

# Kids(playground) ; Le bac à sable numérique



## Rappel rapide du cahier des charges

Le but du projet est de réaliser un environnement numérique destiné à l'apprentissage de la programmation dès le plus jeune âge.

Le projet se base sur Scratch et l'utilisation de ses blocs, ainsi que sur Scratch Jr, l'application smartphone adaptée aux 5 – 7 ans.

Nous voulons donc reprendre le principe de Scratch Jr, l'adapter sur ordinateur via un site web et surtout rajouter des exercices dédiés pour l'apprentissage de chacun des blocs. En effet, même si Scratch Jr se dit adapté aux enfants, la seule différence avec Scratch est l'utilisation de couleurs plus vives et autres dessins attirant l'œil des enfants, mais au final il n'y a concrètement aucun contenu. L'enfant est lancé sur une page où il peut poser des blocs pour interagir, mais il ne dispose ni de tutoriel ni d'objectif réel.

## Rappel du planning

Nous avons pour objectif d'avoir pour décembre :

- la gestion des projets utilisateurs, c'est-à-dire avoir la possibilité de créer un nouveau projet et de le sauvegarder pour pouvoir revenir dessus plus tard, ainsi que la sauvegarde des exercices dédiés déjà résolus ;
- le développement des premiers blocs de base (déplacement dans chaque direction + un bloc condition ou boucle – à déterminer) ;
- une implémentation d'une partie des exercices dédiés (en mettant des "coquilles vides" aux endroits où des blocs n'auraient pas encore été développés).

Suite aux technologies choisies – que nous ne connaissons pas à l'avance – nous avons décidé au cours du semestre de revoir nos objectifs à la baisse. C'est-à-dire :

Mise en place de la machine virtuelle de travail et choix des outils et technologies qui seront utilisées afin d'avoir une réalisation homogène au sein du groupe, mais également la rédaction de la documentation de certains de ces outils par le chef de projet, afin que les autres membres puissent avancer plus rapidement lorsqu'ils en auront besoin.

Réalisation du back-end en Django (avec préalablement l'apprentissage de Django, que nous n'avions jamais utilisé).

Étude des différentes bibliothèques de blocs et réalisation des blocs de déplacement de base suivant la bibliothèque choisie.

Réalisation d'un template pour la structure des exercices et création des maquettes pour les exercices dédiés.

## Travail réalisé

### Gestion de projet

Une coordination au sein du projet a été faite afin de pouvoir voir l'avancée du projet et son évolution au fil du temps. Ces réunions étaient souvent bimensuelles ou bien hebdomadaires pendant les cours et le temps disponible. Elles ont également été complétées par un suivi de l'avancement du développement via GitHub.

### Squelette du projet

Suite à l'installation de la machine virtuelle, nous avons commencé une réalisation rapide du front-end design ; un template a été utilisé pour la page d'accueil. Il a été repris du site [html5up](https://html5up.net/twenty) et réadapté pour notre utilisation de Django (lien du site original : <https://html5up.net/twenty>). L'intérêt d'une telle approche réside dans le fait de pouvoir proposer un visuel de qualité et qui possède un rendu responsive, cross browser et bien souvent une bonne documentation concernant le code du site (HTML, CSS et JS). En effet, l'étude des techniques modernes de web design s'avèrent être de plus en plus poussées et il est très difficile de suivre de manière perpétuelle les dernières évolutions (en plus de rattraper le retard).

Une autre possibilité qui aurait pu être considérée lors de la création du squelette est l'utilisation de bibliothèques CSS telles que Bootstrap ou bien encore Material Design Lite. Cependant Bootstrap possède l'inconvénient d'agir comme un obfuscateur de code en ajoutant un trop gros nombre de classes CSS et d'éléments à la structure HTML d'origine. Il est ainsi très difficile de lire une structure de document HTML / CSS utilisant Bootstrap. De plus, Bootstrap et Material Design Lite possèdent un rendu « aseptisé » puisqu'ils sont très souvent utilisés et donc très reconnaissables. L'avantage des templates est qu'il en existe un grand nombre, ce qui les rend moins susceptibles d'être utilisés.

Du côté du back-end, comme dit précédemment, le framework Django a été utilisé en concordance avec Python 3 pour la partie web. Afin de pouvoir utiliser Python3 et Django de manière recommandée, l'utilisation d'un virtual environment est demandée. Ce dernier permet de choisir la version de Python (Python2.7 par défaut) que l'on veut utiliser. La gestion de Django - qui en réalité un simple module Python - et d'autres modules, tel que psycopg2 (module de connexion au SGBD Postgresql), s'effectue via le gestionnaire PIP qui la référence, ainsi que le gestionnaire de base du langage. Un grand avantage de Django est de bénéficier d'un certain nombre de scripts qui permettent une génération d'applications (sous partie indépendante d'un projet Django) ou encore de générer un projet type qui comporte toutes les briques essentielles.

Postgresql est ensuite utilisé, couplé à la partie administration de Django, pour l'authentification au site. Pour le moment, le module d'identification a été fait sans demande de création de compte.

