

Mini Tramways

Présentation et étude préliminaire

Présentation du projet

Mini Tramways est un jeu de type "Arcade" pour PC qui consiste en la construction et la gestion d'un réseau de tramways qui doit desservir l'ensemble des bâtiments de la carte. Dans ce jeu, le joueur devra créer des lignes de tramways pour relier différents quartiers entre eux (résidentiels, commerciaux ou quartiers d'affaires) de manière à ce que les habitants soient satisfaits. Si le réseau n'est pas suffisamment optimisé et que les voyageurs doivent attendre trop longtemps avant d'être transportés, le joueur perd la partie.

Les quartiers seront des zones prédéfinies sur la carte, mais les bâtiments qui les composent seront générés au fil de la partie, de manière aléatoire. Les stations de trams seront, quant à elles, placées par le joueur et reliées entre elles par des lignes. Chaque ligne nouvellement créée se verra attribuée un wagon, qui effectuera des aller-retours incessants d'un bout à l'autre de la ligne. Il sera possible d'ajouter d'autres stations à une ligne existante en la prolongeant.

Pour construire de nouvelles stations et lignes, le joueur devra dépenser de l'argent acquis en transportant des voyageurs. Un voyageur non satisfait (par exemple, s'il a dû attendre trop longtemps avant d'être transporté) demandera le remboursement de son billet, et ne fera donc pas gagner d'argent au joueur.

Éventuellement, un ensemble de fonctionnalités supplémentaires sont envisageables, comme par exemple la sauvegarde de la partie ou la génération aléatoire des quartiers.

Table des matières

Présentation et étude préliminaire	1
Présentation du projet	2
Étude des solutions existantes	4
<i>Mini Metro</i>	4
<i>Mini Motorways</i>	5
<i>SimCity</i> et <i>Cities : Skyline</i>	5
État de l'Art	6
Les zones de la carte	6
Parcours de Graphe	6
Le quotidien des usagers	7
La satisfaction des usagers	7
Entre résidence et stations	7
Choix des technologies	8
Langages	8
Frameworks	9
Moteurs de jeux	9
Technologies retenues	10

Étude des solutions existantes

Beaucoup des principes de *Mini Tramways* sont inspirés de jeux existants. Quatre jeux en particulier peuvent constituer des sources d'inspiration très utiles au développement de *Mini Tramways*.

Mini Metro

Tout comme *Mini Tramways*, *Mini Metro* est un jeu de construction et de gestion de réseaux de transports en commun.

Pour ne pas perdre la partie, le joueur doit faire en sorte que les voyageurs ne s'accumulent pas trop dans une station. Si trop de personnes attendent à une même station, la partie se termine.

Visuellement, le jeu est tout ce qu'il y a de plus minimaliste. C'est d'ailleurs en partie ce qui a fait son succès, de par le peu de ressources graphiques qu'il nécessite.

En effet, il se présente sous la forme d'une représentation abstraite d'un réseau de transport en commun, à l'instar d'une vraie carte de transports. Les différentes stations sont représentées par des formes géométriques basiques (carrés, triangles, cercles) et les lignes par de simples traits de couleurs.

Le principal défi du jeu est de construire le réseau le plus efficace possible. Chaque passager souhaite se rendre dans une station correspondant à une certaine forme géométrique. Ainsi, tous les passagers ne se déplacent pas simplement d'une station à la suivante : certains peuvent rester dans le wagon pendant de longues minutes

Pour ce qui est du système de jeu, le joueur a la possibilité de relier les stations à l'aide de lignes de couleur. Une fois la ligne créée, un véhicule se met à y faire des aller-retours : les voyageurs peuvent s'en servir pour atteindre leurs destinations. Les lignes peuvent être librement détruites ou réorganisées à n'importe quel moment de la partie. Le joueur a la possibilité de mettre en pause le jeu pour pouvoir réfléchir à sa réorganisation.

