

ANALYSE SYNTH

PARTIE COMMUNE

Description rapide

Analysynth est un jeu d'Action Aventure solo sur PC en 3D. Le joueur incarne Syntheros un épéiste ayant perdu la vue lors d'un duel contre le maître d'arme Analysthes. Suivant les conseils d'Analysthes, Syntheros part améliorer sa maîtrise de l'épée pour le défaire lorsque l'heure de leur prochaine rencontre sonnera...

Description expérience et interactivité

Analysynth est un jeu sur la maîtrise et comment elle peut s'exprimer via le synthétisme. Le jeu sera constitué de quatre niveaux et de 6 boss (Furi). Chaque niveau se terminera par un boss. A chaque boss vaincu, Syntheros passera un cap de maîtrise.

Le jeu a trois composants essentiels :

🛡 Les combats

Les combats se concentrent sur la parade et la riposte (Sekiro). L'ensemble des techniques de combat sont disponibles dès le début. Cependant elles ont toutes une combinaison de touche différentes associés et sont donc complexe à utiliser. Lors de la synthétisation du game design, ces différents coups seront assigné à une même et unique touche.

🎮 La progression

Le jeu commence par un combat contre Analysthes, servant aussi de tutoriel. Suite à cela, le joueur sera libre de choisir le niveau auquel il souhaite jouer. Chaque niveau sera représenté par une île. Pour chaque île, Syntheros devra vaincre l'ensemble des sbires présent avant de pouvoir combattre le boss. S'il échoue avant le boss, sa progression est perdue et il doit recommencer le niveau. S'il échoue contre le boss, il n'a pas à refaire le niveau, seulement à recommencer le combat contre le boss. Lorsque les boss des 4 îles sont vaincus, Syntheros peut alors aller affronter Analysthes le boss final du jeu.

📁 La synthétisation

Chaque île a une "spécialité" parmi : les graphismes, le son, l'ui et le game design. Pour chaque boss vaincu, Syntheros passera un cap de maîtrise de l'épée. Ce changement sera représenté par une synthétisation de la spécialité de l'île associée. Le contraste entre l'expérience avant et après la synthétisation permettra au joueur de voir, entendre, comprendre et jouer la maîtrise de l'épée de Syntheros (The Messenger).

Inspiration Sonores

Inspirations Surchargée

Le groupe ["Nero's day at Disneyland"](#) est une grande inspiration pour la partie sonore surchargée.

La bande originale du jeu [Drakengard](#) colle aussi à l'ambiance sonore visée.

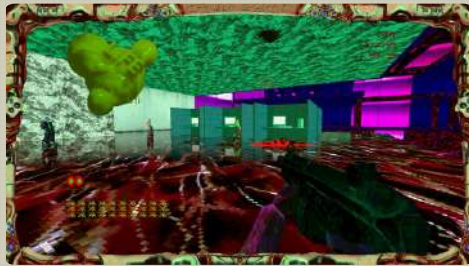
Inspirations Synthétique

Pour représenter le synthétisme, l'inspiration principale est le style ["ambient synthwave"](#).

La partie "ambient" permet au joueur de facilement différencier feedback et musique. De son côté, la partie "synthwave" donnerait un rendu rond et simple à l'oreille

Moodboard

Moodboard Surchargée



Moodboard Synthétique



Une problématique technique

L'objectif de Analsynth est de faire ressentir l'expérience de maîtrise, ainsi la problématique clé du jeu est de ne pas faire perdre sa maîtrise au joueur.

Pour cela il faut :

- Un système de combat bien segmenté, cela permettra de tester les différents composants soit séparément soit en groupe pour détecter leurs points forts et faibles.
- Un effort particulier sur les signes et feedback, ils doivent être clair, compréhensible et précis. Ils devront être facilement customisables et testables dans des environnements prédéfinis.
- Le dernier point de cette problématique est la caméra, en effet la précision de ses mouvements et son placement sont capital dans la lisibilité de ses actions et celles de ses adversaires.

Public Cible

Analsynth trouve son public cible majoritairement parmi les "mid-hard core gamers".

Le jeu demande, pour les combats, une certaine dextérité ; ce qui exclue le public dit "casual".

De plus la compréhension du thème du jeu se fait en contraste avec des expériences de jeu passé du joueur, cela demande une culture minimum du médium et exclu une majorité des "mid-core gamers".

Le public cible ne tombe pas dans celui des "hardcore gamers", car la difficulté ne se veut pas trop importante (perte légère de progression après une mort) et la grammaire vidéoludique nécessaire n'est ni trop précise ni trop importante.

Réalisation d'une IA de Monopoly

Description Problématique

La problématique que j'ai choisi est la création d'une IA joueuse pour le jeu "Monopoly".