Nous avons également prévu d'utiliser Nginx pour faire tourner le serveur, mais n'avons pas eu le temps de nous pencher longuement dessus pour l'implémenter. Néanmoins un compte peut être ajouté grâce à l'interface "admin" de Django, qui permet une édition de chacun des codes

Cette structure va encore évoluer au fur et à mesure de l'avancement du projet. Cette base a surtout été mise en place afin d'avoir une prévisualisation du rendu final. Elle nous permet de travailler sur les différentes parties du projet tout en pouvant observer le résultat dans un contexte concret. L'objectif principal étant de pouvoir réaliser chacune des évolutions de manière totalement incrémentale. Une autre notion importante est l'indépendance de chacune de ces parties qui doivent pouvoir être interchangeables, surtout dans ce contexte

d'incrémentalité. Par exemple, le SGBD doit pouvoir être remplacé par MySQL dans le cas où le serveur d'hébergement n'autorise pas PostgreSQL.

## Gestion des blocs

Nous avons utilisé le tutoriel du site Blockly (lien du site : <https://developers.google.com/blockly/guides/overview>), JQuery (utilisé pour simplifier la gestion du DOM) et JQueryUI (utilisé ici principalement pour gérer les objets "draggable").

Les blocks implémentés pour le moment sont :

- le bloc de départ, permettant de gérer le lancement du "code" assemblé en blocs ;
- le bloc de texte, affichant une boîte de dialogue à l'écran ;
- les blocs de translations gauche et droite, pour déplacer d'un "pas" un objet dans la scène.

Un menu « bac à sable » a été créé pour le test des blocs. Le rendu est encore simpliste, mais sera bien entendu amené à être amélioré par la suite.

Un grand avantage d'utiliser Blockly pour l'implémentation est la documentation officielle qui s'avère être de bonne qualité et compréhensible sans avoir à rechercher un grand nombre de détails techniques ou d'explications touffues et complexes. La génération des blocs personnalisés est faite grâce au site blockly developer tools (<https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>) qui s'avère bien pratique par son côté plus visuelle et compréhensible qu'une génération à l'aide de fichiers XML ou Javascript, devenant souvent très vite illisibles.

JQueryUI permet de gérer les animations qui peuvent être utilisées au sein de l'espace de travail. Pour l'instant seul les fonctionnalités liées au drag & drop et aux déplacements sont utilisées, mais la rotation sera basée dessus. Une étude plus poussée de la documentation nous permettra peut-être de trouver de nouveaux cas d'utilisation par la suite.

## Création des exercices

Des maquettes ont été réalisées pour chaque exercice. Ce sont des maquettes conceptuelles, comprenant le contenu des exercices ; le design pourra donc encore évoluer durant la suite du projet. Chaque bloc possède trois exercices dédiés :

- le premier est toujours un tutoriel simple, utilisant uniquement (dans la mesure du possible) le nouveau bloc, et possèdera un texte explicatif sur le rôle et l'utilisation de cette fonctionnalité ;
- le second est un exercice simple, permettant d'utiliser ce nouveau bloc dans un contexte concret, avec un ou plusieurs autres blocs vus précédemment ;
- le troisième est dernier exercice sera un exercice « avancé », toujours adapté aux enfants, mais permettant de voir si l'utilisateur a vraiment compris les enjeux de cette fonctionnalité avant de passer à la suivante.

Ces maquettes sont disponibles dans les annexes, à la fin de ce document.

## Difficultés rencontrés

La première difficulté que nous avons rencontrée a été de trouver, choisir et apprendre à utiliser les outils pour la réalisation du projet. En effet, il a été nécessaire de se renseigner sur de nombreuses bibliothèques et de nombreux frameworks afin de déterminer lesquels seraient les plus propices à l'utilisation que nous voulons en faire, mais il a également été très long d'apprendre toutes ces nouveautés en recherchant des cours en lignes et autres tutoriels. L'erreur que nous avons faite à ce moment-là a été de voir trop grand en prenant les techniques les plus adaptées sans calculer le temps d'apprentissage. En prenant quelque chose de moins performant, mais que nous connaissions déjà, nous aurions pu économiser un temps précieux.

D'un point de vue de la gestion de projet, un problème fut de ne pas repérer cette difficulté et ainsi de pouvoir mettre toutes les personnes sur un pied d'égalité au niveau de ces technologies en organisant des séances de mises à niveau réunissant tous les membres du groupe.

La seconde difficulté rencontrée et certainement la plus dure à surmonter fut l'intégration de blocs. Bien que rajouter un bloc sur une page a été relativement rapide, grâce au tutoriel en ligne très bien rédigé de Blockly, il est bien plus complexe de configurer son fonctionnement. Un certain nombre de nouveaux bugs sont apparus à chaque modification du code gérant l'exécution des blocs. Seuls la patience et les nombreux tests nous ont permis d'avancer sur cette partie (que nous sommes toujours en train d'étudier en profondeur).

Sur la création des exercices dédiés, la plus grosse difficulté a été d'adapter le contenu à un enfant de 5 à 7 ans. Après plusieurs recherches infructueuses nous avons opté pour un changement d'approche. Après avoir rédigé quelques maquettes de formats différents, l'avancement s'est finalement fait plus sur le visuel que sur le fond. Nous n'arrivions pas à transmettre parfaitement ce que nous voulions avec de simples images et des phrases courtes et compréhensibles pour un utilisateur de l'âge visé. C'est pourquoi au final nous nous sommes tournés vers la réalisation d'un logiciel où l'enfant sera accompagné d'un adulte. Les exercices restent faciles d'accès, mais les mots et les éventuelles démonstrations du parent accompagnateur permettront de ne pas surcharger les énoncés d'explications que l'enfant seul ne comprendra de toute façon pas.

