COMPTE RENDU DU PROJET

Team CodeLab3 : Rémy Phol Asa, Antoine Khow,
Thierry Chhoa, Grégoire Marin

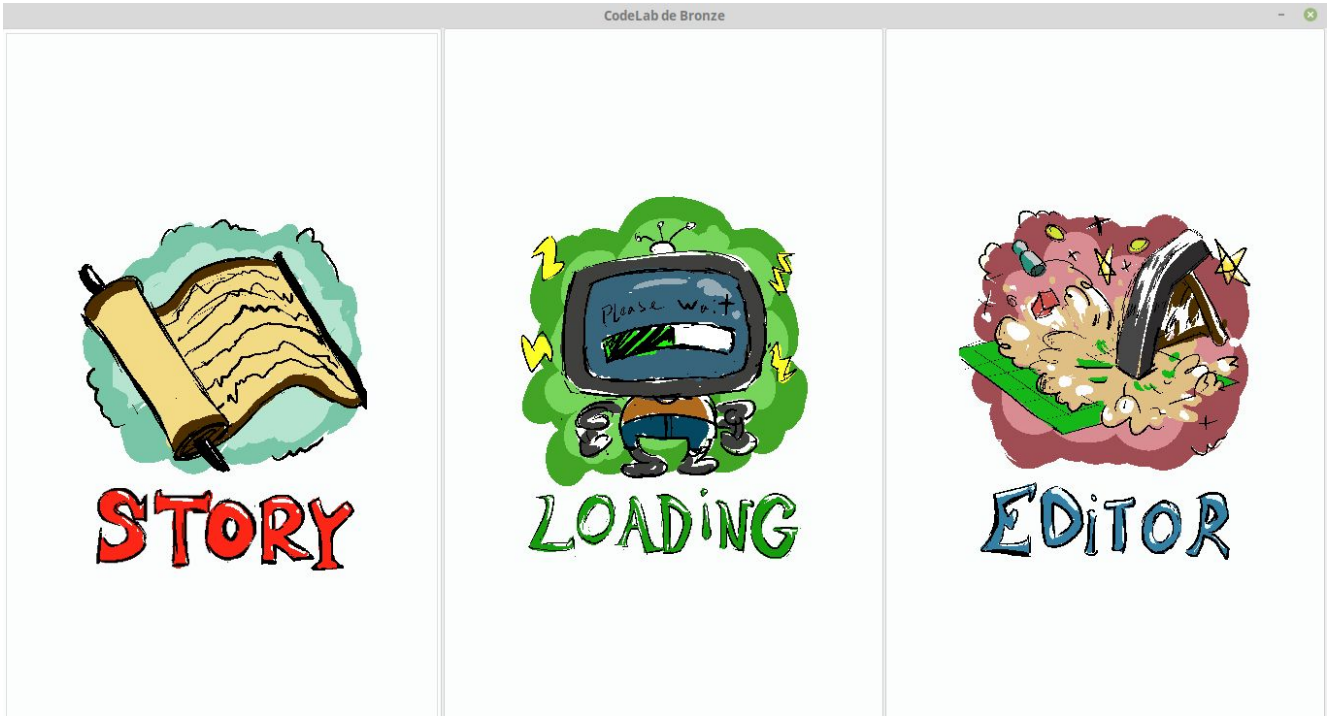
Bienvenue dans notre projet Codelab de Bronze ! Le but de notre projet est de créer une introduction à la programmation sous forme simpliste et pédagogique via le biais du jeu. Le but étant de progresser à travers divers niveaux à difficulté croissante. Cela vous demandera d'employer des instructions de code et de bien gérer les conditions afin d'atteindre l'objectif mais également de créer du code plus ou moins efficace. Nous allons désormais vous introduire notre projet étape par étape :

1) Installation:

Notre installation est simpliste, il vous faudra avant tout compiler le projet avec la commande "make" sur le terminal ouvert à l'emplacement du projet, puis entrée la commande "make run" pour exécuter le Codelab de bronze.

