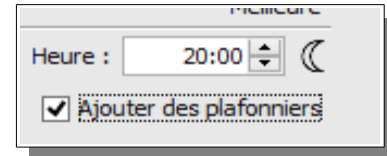
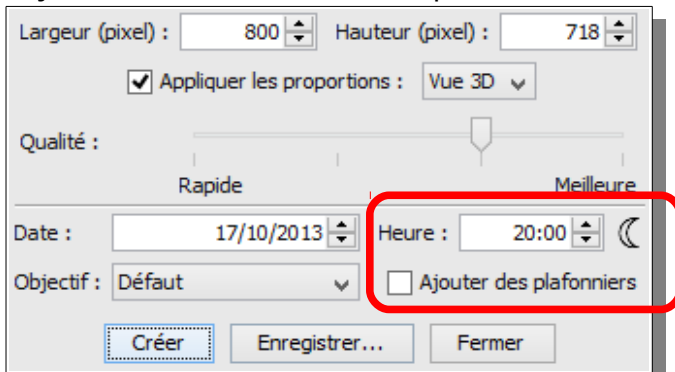
En effet le programme est celui du 'Lutin' de l'interrupteur (Button 3) donc si le pointeur de souris est touché c'est parce que la souris est sur ce 'Lutin'.

BONUS : SPRITES GRÂCE À SWEETHOME3D

Prenons les photos d'un environnement, en qualité 3/4 (ou 4/4), afin de bénéficier d'éclairages différents en fonction de l'heure et ainsi disposer de 2 arrières plans différents :

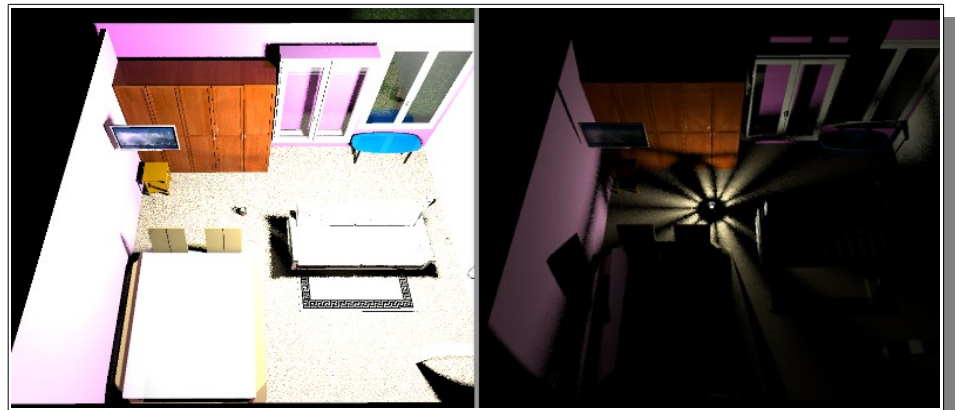


On peut ainsi introduire les économies d'énergies en testant le rendu à des heures différentes de la journée (heures où classiquement on se trouve dans la pièce concernée), puis faire un **rendu de nuit** pour notre exemple sur l'éclairage et un rendu **avec** l'éclairage :