Pour terminer, nous nous sommes fait surprendre par la contrainte temporelle, qui fut beaucoup plus forte que nous le pensions, suite aux autres travaux que nous avions à réaliser au parallèle. Ce manquement nous a fait prendre un certain retard sur la fin du semestre, c'est pourquoi nous avons décidé de revoir à la baisse notre planning originel. Cette contrainte temporelle a également affecté la gestion de projet en ne permettant pas une réunion hebdomadaire facile à configurer. Au fil des semaines, ces réunions se sont souvent retrouvées à être faites autour d'un café durant une pause entre deux cours ou durant la pause de midi. Un accent particulier sera donc apporté sur ce plan pour la suite des événements, afin de pouvoir préparer les réunions et de pouvoir coordonner l'avancement de manière plus efficace.

## Prévisions pour le semestre suivant

Suite aux difficultés d'avancement du semestre qui vient de passer, nous avons cette fois-ci tenté de mettre en place un planning réalisable.

Il est toutefois possible que ce planning évolue au fur et à mesure de l'avancement du projet.

Janvier	Fin de l'implémentation des blocs de base & Réalisation d'une partie des exercices dédiés
Février	Débogage des fonctionnalités de base, pour continuer sur un projet stable & Suite de l'intégration des exercices dédiés
Mars	Sauvegarde de l'état d'un projet et reprise en cours de projet & Recherches sur l'utilisation du micro et de la webcam pour les intégrer à des exercices (ou au moins en proposer l'utilisation dans le bac à sable)
Avril	Continuation de la sauvegarde et reprise de l'état d'un projet & Implémentation des utilisations du micro et de la webcam
Mai	Finalisation du projet (débogage, amélioration du design et de l'ergonomie si possible) & Rédaction du rapport et préparation de la présentation

Comme annoncé plus haut, nous allons cette fois-ci mettre tout en œuvre pour pouvoir organiser régulièrement (de préférence de façon hebdomadaire) des réunions dans un contexte propice au travail. Ces réunions vont nous permettre de restructurer nos travaux et d'imposer un rythme régulier aux membres du groupe. Elles permettront également de détecter de façon plus rapide et efficace d'éventuels problèmes.

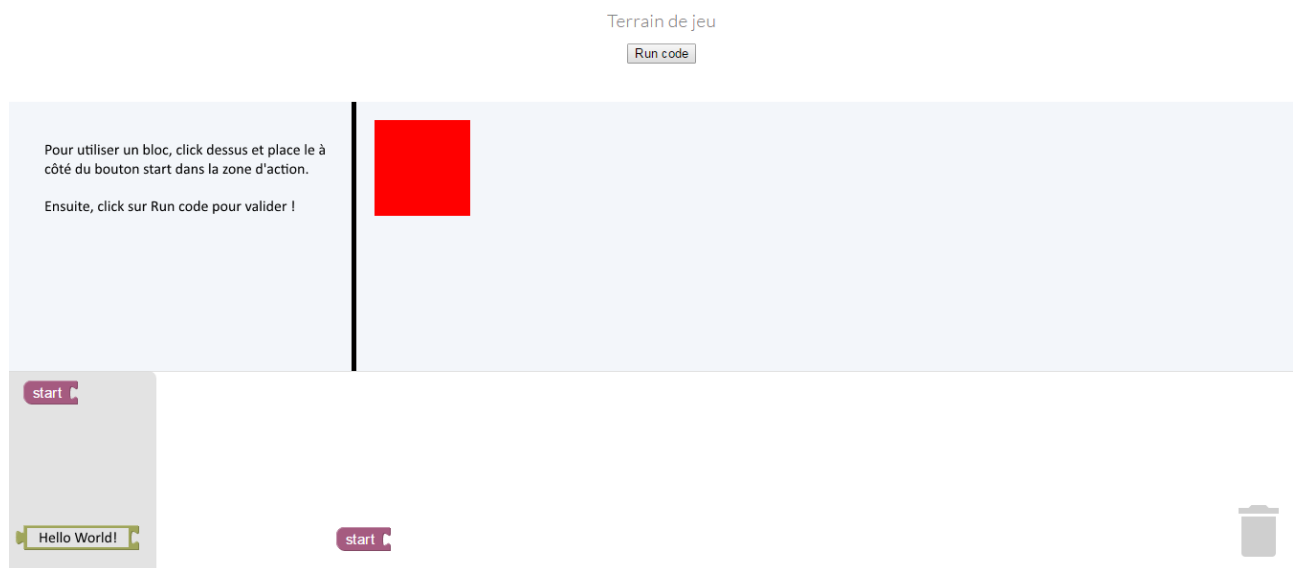
## Annexes

Nous rappelons que ces maquettes sont d'ordre conceptuel. Le design et l'ergonomie ne sont pas définitifs et évolueront au fur et à mesure de l'avancement du projet.