Similitudes avec Mini Tramways

Le système de création et de modification de lignes de Mini Metro est particulièrement intéressant, puisque nous aurons besoin d'implémenter une fonctionnalité sensiblement identique, à la différence près que les stations seront créées par la ligne. De plus, le comportement des voyageurs peut également être un exemple duquel s'inspirer, puisque les problématiques sont très proches dans *Mini Tramways* (notamment l'obtention des itinéraires à emprunter).

Mini Motorways

Suite de *Mini Metro*, *Mini Motorways* reprend le concept de gestion de réseau de transports. Mais au lieu de lignes de métro, il s'agit cette fois de construire un réseau routier efficace. Le but du jeu est donc de connecter des maisons de différentes couleurs, d'où sortent des voitures, à des bâtiments commerciaux et autres entreprises. Dans ces entreprises apparaissent des « épingles » que les voitures de la bonne couleur doivent collecter. Il s'agit donc d'avoir un trafic suffisamment fluide pour réussir à collecter les épingles avant qu'elles ne s'accumulent trop, au risque de perdre la partie. Au fil du temps, de nouveaux bâtiments apparaissent, de manière plus ou moins chaotique ; des maisons de même couleur auront tendance à former des petits « quartiers », des hameaux.

Similitudes avec Mini Tramways

Le système de génération des bâtiments de *Mini Motorways* est assez proche de ce que nous cherchons à réaliser pour *Mini Tramways*, exception faite du système de « quartiers » : en effet, nous ne distinguons pas les bâtiments d'un même type par différentes couleurs, mais par leur appartenance à une zone de la carte que l'on nommera « quartier ». De plus, à chaque quartier est associé un type de bâtiments donné : commerces, bureaux, ou habitations.

SimCity et Cities : Skyline

Cities : Skyline et *SimCity*, qui sont deux piliers des jeux de gestion de villes, possèdent quelques similitudes avec les jeux précédemment cités. Par exemple, dans les deux jeux, le joueur dispose d'outils lui permettant de "colorier" des zones sur la carte, pour choisir de créer des zones résidentielles, commerciales ou industrielles. Une fois cela fait, le jeu va générer automatiquement les bâtiments petit à petit dans ces zones, selon la demande.

Ces deux jeux permettent également de créer des réseaux de trains, trams ou métros, même si ce n'est pas leur intérêt principal, et que leur gestion est limitée (on peut tout de même, dans *Cities : Skyline*, gérer les lignes individuellement, le nombre de véhicules qui circulent sur celles-ci, et disposer de statistiques complètes sur le trafic).

Similitudes avec Mini Tramways

La génération de bâtiments d'un certain type dans des zones données est un principe que nous exploiterons dans *Mini Tramways*. Il peut donc être intéressant de se pencher sur la façon dont les bâtiments sont générés dans ces jeux (notamment leur répartition dans la zone) afin de disposer d'éventuelles bases.

La gestion des trains, trams et métros dans ces jeux est, quant à elle, trop rudimentaire et trop éloignée de ce que nous recherchons pour *Mini Tramways*. Ainsi, nous nous en tiendrons à *Mini Metro* concernant ce domaine.

État de l'Art



Dans le cadre des études préalables au développement de *Mini Tramways*, nous avons pu nous appuyer sur un ensemble de jeux existants reprenant des fonctionnalités similaires à celles que nous devons développer. Nous allons donc établir un État de l'Art afin de déterminer quelles recherches nous devons réaliser pour répondre aux différents problèmes que nous rencontrerons au cours du développement.

Les zones de la carte

Dans *Mini Tramways*, la carte comprendra trois types de zones : résidentielles, commerciales, et de bureaux. La question est de savoir comment générer ces trois zones. Au début du développement, les zones seront prédéfinies « en dur » dans le code, mais l'objectif à terme est de pouvoir implémenter une génération procédurale des zones. Ce sujet est un sujet largement exploré, par un grand nombre de jeux et d'études. Par exemple, une thèse de Adrien Peytavie sur le sujet explique le principe en détail. Nous pourrions également nous inspirer des systèmes de générations de biomes de *Minecraft*, ou de mods pour ce jeu qui disposent d'une fonctionnalité similaire.