2) Mode d'emploi (et capture d'écran) du projet

Quand vous exécutez le projet, après un délais de compilation, vous arriverez sur la page ci-dessous:



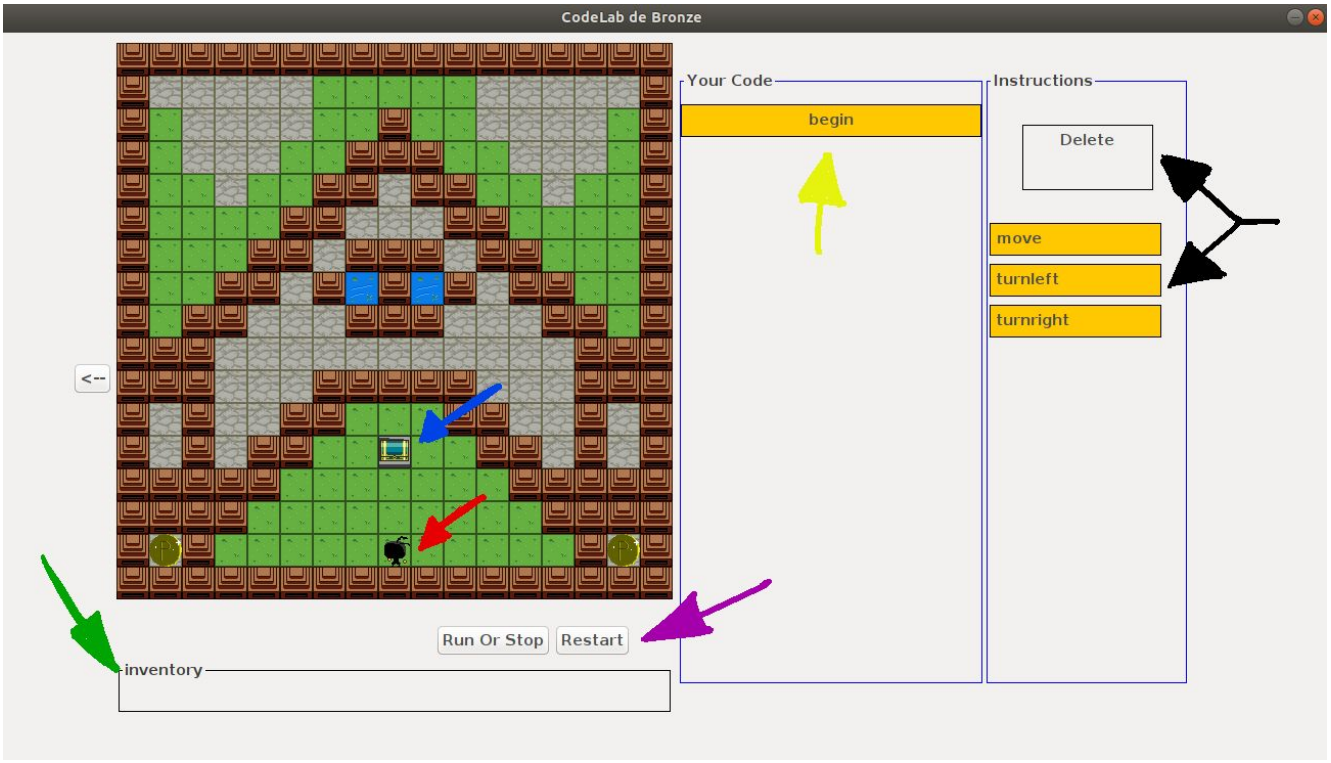
Cette page d'accueil se compose de 3 parties distinctes, respectivement :

- "Story" pour le mode histoire qui présente des niveaux à difficulté croissante
- "Loading" qui permet de charger un niveau déjà créé
- "Editor" qui permet à l'utilisateur de créer son propre niveau

A présent, rien de plus facile, il vous suffit de cliquer sur la zone de votre choix. Nous détaillerons à présent chacune de ces fonctionnalités point par point.

a) ***Story***

Vous avez sélectionné la partie Story, vous ne pourrez que suivre les niveaux proposés par le projet de façon linéaire. Après une pop up indiquant le niveau actuel (et une petite description), vous arriverez sur l'interface suivante :



A votre gauche vous avez le niveau actuel d'affiché, vous pouvez donc faire un constat des lieux. Vous êtes le ninja et vous devez atteindre (vous asseoir) sur le coffre bleu pour terminer le niveau.