### Exercices de prise en main du site

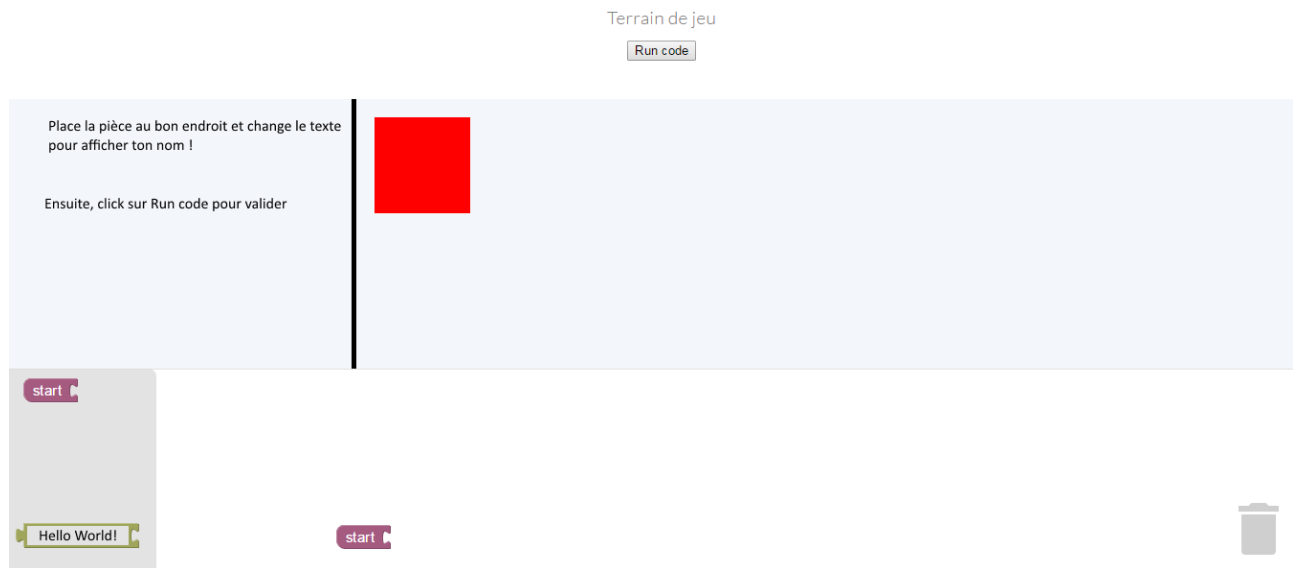
#### Hello World !

L'introduction à ce premier exercice sera faite à l'aide de textes stylisés apparaissant au moment opportun, dans le but de guider avec précision le nouvel utilisateur. On montre à l'utilisateur l'utilisation d'un simple bloc d'affichage.



#### Détails du bloc d'affichage

Le but est ici d'avancer de façon incrémentale et d'informer l'utilisateur de la possibilité d'utiliser les champs d'un bloc.

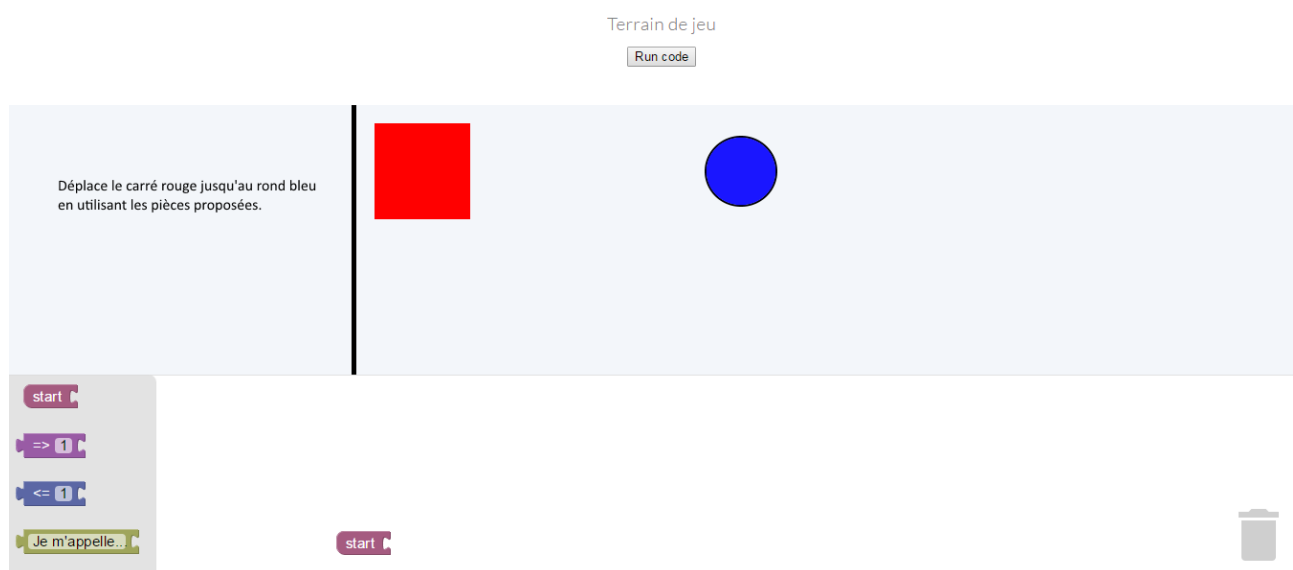


## Les déplacements

Toujours dans le principe de l'incrémentalité, les rotations ont été séparées des translations. L'exercice des déplacements ne concernera donc que les translations.