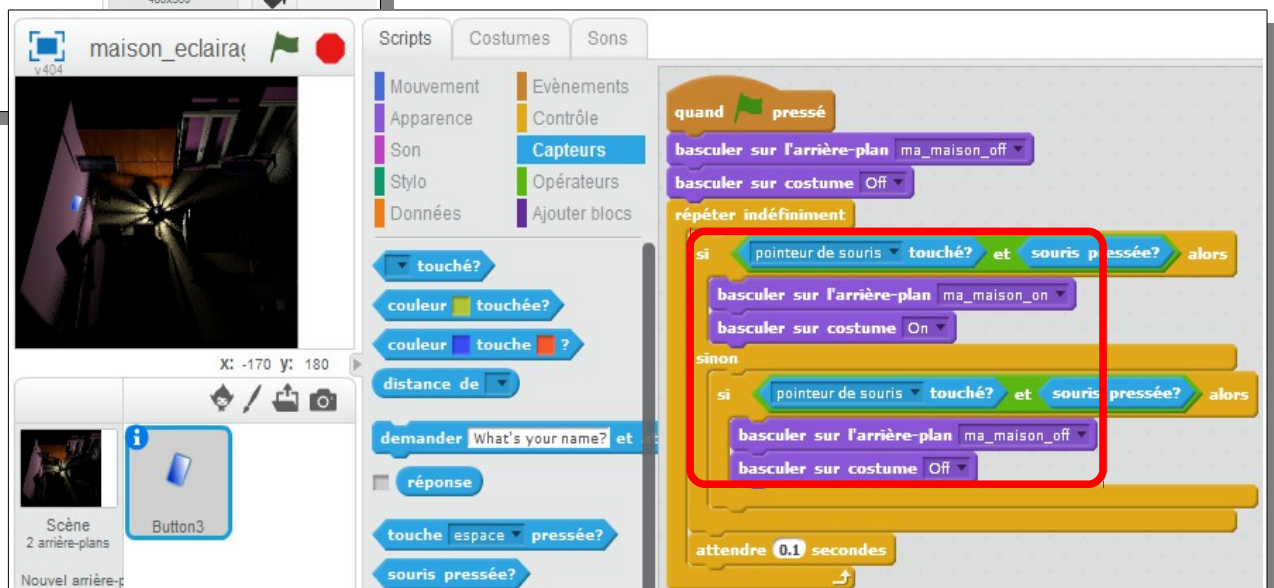
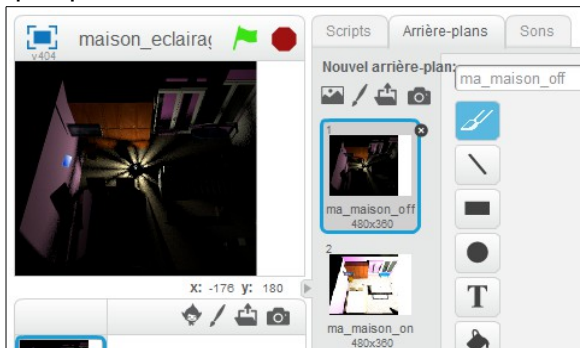


Attention ! Ça met du temps en calculs.

On obtient nos 2 arrières plans :