Au vu des multiples couches complexe du Monopoly, je compte spécialiser mon IA sur la partie achat/vente des propriétés et bâtiments; les cases chances, caisse communautées et prison étant plutôt séparé du reste du jeu ne seront donc pas pris en compte.



La résolution de ce problème se fera en plusieurs étapes : définition des règles utilisées, définition des termes utilisés, objectifs de l'IA, ... et une ouverture vers d'autres pistes d'exploration.

Bibliographie

Règles du Monopoly par *Monopolypedia.fr*

Ce site m'a permis d'avoir un accès facile aux nombreuses règles du Monopoly. Cela m'a permis de partir d'une base solide pour construire mon IA.

Gagnez au Monopoly à tous les coups ! par *Zonebourse*

Cette vidéo couvre les bases des stratégies efficace surlesquels j'ai basé mon IA.

Rendez-vous avenue Henri-Martin ou comment gagner au Monopoly grâce aux chaînes de Markov par *Sébastien Ferenczi, Rémy Jaudun et Brigitte Mossé*

Cette article expose les résultats de recherche quand à la fréquentation des différentes cases de monopoly. Ces données me seront fortement utiles pour calculer les gains et risques lors de la création de mon IA (image plus bas).

Description de la solution

Définition des règles utilisées

Le but du Monopoly est d'être le dernier joueur n'ayant pas fait **faillite**. Tous les joueurs commencent sur la **case départ** avec **1500M**.

Le tour de chaque joueur commence par un lancer de **2 dés**, le joueur avance d'autant de cases que le nombre indiqué par les dés. S'il a fait un **double**, il **rejoue** immédiatement après la fin de son tour.

Lorsque le joueur arrive sur une **case propriété n'appartenant à personne** il peut **l'acheter** pour le prix indiqué sur le plateau.

Lorsque le joueur arrive sur une **case propriété appartenant à un autre joueur**, il doit lui payer un **loyer** dépendant du loyer associée à la case.

Lorsque le joueur arrive sur une **case propriété lui appartenant**, rien ne se passe.

Lorsque le joueur arrive sur une **case "Impôts - Taxe de Luxe"** il devra **payer** le montant indiqué (similaire à une case propriété ne lui appartenant pas).

Si un joueur ne peut pas **payer une dette** il se voit obliger de **vendre à la banque** (au prix original) ses propriétés jusqu'à ce qu'il puisse rembourser l'intégralité (sinon il fait faillite).

Les enchères, l'hypothèque, les constructions (maisons et hôtels), les cases caisses de communauté et chance ne seront **pas pris en compte** par l'IA par mesure de **simplicité**. Je souhaite construire une IA efficace sur les **règles les plus importantes** du jeu, les autres couches pourront être rajoutée ensuite pour complexifier le jeu.

Définition des termes utilisés

On définit :

- **"Le joueur"** réfère au joueur en train de jouer son tour.
- **T** une case du plateau ; **T_A** la case du plateau sur laquelle le joueur se trouve ; **T_B** une case du plateau statistiquement moyenne.
- **C(T)** le coût d'une case **T** à l'achat ; **C_C(T)** le coût contextuel d'une case **T** (en fonction de la position du joueur par rapport à la case).
- **L(T)** le loyer d'une case.
- **R(T)** le rendement d'une case.
- **P(X)** la probabilité d'un événement **X**.
- **D** le score d'un lancer de dé quelconque ; **D_T** obtenir un lancer de dés permettant d'aller sur la case **T**.
- **Z** l'argent du joueur ; **Z_C** l'argent contextuel du joueur (son argent + l'apport potentiel des cases lui appartenant)
- **A(T_n)** l'apport contextuel de la n-ième case appartenant au joueur ; **A_p(T_n)** l'apport contextuel de la n-ième case appartenant au joueur par rapport au p-ième autre joueur de la partie.
- **M(T)** la dette associée à une case non achetable ; **M_C(T)** la dette associée contextuel (prends en compte la position de la case par rapport au joueur)
- **Δ** une contante associée à l'IA représentant son envie de parier sur sa chance, **Δ ≥ 0**.
- **β** une contante associée à l'IA représentant sa confiance en ses propriétés (en l'argent qu'elles pourraient rapporter), **β ≥ 0**.

Objectifs de l'IA

L'objectif principal de l'IA est de ne pas faire **faillite**.

Chaque tour elle devra choisir **d'acheter ou non** une propriété basé sur les nombreuses informations à sa disposition.

Deux **paramètres** permettront un certain **contrôle** dessus pour lui associer des **comportements différents** si besoin (niveaux de difficultés, personnalités de l'IA, QA Test).