## Tutoriel de découverte

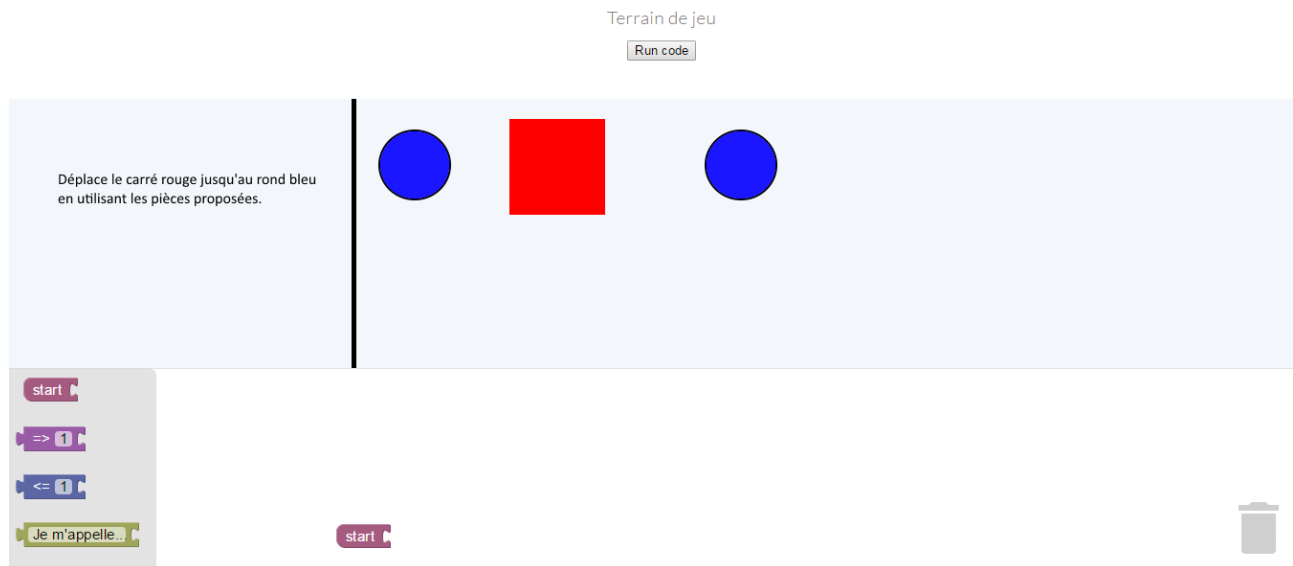
Permet l'utilisation simple d'un nouveau bloc pour une bonne prise en main de l'outil.





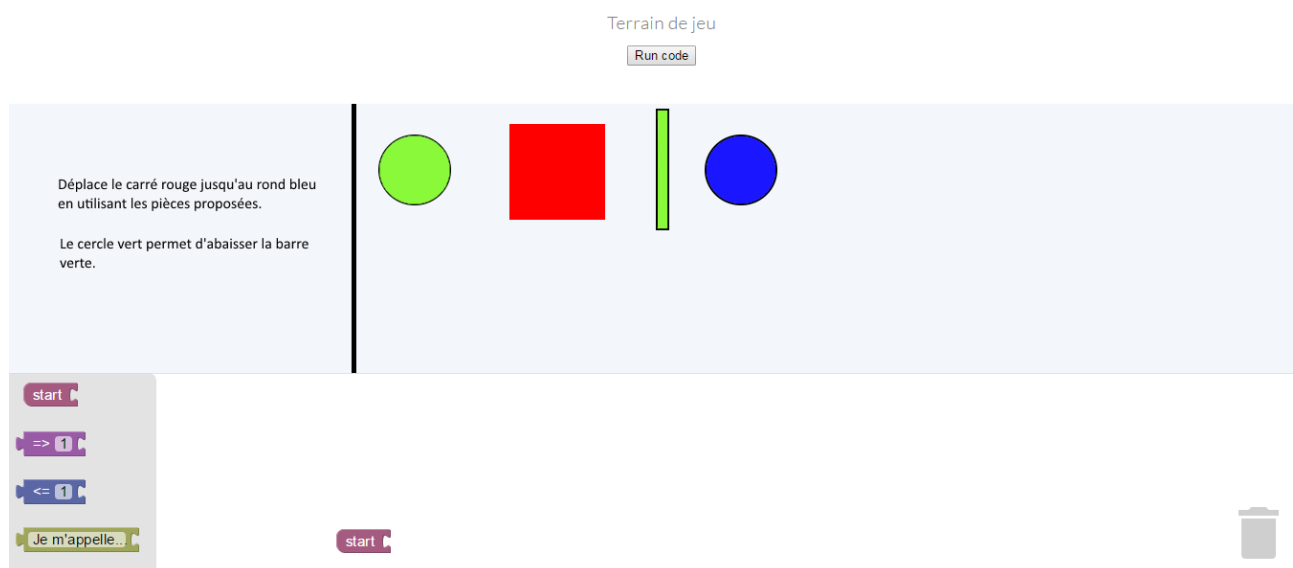
## Exercice simpliste

Afin de s'assurer que tout soit bien compris.



## Exercice avancée

Pour être certain que la notion soit assimilée avant de passer à la suite. On utilise ici une mécanique nouvelle, un peu plus intéressante pour l'utilisateur, qui donnera plus d'intérêt à l'utilisateur qu'un tutoriel trop simplifié.