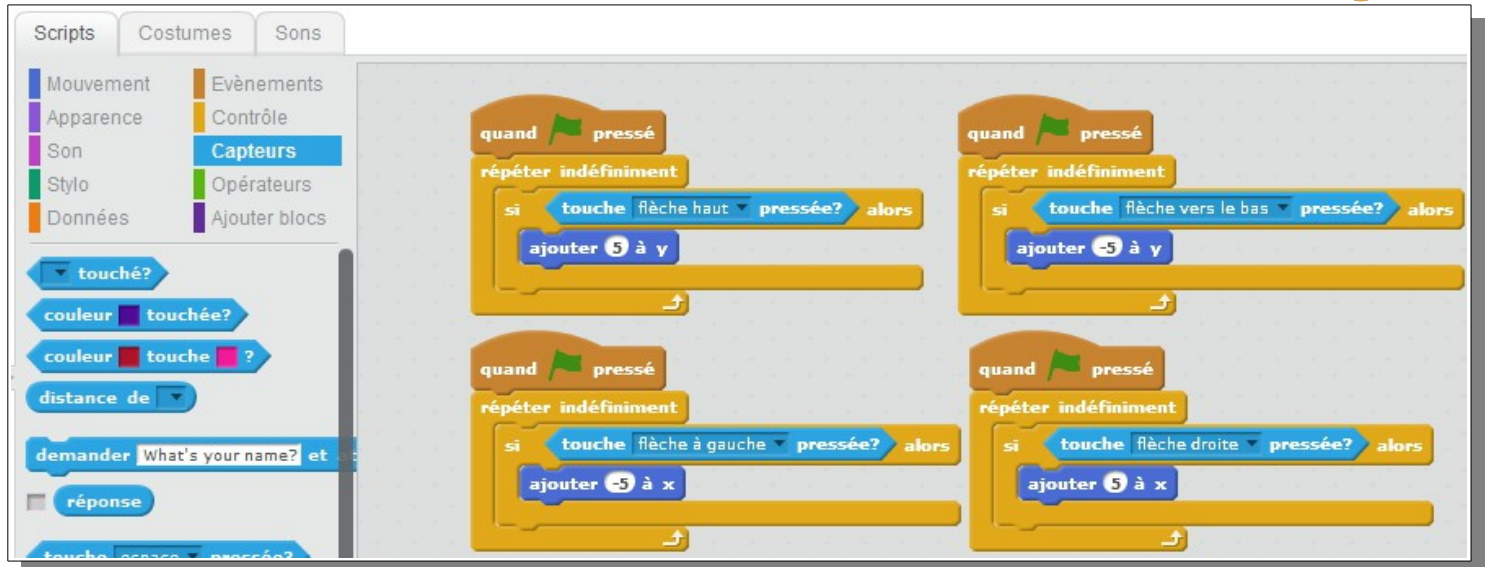


On les utilise dans Scratch avec la même technique d'interrupteur que précédemment :



AUTOMATISONS UN ÉVÉNEMENT

Nous n'allons pas simplement diriger le personnage à l'aide des flèches, c'est trop simple 😊



Toutefois ce qui est à retenir de cet exemple :

- tout se fait dans l'onglet 'Scripts',
- chaque 'Sprite', ou 'Lutin', ou 'Arrière plan', bénéficie de sa propre zone de 'Scripts',
- le programme ne modifie la position (x, y) *que du personnage pour lequel on est dans l'onglet 'Scripts'* : donc pas de conflit entre les positions des personnages (pas de x_{sprite1} , x_{sprite2} , etc),
- on peut avoir des programmes qui s'exécutent en parallèle et indépendamment,
- on choisit de surveiller en *parallèle* les 4 touches de direction afin de bénéficier des déplacements en diagonale : on peut **à la fois** modifier x **ET** y car il n'y a pas à attendre une itération complète, ce qui est très différent du cas suivant
- tous les programmes sont coordonnés au lancement du *top départ* symbolisé par le **drapeau vert global**, mais on peut lancer chaque programme indépendamment en cliquant sur **son** drapeau vert.
- de la même façon il sera possible d'utiliser des sons *spécifiques à chaque 'Lutin'*.



Le choix de l'architecture du programme qui répond au cahier des charges **sera propre à chacun**, à sa façon de penser, d'imaginer la décomposition du scénario. **La formulation de son idée par l'élève sera donc un point très important de sa capacité à s'exprimer : il faut lui demander d'explicitier pour l'aider à développer ses compétences langagières**

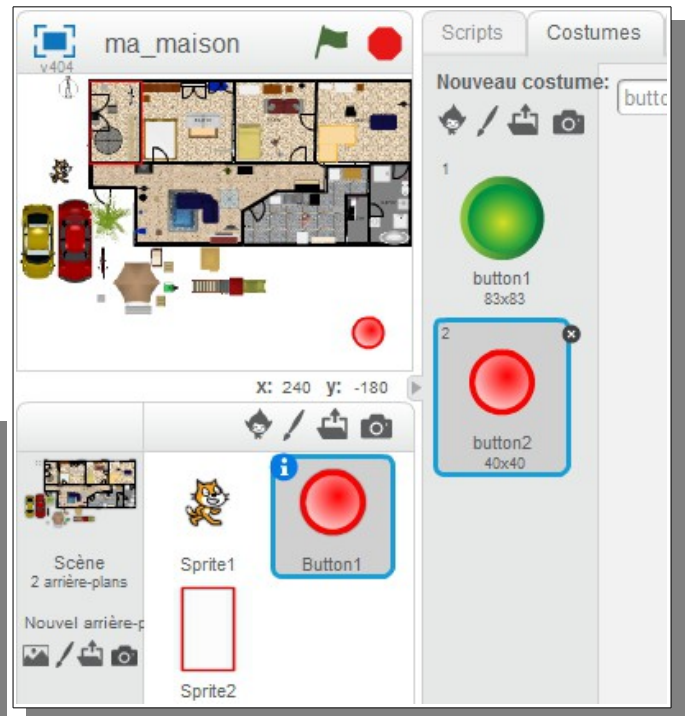
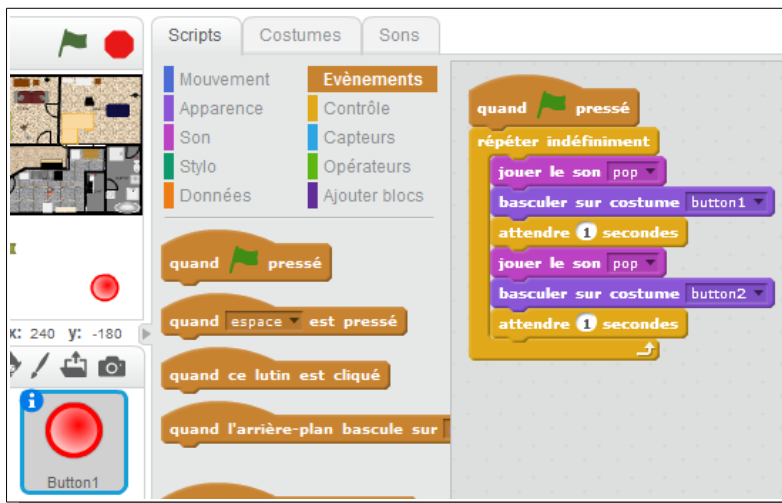
Rajoutons donc des 'Lutins' pour enrichir par des interactions !