Calcul des données

Le **coût** et **loyer** des différentes cases sont indiqués ici :

| Couleur | Titres de propriété | Prix d'achat | Loyer |
|------------|------------------------|--------------|-------|
| Marron | Rue Lecourbe | 60M | 4M |
| | Boulevard de Bellevil | 60M | 2M |
| Bleu ciel | Avenue de la Républi | 120M | 8M |
| | Rue de Courcelles | 100M | 6M |
| | Rue de Vaugirard | 100M | 6M |
| Rose | Rue de Paradis | 160M | 12M |
| | Avenue de Neuilly | 140M | 10M |
| | Boulevard de la Villet | 140M | 10M |
| Orange | Place Pigalle | 200M | 16M |
| | Boulevard Saint-Mich | 180M | 14M |
| | Avenue Mozart | 180M | 14M |
| Rouge | Avenue Henri-Martin | 240M | 20M |
| | Boulevard Malesherb | 220M | 18M |
| | Avenue Matignon | 220M | 18M |
| Jaune | Rue La Fayette | 280M | 24M |
| | Place de la Bourse | 260M | 22M |
| | Faubourg Saint-Honc | 260M | 22M |
| Vert | Boulevard des Capuc | 320M | 28M |
| | Avenue Foch | 300M | 26M |
| | Avenue de Breteuil | 300M | 26M |
| Bleu foncé | Rue de la Paix | 400M | 50M |
| | Avenue Champs-Élys | 350M | 35M |

La **probabilité** de faire un certain score avec un **lancer de 2 dés** est :

| Score | Probabilité |
|-------|-------------|
| 2 | 1/36 |
| 3 | 2/36 |
| 4 | 3/36 |
| 5 | 4/36 |
| 6 | 5/36 |
| 7 | 6/36 |
| 8 | 5/36 |
| 9 | 4/36 |
| 10 | 3/36 |
| 11 | 2/36 |
| 12 | 1/36 |

La **probabilité** d'arriver sur les différentes **cases** est indiqué ici :

Long Term Probabilities for Ending Up on Each of the Squares in Monopoly®

| Square | Probability % (Jail Short) | Rank | Probability % (Jail Long) | Rank |
|-----------------------|----------------------------|------|---------------------------|------|
| Go | 3.0961 | 3 | 2.9143 | 3 |
| Mediterranean Avenue | 2.1314 | 36 | 2.0073 | 36 |
| Community Chest | 1.8849 | 37 | 1.7750 | 37 |
| Baltic Avenue | 2.1624 | 35 | 2.0369 | 35 |
| Income Tax | 2.3285 | 28 | 2.1934 | 27 |
| Reading Railroad | 2.9631 | 6 | 2.8010 | 8 |
| Oriental Avenue | 2.2621 | 32 | 2.1317 | 32 |
| Chance | 0.8650 | 40 | 0.8152 | 40 |
| Vermont Avenue | 2.3210 | 29 | 2.1874 | 28 |
| Connecticut Avenue | 2.3003 | 30 | 2.1680 | 30 |
| Visiting Jail | 2.2695 | 31 | 2.1392 | 31 |
| St. Charles Place | 2.7017 | 15 | 2.5560 | 15 |
| Electric Company | 2.6040 | 20 | 2.6140 | 13 |
| States Avenue | 2.3721 | 26 | 2.1741 | 29 |
| Virginia Avenue | 2.4649 | 24 | 2.4255 | 22 |
| Pennsylvania Railroad | 2.9200 | 8 | 2.6354 | 11 |
| St. James Place | 2.7924 | 12 | 2.6802 | 9 |
| Community Chest | 2.5945 | 21 | 2.2957 | 24 |
| Tennessee Avenue | 2.9356 | 7 | 2.8210 | 6 |
| New York Avenue | 3.0852 | 4 | 2.8118 | 7 |
| Free Parking | 2.8836 | 9 | 2.8253 | 5 |
| Kentucky Avenue | 2.8358 | 10 | 2.6143 | 12 |
| Chance | 1.0480 | 38 | 1.0448 | 38 |
| Indiana Avenue | 2.7357 | 13 | 2.5671 | 14 |
| Illinois Avenue | 3.1858 | 2 | 2.9929 | 2 |
| B & O Railroad | 3.0659 | 5 | 2.8930 | 4 |
| Atlantic Avenue | 2.7072 | 14 | 2.5370 | 16 |
| Ventnor Avenue | 2.6789 | 16 | 2.5191 | 18 |
| Water Works | 2.8074 | 11 | 2.6507 | 10 |
| Marvin Gardens | 2.5861 | 22 | 2.4381 | 21 |
| Go To Jail | 0.0000 | 41 | 0.0000 | 41 |
| Pacific Avenue | 2.6774 | 17 | 2.5236 | 17 |
| North Carolina Avenue | 2.6252 | 19 | 2.4721 | 20 |
| Community Chest | 2.3661 | 27 | 2.2276 | 26 |
| Pennsylvania Avenue | 2.5006 | 23 | 2.3531 | 23 |
| Short Line | 2.4326 | 25 | 2.2906 | 25 |
| Chance | 0.8669 | 39 | 0.8158 | 39 |
| Park Place | 2.1864 | 33 | 2.0595 | 33 |
| Luxury Tax | 2.1799 | 34 | 2.0521 | 34 |
| Boardwalk | 2.6260 | 18 | 2.4832 | 19 |
| In Jail | 3.9499 | 1 | 9.4569 | 1 |

