

L3Games

Réviser et apprendre avec des jeux

Projet transverse L3

Groupe 132

Promo 2023

Tuteur : Ange KOUAME

Tom CHANG
Karis GWET
Jules RANGE
Tom GAULTIER
Houssam OUMOULOUD

Sommaire

I.	Introduction	3
II.	Le choix des jeux	4
1.	Système Linéaire	4
2.	Théorie du signal	5
3.	Cryptographie	6
4.	Théorie des graphes	7
5.	Web	8
III.	Réalisation de l'application	9
1.	Moteur de jeu Solar2D (CoronaSDK)	9
2.	Sauvegarde (Flask & MySQL)	9
3.	Organisation	10
IV.	Conclusion	11

I. Introduction

Apprendre ou réviser tout en jouant n'a pas la réputation d'être efficace hors de l'école élémentaire. Effectivement, jouer possède cette image de distraction ce qui entache souvent l'idée d'apprendre ou réviser efficacement sur des matières complexes. Pourtant, cela n'a pas empêché les nombreuses tentatives de corrélation entre ces deux dimensions : il suffit de naviguer sur le web quelques secondes pour trouver des jeux « sérieux ». Notons l'université de Strasbourg qui emploie cette méthode sur un module pour évaluer en permanence ses étudiants.

N'ayant pas eu l'occasion de voir une analogie à l'EFREI, nous avons donc saisi cette occasion pour pouvoir réaliser un projet similaire qui puisse répondre à cette problématique :

Comment apprendre et réviser tout en s'amusant ?

Pour répondre à ce problème, nous avons fait le choix de créer une application proposant divers jeux dans le thème des matières de L3 de l'EFREI. Le support visé est bien évidemment l'appareil qu'un grand monde utilise facilement et partout, c'est-à-dire le smartphone et plus précisément Android. Notre but est de créer un jeu par matière qu'on aura sélectionné au préalable grâce à un sondage. Notre effectif est limité à 5 personnes, nous allons donc réaliser 5 jeux, un jeu par matière qui a de la demande. Pour intéresser les étudiants, nous avons décidé de reprendre des genres de jeux connus que nous aborderons plus tard.

Afin de réaliser cette application, ce n'est pas directement en Android que nous allons coder, mais nous allons nous servir d'un moteur de jeu : Solar2D. Cependant, nous n'allons pas nous limiter à la création des jeux, nous avons aussi décidé d'enregistrer la progression d'un utilisateur grâce à une base de données, ce qui implique un système de comptes.

Tout d'abord, il serait intéressant de voir comment nous avons abouti à la réalisation de ce projet au niveau des concepts pour chaque matière traitée, puis de présenter les outils qu'on a utilisé afin de réaliser ce projet, ainsi qu'un bilan sur l'organisation du projet.

II. Le choix des jeux

1. Théorie du signal (Tom GAULTIER)

En ce qui concerne l'enseignement de théorie du signal, nous nous sommes concentrés sur l'usage des diagrammes de Bode. C'est un outil très utilisé dans les différents domaines de transmissions d'informations. Les étudiants doivent être capables de reconnaître et comprendre une fonction en visualisant un diagramme de Bode. Pour cela, ils doivent s'aider des bases que sont les fonctions caractéristiques qui permettent de former celles plus complexes.

En physique la pratique est essentielle pour comprendre, il faudrait donc que l'étudiant s'entraîne à dessiner ces graphes ainsi qu'à les reconnaître. C'est pourquoi nous nous sommes basés sur une pédagogie expérimentale. Nous avons donc décidé d'élaborer un quiz, inspiré de la méthode des évaluation Moodle. Mettant en avant des questions de difficultés variantes pour permettre d'une part une progression et d'autre part une validation des connaissances.

Lorsque vous entrez dans ce jeu, vous arrivez sur un menu comportant trois sélections. Les deux modes de jeux reprennent les deux parties du diagramme de Bode, le graphe d'amplitude et celui de la phase. La section d'aide quant à elle permet de se renseigner sur les fonctions caractéristiques sans avoir à retourner dans le cours. Enfin, le meilleur score du joueur s'affiche pour chacun des modes, ce qui permet d'évaluer son niveau. Ce score nous montre ici que le joueur a atteint le score maximum sur Phase Mode, colorisation en rose.