Revenons à notre domotique à l'aide d'un scénario simple : quand le personnage rentre dans la pièce surveillée l'alarme est déclenchée.

Nous savons comment déplacer le personnage, mais il nous faut rajouter des 'Lutins', ou 'Sprites', supplémentaires pour symboliser les éléments de notre scénario :

- le 'Sprite 2' va définir la zone surveillée,
- le 'Button 1' symbolisera le signal d'alarme, c'est pourquoi **nous lui attribuons 2 costumes** afin de clignoter.

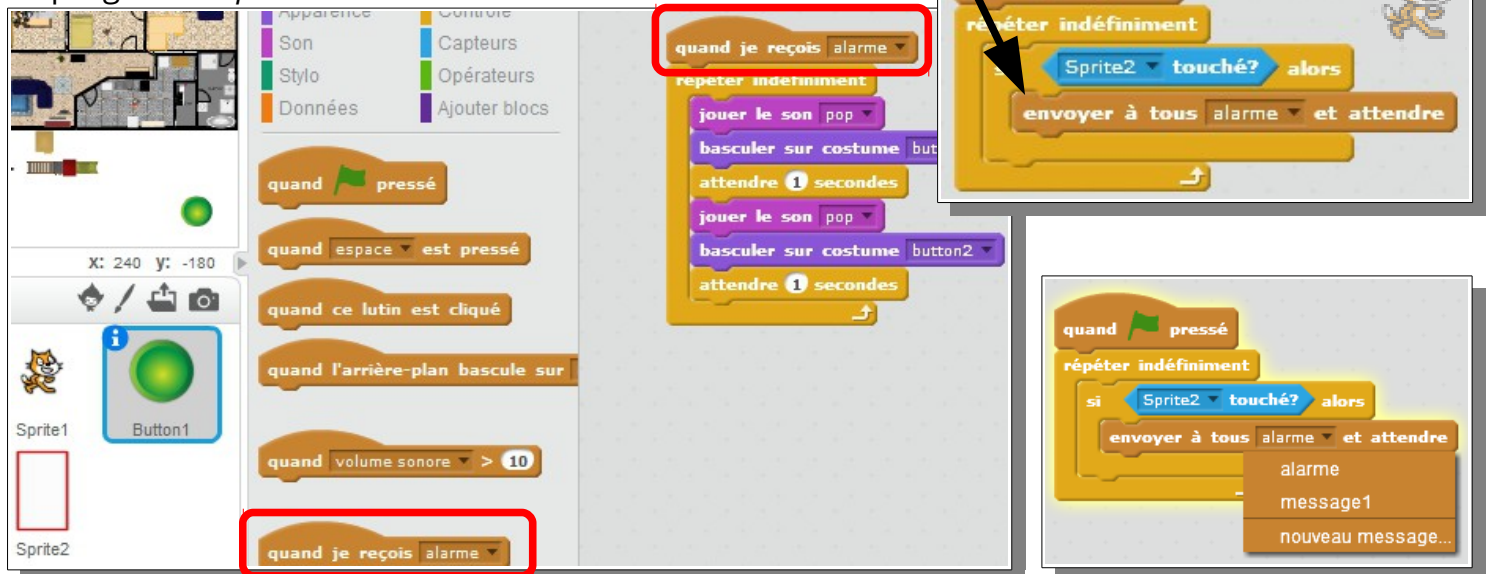
Voici donc l'exemple pour une alarme :