## Les rotations

## Tutoriel de découverte

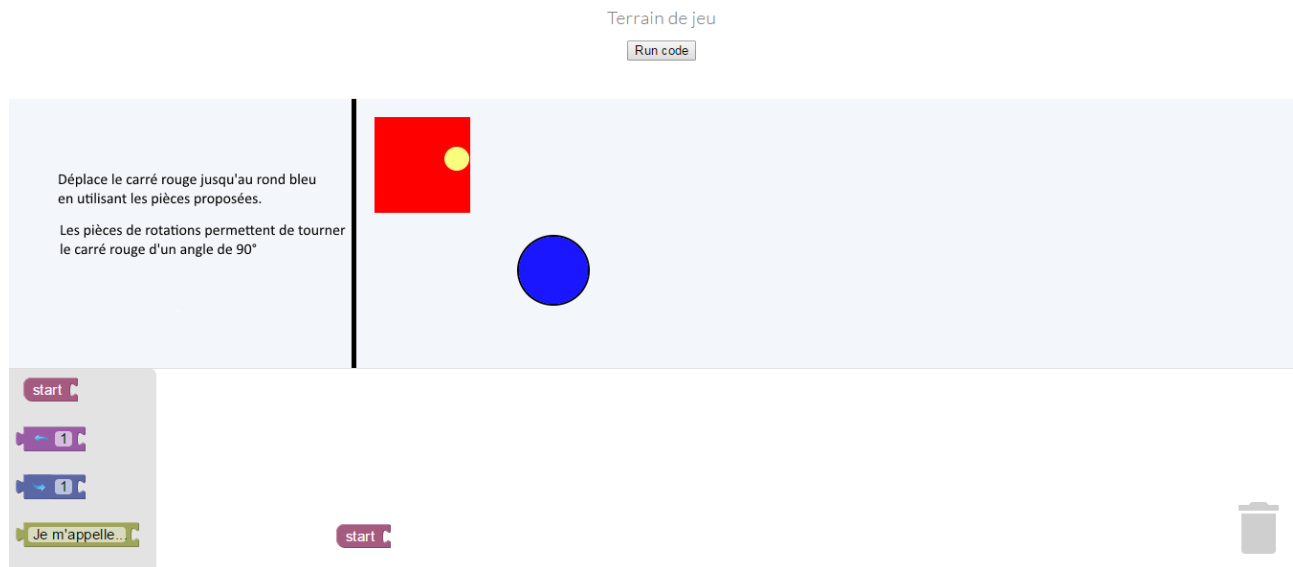
Permet l'utilisation simple d'un nouveau bloc pour une bonne prise en main de l'outil.

Terrain de jeu

Run code

Déplace le carré rouge jusqu'au rond bleu en utilisant les pièces proposées.

Les pièces de rotations permettent de tourner le carré rouge d'un angle de 90°



## Exercice simpliste

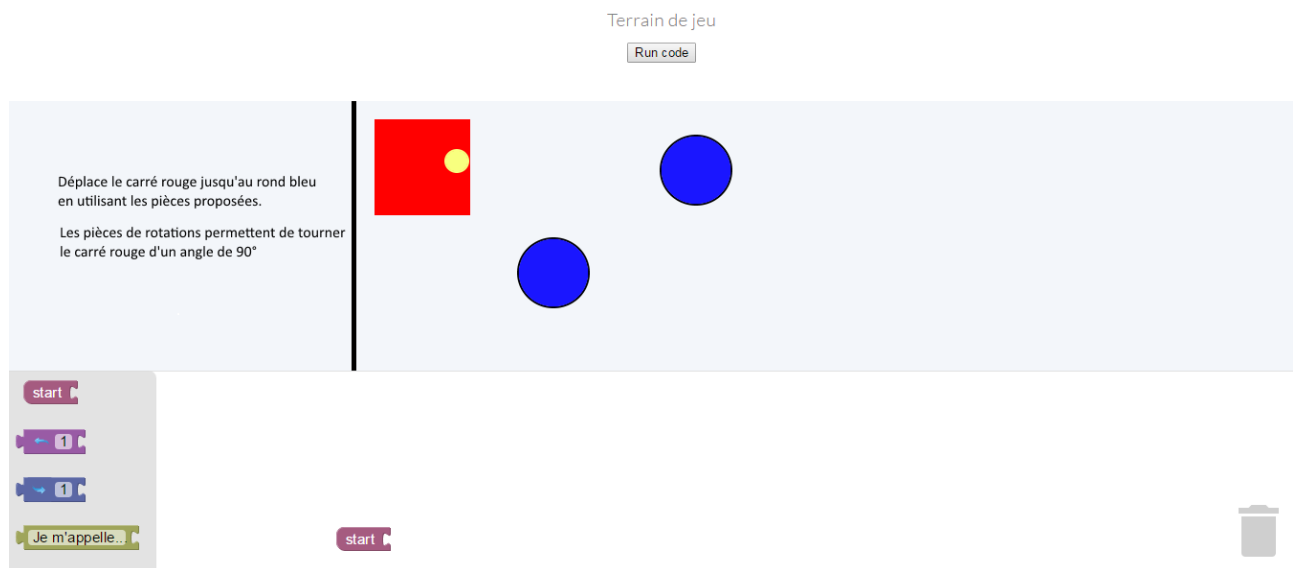
Afin de s'assurer que tout soit bien compris.