La partie aide comporte les cinq fonctions caractéristiques $H_1(j\omega)$, $H_2(j\omega)$, $H_3(j\omega)$, $H_4(j\omega)$, $H_5(j\omega)$. Pour présenter au joueur, des valeurs ont été données à $\tau = 10$ et $k = 10$ pour qu'il puisse visualiser la pulsation de coupure. Cliquer sur l'une des fonctions vous ouvrira donc une fenêtre avec les informations correspondantes à celle-ci.



Th du signal – menu principal

Dans les onze fonctions que comportent le quiz, nous avons l'essentiel des fusions de fonctions caractéristiques. Lorsque le joueur est capable de répondre à l'ensemble de celles-ci, il possède les bases sur l'ensemble des sujets se rapportant aux transmissions de signaux. Lorsque vous lancez une partie, une fonction apparaît en haut de l'écran. L'ordre d'apparition est aléatoire et unique pour éviter de répondre sans réflexion. Vous aurez donc une question sous forme de fonction avec quatre graphes de réponses. Seulement l'un d'eux est évidemment le bon, il suffit de cliquer dessus pour passer à la prochaine question et incrémenter le score. Attention car certaines réponses se ressemblent, il faut bien lire les valeurs d'abscisses et d'ordonnées aux endroits clés. En cas d'échec, les images se bloquent et une fenêtre apparaît donnant la bonne réponse à la question. Ce passage permet aux étudiants de prendre du temps pour comprendre leurs erreurs. Et en cas de non-compréhension, ils peuvent toujours retourner sur la section d'aide.

2. Théorie des graphes (Houssam OUMOULOUD)

Pour la matière Théorie des graphes, **Froggy** permettra de consolider un minimum les connaissances sur la matière via une approche plus orientée ludique.

Pour cela, il a fallu d'abord imaginer le concept du jeu en réfléchissant à l'action et aux différentes étapes qu'il faudrait préparer en prenant inspiration sur les exercices que l'on a effectué en TD. Le but du jeu est de tester ses connaissances sur la théorie des graphes à travers des questions sur la matière. Un concept de jeu simple, où il ne s'agira que de déplacer le personnage de **Froggy** (une grenouille) qui doit à chaque niveau répondre à une question en choisissant le parcours vers la bonne réponse.

Pour la réalisation, il a été nécessaire de commencer par poser sur papier les croquis du menu et les différentes scènes du jeu afin de visualiser les étapes de création ainsi que les questions pour chaque niveau et les différentes propositions de réponses.

En ce qui concerne l'interface, des difficultés se sont présentées en termes d'originalité. La scène d'accueil le montre bien, ce qui est dommage. La texture des différentes cartes du jeu a été réalisée avec le logiciel Gimp à partir de différentes images (le sol, l'eau etc.), des textures qui vont dans le thème de la grenouille. Ensuite, la création des niveaux aura nécessité l'utilisation de plusieurs matrices aléatoires.