A noter également que durant les niveaux vous serez amenés (ou pas) à récupérer une clé pour ouvrir des portes ou collecter des pièces, si cela est fait, vous pourrez voir ces objets s'afficher dans votre inventaire.

Maintenant que vous êtes familier avec le niveau, il va falloir coder, l'instruction "Begin" indiqué par la flèche jaune vous indique le début de votre code, en l'état, il n'y a rien. Pour l'étoffer vous devez glisser les instructions en gris indiqué par la flèche noire vers "Begin" et si vous faites une erreur rien de compliqué il suffit de glisser à nouveau l'instruction situé dans la zone de "Begin" vers "Delete" pour la supprimer.

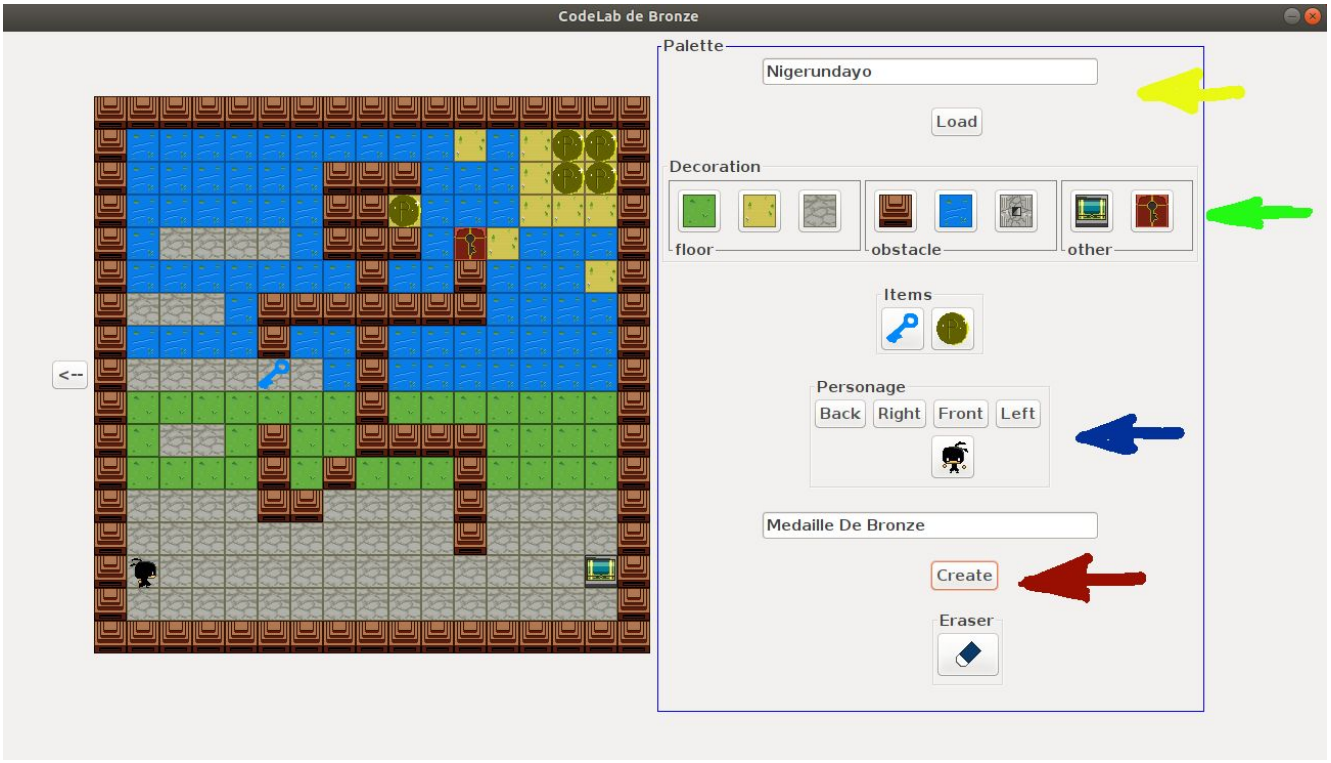
A présent l'exécution, il vous suffira de cliquer sur "Run or Stop" pour exécuter ou mettre en pause votre programme, si vous voulez recommencer il suffira de cliquer sur "Restart" . Si vous gagnez vous pourrez passer au niveau suivant, sinon vous serez invité à réessayer.