Parcours de Graphe

Dans *Mini Tramways*, les effectueront des aller-retours de bout en bout de leur ligne. Afin de déterminer le trajet que doit emprunter une personne, nous aurons besoin d'évaluer le temps que peut représenter chaque portion de ligne. Si deux lignes de tramways ont un arrêt en commun, le problème est la synchronisation des trams : déterminer si le passager doit attendre le prochain tram ou faire un détour pour arriver à destination plus rapidement n'est pas trivial. L'algorithme de Dijkstra pourrait être une bonne base, mais le problème de la synchronisation rendra cet algorithme non-optimal dans de nombreux cas. Une thèse de Grégoire Scano détaille ce problème.

Cf. Calcul d'itinéraires multiples et de trajets synchronisés dans des réseaux de transport multimodaux, Grégoire Scano, 2016

Le quotidien des usagers

Dans *Mini Tramways*, les habitants voudront se rendre soit dans une zone commerciale, soit dans une zone de bureaux. Afin de dresser un comportement pseudo-réaliste de ceux-ci, nous pourrions nous appuyer sur une étude de l'INSEA qui nous permet de déterminer que 35% des personnes dans le tram sont des étudiants, et que ce sont les personnes les plus souvent dans un tramways hors nous n'avons pas d'étudiants dans Mini Tramways, nous pourrions couper la poire en deux pour avoir 50% de personnes voulant aller en zone industrielle car 29.6% des personnes sont déjà des travailleurs, ainsi le 50% restant irais dans les zones commerciales, on considèrera également qu'une personnes partie dans une zone devra revenir dans sa maison au bout d'un certain temps

Cf. Enquête sur la satisfaction des usagers du tramway dans la Région de Rabat-Salé, Institut National de Statistique et d'Économie Appliquée du Maroc, 2011-2012

La satisfaction des usagers

Mini Tramways possèdera un système de gestion de la satisfaction qui, si la satisfaction des voyageurs est trop basse, déclenchera la fin de la partie. Dans la même étude de l'INSEA, nous apprenons que 80% des personnes attendant le tram sont prêtes à attendre 20 minutes maximum, et 20% sont prêts à attendre plus longtemps. Ces données pourraient servir de base pour fixer les valeurs permettant de calculer la satisfaction des personnes.

Cf. Enquête sur la satisfaction des usagers du tramway dans la Région de Rabat-Salé, Institut National de Statistique et d'Économie Appliquée du Maroc, 2011-2012

Entre résidence et stations

Comme dans la vraie vie, dans *Mini Tramways* les personnes se dirigeront naturellement vers les stations les plus proches de chez eux. Toujours dans l'étude citée précédemment, nous pouvons lire que 57,7% des personnes choisissent la station la plus proche de chez eux, on pourrait donc considérer que 60% des gens iraient dans les stations les plus proches et que les 40% restant choisiraient leur stations en fonction de la saturation de celle-ci, dans un souci de réalisme.

Choix des technologies



Langages

C

Le langage C est un langage de bas niveau, il offre donc un contrôle important, notamment sur la gestion de la mémoire, mais nécessite plus de temps de développement qu'un langage de plus haut niveau. Il n'utilise pas le paradigme orienté objet, ce qui constitue un inconvénient indéniable pour le développement d'un jeu. Les bibliothèques natives du langage C sont beaucoup moins développées que celles des langages de plus haut niveau, car le langage C est principalement utilisé pour créer des compilateurs ou interpréteurs. Il n'est donc pas adapté pour le développement de ce jeu.