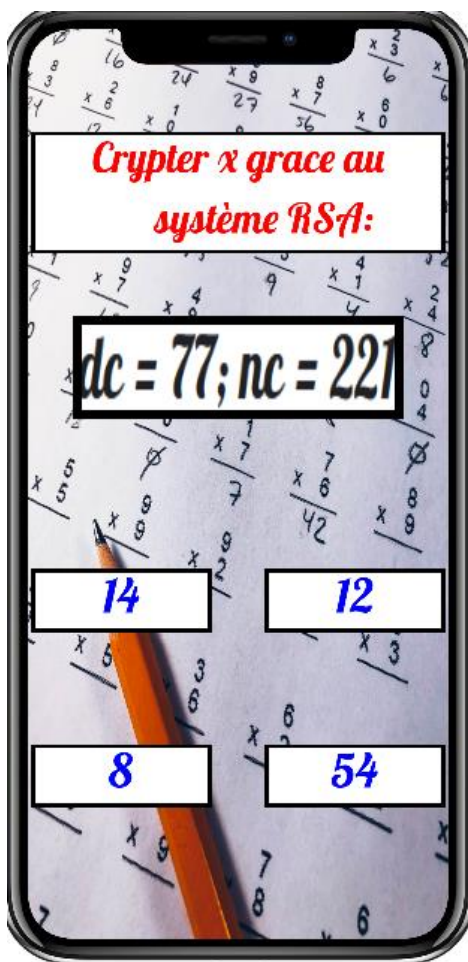


Niveau 1 Froggy

Comme dans le vrai jeu, **Froggy** doit sauter de nénuphar en nénuphar afin de rejoindre celui correspondant à la bonne réponse de la question du niveau. Il suffira d'appuyer sur les nénuphars orange pour connaître leur proposition de réponses. Au cas où il se dirigerait vers une mauvaise réponse, un simple message l'avertira, ce qui remettra en question ses connaissances. Le joueur pourra donc réessayer d'autres réponses afin de pouvoir passer au niveau supérieur.

3. Cryptographie (Jules RANGE)

Les mathématiques pour la cryptographie est une nouvelle matière abordée pendant le S5 qui regroupe de nombreuses techniques différentes de cryptage et déchiffrement et qui peuvent s'avérer essentiel par la suite dans la programmation. Le but du jeu est donc d'avoir une occasion de tester ses connaissances ou d'en apprendre de nouvelles à travers un quizz ludique regroupant différents niveaux portant tous sur des thématiques différentes de la cryptographie.



Cryptographie - Jeu



Cryptographie - Menu

Chaque niveau est plus dur que le précédent, et le jeu finit donc sur le système **RSA** et **El Gamal** qui sont les deux domaines centraux à la cryptographie en L3. Si l'utilisateur se connecte, il pourra alors sauvegarder sa progression et reprendre à la dernière question répondue. Le jeu est donc composé de plusieurs scènes, chacune représentant un niveau de difficulté, toutes liées entre elles par le Scene Manager de LUA. Le jeu se termine par un Game Over affichant alors le score final et une option pour recommencer les réponses erronées.

4. Système Linéaire (Karis GWET)



Dodgin – Menu principal

En ce qui concerne ce jeu, nous avons pour objectif de divertir l'utilisateur pour qu'il trouve de l'intérêt à apprendre sur cette plateforme la matière **Système Linéaire**, proposé à Efrei Paris, plutôt que dans ses cahiers ou sur un autre type de lieu.

C'est pour cela que nous avons implémenté **Dodgin**, un jeu solo player dont le principe est simple. Il consiste à esquiver des boules tombantes venant du ciel, à une certaine vitesse, et à atteindre un certain score, variant selon les niveaux, afin de faire apparaître une case verte.

Une fois la case verte prise, une case verte apparaîtra et permettra au joueur d'apprendre un thème sur la matière.

Le joueur découvrira donc au fur et à mesure les différentes leçons proposées, tout en s'amusant, ce qui est le **principal objectif** de notre application.

Il était important pour nous de nous centrer sur cet aspect étant donné que nous avons fait plusieurs sondages sur les effets néfastes du confinement et de la crise sanitaire exceptionnelle que nous avons rencontré cette année, et malheureusement les résultats étaient tous démoralisant. C'est pour cela que grâce à **Dodgin**, les élèves pourront continuer d'apprendre tout en s'amusant et même se comparer à d'autres élèves grâce à l'affichage des meilleurs scores.



Dodgin – Level 1



Dodgin – Leçon 1

5. Web (Tom CHANG)

Pour cette matière, le jeu se nommera « **WebPG** ». La raison est claire, c'est tout simplement car le jeu sera du genre **RPG** (soit **Jeu vidéo de rôle**) construit de A à Z. Ici, pour faire apprendre au joueur, il faudra contrôler un personnage qui sera capable d'interagir avec son environnement.