b) ***Loading***

Vous avez sélectionné la partie Loading, elle vous permet de charger un niveau déjà créé, il vous suffit de rentrer le nom du niveau que vous voulez charger (ce nom est normalement celui que vous avez donné à votre niveau lors de sa création dans l'éditeur de niveau) puis de cliquer sur le bouton "Confirm" ce qui lancera votre niveau avec la même interface que pour la Story (cf. partie Story). Vous disposez aussi d'un bouton de retour à la page d'accueil juste à gauche de la barre de texte.

c) ***Editor***

Vous avez sélectionné la partie Loading, elle vous permet de créer vos propres niveaux, génial n'est-ce-pas ? Cet éditeur est séparé en 2 parties, à gauche la map où vous pourrez placer le décor et les entitées et à droite l'emplacement du générateur d'éléments (murs, avatar, coffre, …). Pour pouvoir placer une entité ou un décor, il faudra sélectionner le bouton voulu puis cliquer sur la case du terrain choisie.



Passons désormais à la fonctionnalité "Load" qui vous permet de charger un niveau déjà créé pour faciliter les modifications en cas d'oubli ou de commencer avec une base pour éviter de tout replacer.

Cap sud-est (en bas à droite) vous pouvez voir une gomme, il s'agit d' "Eraser" vous permettant de supprimer une Entitée présente sur le terrain, à noter toutefois que cette fonctionnalité ne marche pas pour le décor, il faudra utiliser le décor par défaut "Terrain gazonné" pour en écraser un (exception du coffre vous devrez le déplacer en posant un autre coffre ailleurs).

Vous voulez à présent placer le personnage, regardez donc la zone "personage" vous permet de choisir la position de départ du personnage, il faut d'abord sélectionner l'une des 4 directions avant de placer le personnage, une fois placer cette position est définitive. Pour le changer il vous faudra replacer le personnage avec le bon positionnement.

Une fois que vous avez fini, le bouton "Create" vous permet, dès que votre niveau est prêt, de le créer, il vous suffira juste de donner un nom à votre niveau pour pouvoir le sauvegarder. Si vous lui donner un nom déjà existant, un message d'erreur vous le signalera.

1) Essentiels du CodeLab

A. Language model :

Les instructions composant notre langage sont représentées sous forme d'objets. Il y a trois classes abstraites dont héritent les instructions : les conditions, les actions ainsi que les structures de contrôle (if et while) qui sont particulières. Les instructions sont regroupées dans une liste et chaque instruction est exécutée grâce à une méthode run(). Chacune d'entre elles sont liées au joueur et l'ordonne, soit de se déplacer, soit d'observer les alentours. Si la syntaxe présentée par l'utilisateur est incorrecte, un message d'erreur apparaîtra.

B. Language vue :

Quant au langage, celui-ci est manipulé par des blocs d'instructions. Il n'y a pas d'édition par texte. L'utilisateur peut créer, déplacer et supprimer ces blocs pour fabriquer son programme.
La vue du langage n'est pas une vue dans le pattern Modèle-Vue-Contrôleur (mvc). Si l'on change la représentation visuelle cela ne changera pas les instructions du modèle. C'est uniquement lorsque l'utilisateur appuie sur le bouton pour exécuter son programme que la représentation visuelle créé un équivalent avec les instructions du modèle.

La partie visuelle du langage est séparée en deux parties:

- A droite sont placés des générateurs d'instructions. L'utilisateur peut glisser ou déposer (drag n' drop) des panels d'instructions pour créer son programme.
- A gauche se trouve le programme de l'utilisateur. Chaque instruction est encapsulée dans un panel la représentant visuellement. Les panels sont regroupés dans une liste chaînée. Chaque panel est "écouté" par un contrôleur pour être manipulable par la souris.