Problème à résoudre : l'alarme n'est pas déclenchée en même temps que le drapeau vert, mais si le 'Sprite 1' touche le 'Sprite 2'.

Hors nous avons vu que les programmes sont en parallèle, attribués au 'Lutin' pour lequel on crée ses 'Scripts', il faut donc trouver un moyen d'échanger des informations d'un programme à l'autre :

Le programme pour l'alarme sera alors :



Il est ainsi très facile de mettre en place un **travail collaboratif par équipe** en **répartissant** les programmes à faire dérouler : un élève s'occupe de l'alarme, un autre du déplacement, etc.

BONUS : INTERAGIR AVEC LA RÉALITÉ

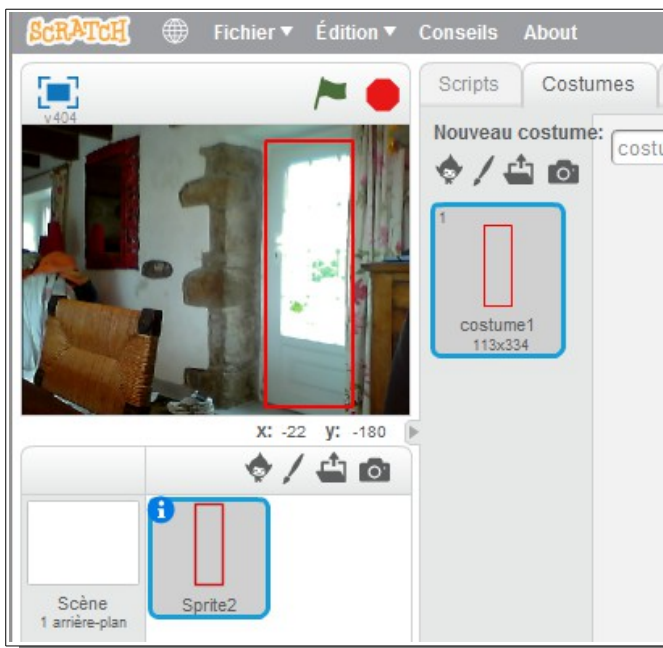
Nous savons créer des zones 'surveillées' et envoyer une information d'un programme à l'autre, mais ce que nous allons utiliser en lien avec le réel est la **webcam**.

Toujours sans interface supplémentaire, Scratch2 permet d'utiliser la webcam intégrée à un ordinateur (ou sur port USB).

Rappel : ce sont les capteurs associés au 'Lutin' pour lequel nous allons programmer un 'Script'.

Scénario simple de domotique : surveiller le franchissement d'une porte sous alarme.

On commence par initialiser la webcam :



Puis il faut tracer la zone réactive qui fera office de zone surveillée par notre système d'alarme virtuel :

Ensuite on programme une surveillance qui scrute indéfiniment s'il y a des mouvements dans cette zone (le 'Sprite 2' → 'ce lutin') :



Ici, le seuil est mis à 50 mais libre à vous de changer la tolérance de mouvements.

Le terme 'pointeur de souris' est ici générique.

Tout mis bout à bout, on obtient :



Libre à vous d'**imaginer toutes les combinaisons possibles** :

- vidéo-surveillance d'une salle en live ;
- plan dessiné/scanné/SweetHome/capture d'écran SketchUp/photo de maquette ou lieu réel ;
- déplacements programmés/contraints par le prof/ directement avec des touches ;
- déclenchement d'une **interface**.