Pour le calcul de l'argent contextuel :

$$Z_C = Z + A(T)$$

Pour calculer A(T) on fait la somme des apports contextuels des cases appartenant au joueur, pour tous les autres joueurs :

$$A(T) = [\sum_{p=0} (\sum_{n=0} (A_p(T_n)))] * \beta$$

On calcule l'apport potentiel de la n-ième case par rapport au p-ième joueur à partir de son loyer et de la probabilité qu'il arrive sur celle-ci :

$$A_p(T_n) = L(T_n) * P(D_T)$$

Pour le calcul de l'intérêt actuel :

$$I(T) = [P(T) / P(T_B)] * [R(T) / R(T_B)]$$

On a P(T) en lisant les tableaux au dessus et P(T_B) en faisant la moyenne des résultats au dessus et on calcule R(T) avec :

$$R(T) = L(T)/C(T)$$

De même pour L(T) et C(T) que l'on peut calculer avec les valeurs fournies au dessus.

On calcule l'intérêt contextuel des cases comme ceci :

$$I_C(T) = P(D_T) * [I(T) / I(T_A)]$$

Enfin, pour calculer la dette associée contextuelle :

$$M_C(T) = P(D_T) * M(T)$$

On calcule la dette associée :

$$M(T) = [Z_C - L(T)] / Z_C$$

Prise de décision

Le joueur commence son tour par **lancer les dés** puis **se déplacer** du nombre de cases demandé.

Une fois arrivé sur la case s'il doit simplement **payer**, il fournit l'argent nécessaire.

S'il lui **manque de l'argent** il **vend à la banque** les cases lui appartenant en commençant par celles dont l'**intérêt actuel I(T) est le plus faible** jusqu'à pouvoir rembourser sa dette.

Si la case sur laquelle il est arrivé **n'appartient à personne** et vaut **plus que l'argent actuel Z**, il ne **l'achète pas** et finit son tour.

Dans le cas où la case **n'appartient à personne** et vaut **moins que l'argent actuel Z** :

Tout d'abord, on calcule **Z_C** l'argent contextuel.

On parcourt ensuite la liste des **cases achetables** sur lesquelles le joueur pourra arriver au **tour suivant** et on garde dans une liste l'ensemble des cases où **C(T) ≤ Z_C** et où **C(T) + C(T_A) > Z_C**.

Cela nous permet **d'exclure de notre réflexion** les cases achetables dans le cas où :

- Le joueur est sûr de **ne pas pouvoir les acheter**.
- Le joueur est sûr de **pouvoir les acheter** même s'il **achète la case actuelle**.

L'étape suivante est de calculer pour tous les **cases achetables restants** leur **intérêt I(T)** pour ensuite trouver leur **intérêt contextuel I_C(T)**.

Pour l'ensemble des cases **non achetables** ayant un **loyer L(T)**, on calcule leur **dette associée**, puis leur **dette associée contextuelle**.

Pour finir si **∑ (I_C) + ∑ (M_C) > Δ** le joueur **n'achète pas**, sinon il **achète**.

Ouverture

L'IA ainsi créée présente l'avantage de proposer une **base solide** sur laquelle travailler pour **l'enrichir**. Elle reste malheureusement **très limitée** et mériterait du travail supplémentaire pour implémenter des réactions au **reste des règles** du Monopoly.

Les **deux paramètres** présents permettent à l'IA d'être **customisable**, ajouter d'autres paramètres permettant une **plus grande liberté de contrôle** serait intéressant.

Pour finir, une IA joueur de Monopoly pourrait être construite grâce à du **machine learning**, une surveillance particulière de l'IA serait nécessaire pour **laisser la place à l'erreur** dans ses raisonnements.

De plus si un **Game Designer** en a besoin il faudra pouvoir **expliquer son comportement** et le **moduler** au besoin.