Deux interfaces sont utilisées pour factoriser les comportements des panels d'instructions. Les panels contenant une condition (if, while, not...) et les panels contenants une liste d'action (if, while, EditPanel)

C. ***World Model :***

Le modèle du “World” (ou Monde) correspond à la modélisation du terrain de simulation, c’est à dire l’espace dans lequel le programme écrit par le joueur va s'exécuter. Le modèle se divise en deux grandes sous-parties :

- Le Board et ses Décors qui vont définir les comportement du terrain. Les Décors sont composés de Mur, de Porte et de Sol. Nous avons séparé les Décors en deux parties : les Obstacles et ceux qui n’en sont pas. Le Board à lui pour rôle d’effectuer toutes les actions demandés par les Entités, il pourra ainsi déplacer une Entité, la supprimer, etc… Il effectue ainsi les actions bas niveau en prenant en compte les Décors et Entités présents.

- Les Entités sont des éléments ayant des comportements pouvant affecter le Board. Il en existe deux types : les Personnages qui sont des entités pouvants être exécutés et interagir avec le monde selon un comportement plus ou moins complexe. Ils ont la capacité de demander au Board d’effectuer certaines actions comme son propre déplacement ou bien de demander des informations sur les Décors environnants. Ce sera ce type d’Entités auquel le Joueur peut affecter des Actions.
  Les Collectables qui sont des Entités plus simple. Elles n’interagissent pas ou peu avec son environnement. Ceux-ci peuvent être des Pièces (pas de comportement particulier) ou des Clés (permettant d’ouvrir les portes). Ils disparaissent du plateau lorsqu’ils sont ramassés.

D. ***World Vue :***

La vue du “World” est faite en deux Étapes :

- Peindre chaque case du plateau une par une selon le Décor sélectionné.
- Peindre les Entités en regardant leurs positions respectives.

Nous avons utilisé des images de 32 x 32 pixels pour peindre chaque élément du board, dont le chargement se fait via l’utilisation de l’objet “ImageLibrary” qui permet de charger et stocker toutes les images nécessaires selon les besoins.

E. ***Level :***

Le level est une classe à part qui permet de gérer tous les comportements qui ne sont pas directement liés à la simulation du jeu comme la capacité à recommencer le niveau, à charger un Board et l’initier ou à lancer la simulation. Il fait le pont entre le “World” et le “Language”.

2) ***Les petits rouages du CodeLab***

A. ***Editeur***

L'éditeur de niveau a été réalisé en utilisant des boutons spéciaux fait main, les “PlacementsButtons”. En effet, ces boutons ont la capacité de donner un “algorithme/recette”

particulier aux “ControllerEditor” pour qu’il puisse affecter le Board en y ajoutant les différents Décors et Entités. Les comportements généraux de ces boutons peuvent être résumés par : “Je créer mon Entité/Décors et je veux la placer à cet endroit là!”
Le placement des éléments se fait ensuite via le “ControllerEditor” qui va récupérer la position choisie grâce au “BoardEditor” correspondant.
Une fois le Board bien initialisé, on sauvegarde une copie du niveau en récupérant le nom choisi. Les classes qui divisent les éléments en plusieurs parties n’ont qu’un but esthétique et pour ne pas avoir un éditeur en désordre. A noter que ces niveaux sont sauvegardés dans le répertoire resources/.
Le chargement pour l'éditeur est tout simplement le même que le chargement d’un niveau mais sans la capacité de donner une interface language à l’utilisateur.

B. Histoire

Le mode histoire permet de progresser à travers les niveaux de manière fluide. Les fichiers du dossier resources/story/ permettent de charger les niveaux de l’histoire et de sauvegarder la progression à l’aide du fichier sauvegarde.json. Chaque niveau de l’histoire affiche une pop up à son chargement en guise d’introduction. Également, les instructions disponibles pour les niveaux du mode histoires sont limitées et se débloqueront en parallèle de la progression du joueur. Enfin, lorsque toute l’histoire est terminée, un magnifique dessin apparaît pour féliciter le joueur.

C. Contrôleurs

Les Contrôleurs sont toutes les classes permettant de faire le lien entre la “Vue” et le “Model” et ne servent en général pas plus que de relais. Le seul Contrôleur “compliqué” est le “ControllerLevel” qui gère des actions complexes comme l'exécution de la simulation en temps réel en utilisant les Thread et un système “particulier” d’interrupteur.