Terrain de jeu

Run code

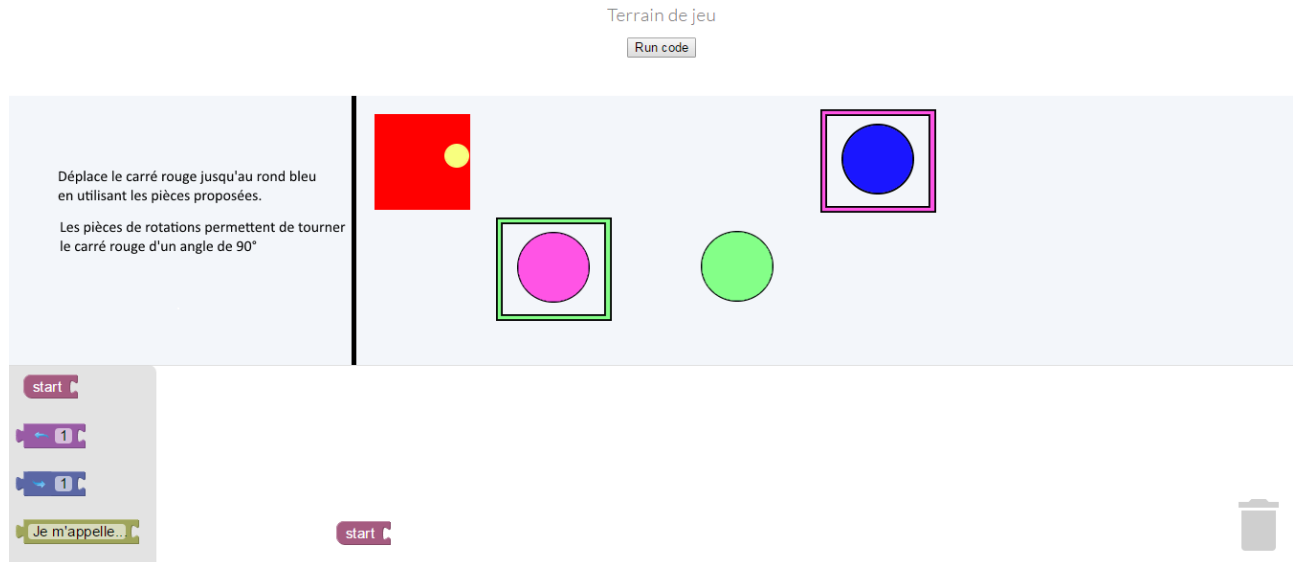
Déplace le carré rouge jusqu'au rond bleu en utilisant les pièces proposées.

Les pièces de rotations permettent de tourner le carré rouge d'un angle de 90°



## Exercice avancée

Pour être certain que la notion soit assimilée avant de passer à la suite.



## La boucle for

Les exercices seront plus détaillés pour la boucle for lorsque celle-ci sera implémentée. En effet, selon la façon dont nous déciderons finalement de l'intégrer, le fonctionnement peut être légèrement modifié.

De la même façon que pour le tout premier tutoriel, des explications supplémentaires seront ajoutées avant que l'utilisateur ne puisse commencer (s'agissant d'une notion beaucoup plus complexe que les précédentes).

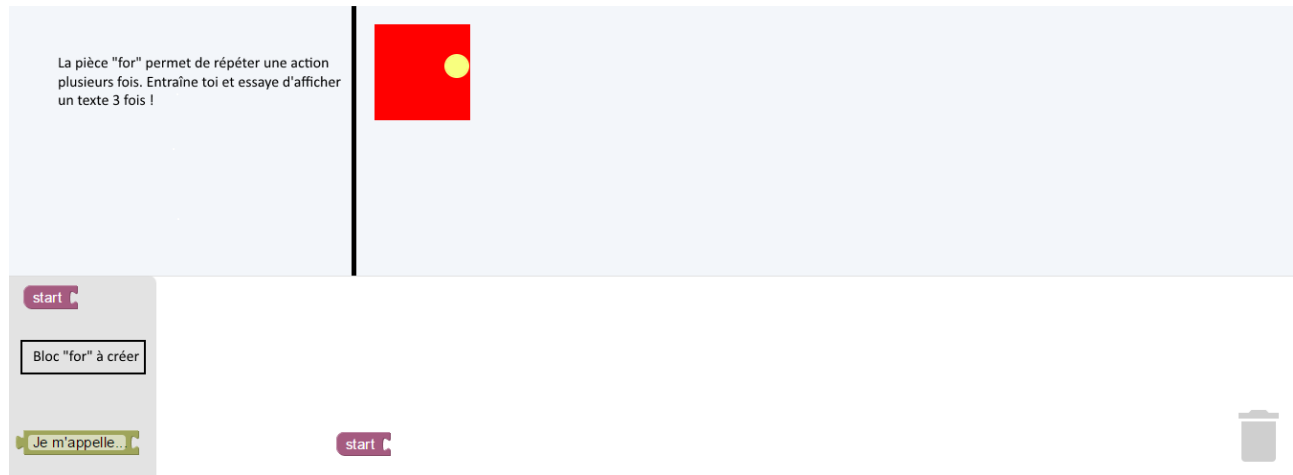
## Tutoriel de découverte

Permet l'utilisation simple d'un nouveau bloc pour une bonne prise en main de l'outil.

Terrain de jeu

Run code

La pièce "for" permet de répéter une action plusieurs fois. Entraîne toi et essaye d'afficher un texte 3 fois !



The image shows a Scratch-like environment. On the left, a vertical panel contains a 'start' button, a text box with 'Bloc "for" à créer', and a 'Je m'appelle...' block. In the center, a red square with a yellow dot is visible. On the right, a large light blue area represents the 'Terrain de jeu' (game field). A 'Run code' button is located at the top right. A trash can icon is on the far right.