C++

Le langage C++ est un langage de bas niveau, évolution du C. La principale différence avec le C est qu'il utilise le paradigme objet, comblant ainsi son plus gros défaut. Il offre de meilleures performances à l'exécution que la plupart des autres langages. Le langage C++ est un standard de l'industrie, il existe donc beaucoup de support et d'outils pour ce langage.

Il dispose de nombreuses bibliothèques telles que SDL qui permet de gérer l'audio, les différentes entrées clavier/souris ou joystick. De plus cette bibliothèque est multiplateforme. Elle est accessible sur de nombreux logiciels grâce à la licence Zlib

C#

C# se rapproche du Java. Il utilise le paradigme objet. Il existe beaucoup de bibliothèques et Frameworks comme Unity ou Monogame qui permettent de créer des jeux plus rapidement. C# propose des fonctionnalités dont Java ne dispose pas, comme les propriétés explicites ou la surcharge des opérateurs.

Java

Le langage Java est un langage de haut niveau. Il utilise le paradigme orienté objet. Très utilisé, ce langage dispose de nombreux frameworks et outils pour accélérer le développement. Il existe de nombreuses bibliothèques Java adaptées aux jeux, comme par exemple LWJGL qui permet d'exploiter des moteurs graphiques de bas niveau comme OpenGL ES ou Vulkan. Cependant, Java nécessite plus de ressources que la plupart des autres langages, car il est exécuté dans une machine virtuelle (ce qui, par la même occasion, le rend parfaitement multiplateformes).

Python

Python est un langage de haut niveau. Il utilise le paradigme orienté objet. Il existe de nombreuses bibliothèques pour créer des jeux en python comme Pygame qui elle-même est basée sur la bibliothèque SDL. Pygame permet de gérer les différentes entrées sorties et la partie graphique du jeu. Python est un langage interprété, ce qui le rend moins performant que C++ ou C#.

Frameworks

JavaFX

JavaFX est un Framework qui permet de gérer une interface graphique utilisateur avec Java. Sous licence GNU, ce framework multiplateforme est plutôt adapté à la création de jeux, car il permet de gérer toutes sortes d'éléments visuels (images, dessin de formes...).

Swing

A l'instar de JavaFX, Swing est un framework qui permet de créer des interfaces graphiques. Il est compatible uniquement avec des machines virtuelles Java. Sous licence GPL, Swing utilise nativement une structure MVC.

MonoGame

MonoGame est un framework pour le langage C# multiplateforme. Il utilise les mêmes principes que OpenGL ou DirectX, et prend en charge les aspects graphiques, audio, et les entrées/sorties du jeu.

Moteurs de jeux

Unity

Le moteur Unity est une solution « tout-en-un » qui permet de développer des jeux de toutes sortes sur différentes plateformes : ordinateurs, consoles, mobiles, web, etc. Il utilise le langage C#. Il est compatible avec de nombreuses bibliothèques comme OpenGL ou Vulkan. Unity possède une importante communauté. Le premier inconvénient de ce type de moteur est qu'il est beaucoup plus demandeur en ressources, à cause du grand nombre de fonctionnalités qu'il prend en charge. De plus, son prix doit être un élément pris en compte.

Godot

Godot est un moteur qui permet de développer des jeux en 2D mais aussi en 3D. Il possède aussi un moteur physique et un gestionnaire d'animation. Il est multiplateforme comme Unity. Il supporte différents langages comme C#, C++ ou encore Python. Il est sous licence MIT, c'est-à-dire qu'il est libre et open source, et peut donc être modifié à souhait selon les besoins.

Technologies retenues

D'un point de vue purement technique, le langage le plus adapté au développement de *Mini Tramways* semble être le C#, de par les fonctionnalités intéressantes qu'il propose et les performances légèrement supérieurs à celles du Java. Néanmoins, la solution retenue par le client est le langage Java couplé au framework JavaFX, ce sont donc ces outils qui seront utilisés pour le développement de *Mini tramways*.