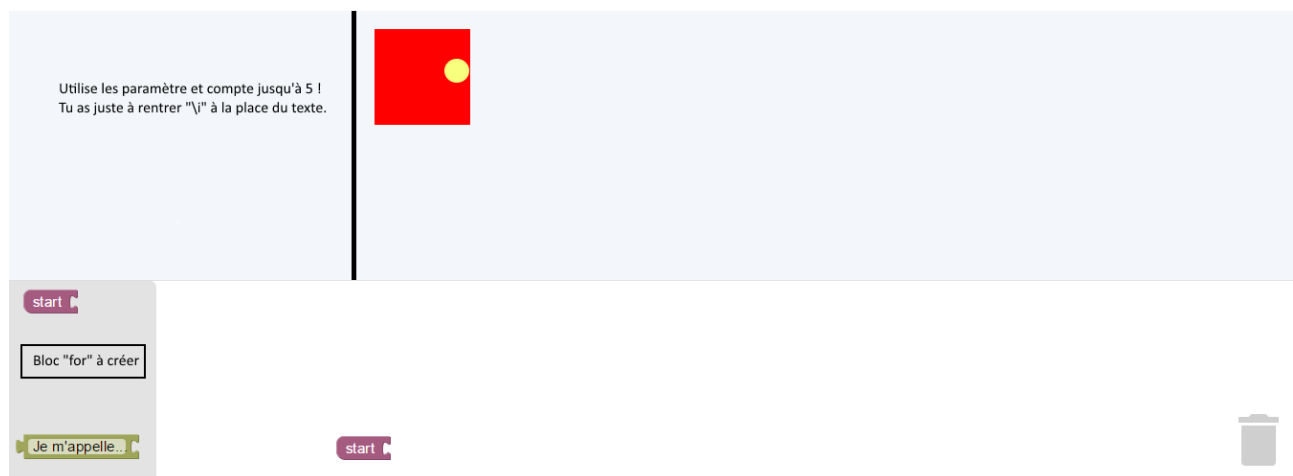
### Exercice simpliste

Afin de s'assurer que tout soit bien compris.

Terrain de jeu

Run code

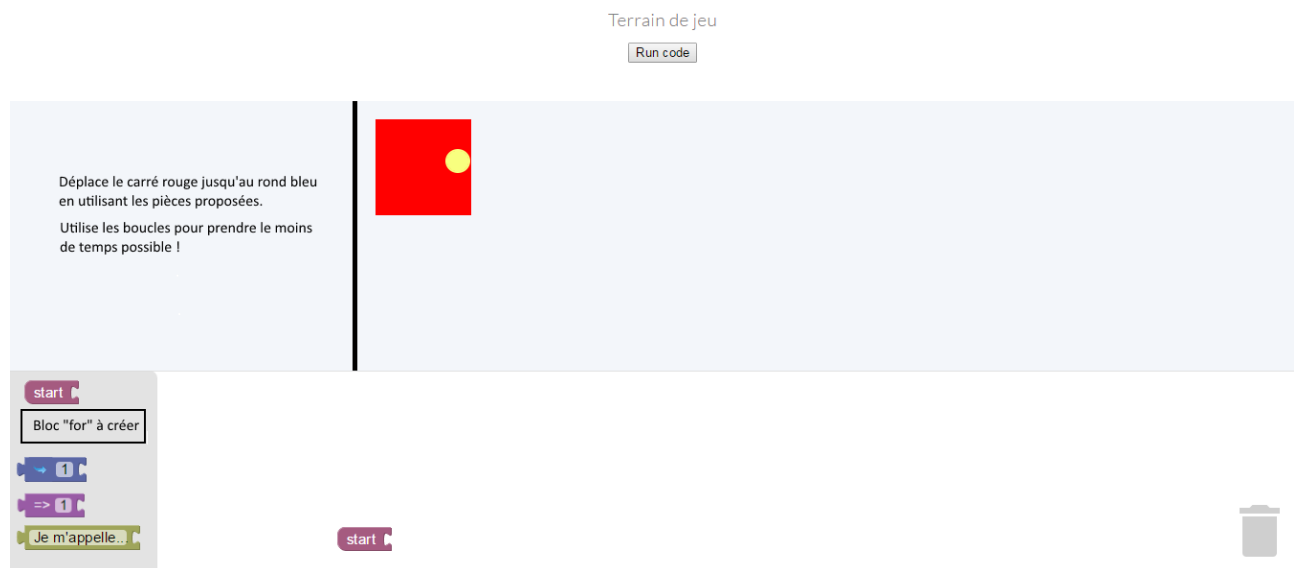
Utilise les paramètres et compte jusqu'à 5 !  
Tu as juste à rentrer "\i" à la place du texte.



The image shows a Scratch-like environment. On the left, a vertical panel contains a 'start' button, a text box with 'Bloc "for" à créer', and a 'Je m'appelle...' block. In the center, a red square with a yellow dot is visible. On the right, a large light blue area represents the 'Terrain de jeu' (game field). A 'Run code' button is located at the top right. A trash can icon is on the far right.

### Exercice avancée

Pour être certain que la notion soit assimilée avant de passer à la suite.



La boucle for constitue le dernier bloc de base que nous souhaitons implémenté (en reprenant le principe de Scratch Jr). Il est fort probable que nous rajoutions des blocs supplémentaires par la suite (les conditions, par exemple) selon l'avancement du projet. A ce moment-là, nous rajouterons des exercices supplémentaires pour ces blocs, en se basant sur le même principe de fonctionnement.