Web 1: Probe face au serveur et BDD

L'idée derrière ce jeu est que pour comprendre comment fonctionne le web avec VueJS, il faut voir comment les différents fichiers interagissent entre eux. Pour ça, quoi de mieux que de faire incarner chaque élément par des personnages ?

Dans ce jeu, on incarne un petit robot appelé « **probe** ». C'est le personnage que l'utilisateur va contrôler afin d'interagir avec les différentes entités comme le serveur ou la base de données par exemples.

Chaque entité explique son rôle et délivre des informations à l'utilisateur. La communication se fait grâce à des boîtes de dialogue où l'entité explique ses actions. Grâce à cela, nous pouvons avoir en visuel comment chaque composant (qui permettent à VueJS ou le web en général de fonctionner) communiquent.

Le joueur s'immergera dans 3 niveaux qui traiteront plusieurs cas d'interaction : une erreur, une authentification et un événement simple enclenché par un utilisateur. On commence par la plus facile, l'erreur, qui servira de complément au tutoriel précédent. A travers les différents niveaux, quelques mécaniques et de nouvelles entités apparaissent pour diversifier les interactions.

Afin de ne pas perdre le joueur, il y a aussi un système de quête que l'utilisateur a accès en permanence. Cela lui permet de le diriger au cas où il serait perdu.

Le joueur terminera un niveau seulement lorsqu'il aura respecté toutes les conditions nécessaires : suivre les instructions du guide pour les premiers niveaux, compléter toutes les missions pour les autres niveaux. En général, les missions sont majoritairement 'communiquer avec telles personnages' pour écouter ce qu'il a dire et ce qu'il nous ordonne de faire par la suite, ce qui va parfaitement dans la visualisation des interactions entre les différents composants.



Web 2: Probe interagi avec la BDD dans le tutoriel

III. Réalisation de l'application

1. Moteur de jeu CoronaSDK (Solar2D)

Après avoir présenté les concepts de nos jeux, présentons un peu les outils qui nous ont permis de créer notre projet. Pour commencer, le développement de nos jeux s'est fait avec le moteur de jeux **Solar2D**, précédemment appelé CoronaSDK. C'est un moteur de jeu codé en LUA, un langage de script. La raison pour laquelle nous avons utilisé ce moteur est qu'il est très flexible et très facile à prendre en main. Il nous a permis de rassembler aisément nos jeux grâce à la fonctionnalité de scènes. En rassemblant les jeux, il n'aura suffi que d'intégrer un menu principal dans lequel nous pouvons avoir accès à tous les jeux. Ici, nous donnons le choix à l'utilisateur de pouvoir jouer sans se connecter, ce qui est tout à fait normal. Dans ce cas, il n'aura pas la chance de voir sa progression sauvegardée.



Menu principal
(utilisateur non connecté)

Choix des différents jeux

2. Sauvegarde



L'inscription

Notre jeu est effectivement capable de sauvegarder la progression de l'utilisateur à condition qu'il soit connecté. Pour cela, il va falloir intégrer une connexion entre le jeu, le serveur et la base de données. Notre solution face à ce problème est **Flask**, *framework* pour le web capable de créer un serveur local très facilement d'une part car c'est en Python, mais aussi car c'est un outil plus récent qui a le mérite d'être essayé dans le cadre de notre apprentissage. Pour la base de données, ce sera sans surprise MySQL, où l'on enregistrera l'identifiant, l'email, le mot de passe (hashé), ainsi que ses données sur les jeux. Le fonctionnement entre ces trois éléments est simple : notre jeu effectue une **requête http** au serveur local sous Flask, et dépendant de la route prise qu'on a établi :

- enverra une réponse à notre jeu
- mettra à jour la base de données

Notre serveur étant connecté à la base de données, il nous est donc possible de demander à partir du jeu, des informations de la base de données.

3. Organisation

Intéressons-nous désormais aux différents succès et difficultés rencontrées lors de la réalisation de ce projet. Tout d'abord, au niveau de la planification du cahier des charges, les délais étaient approximativement respectés par tout le monde. On pense notamment à la maîtrise du langage LUA pour Solar2D, qui est certes simple pour l'un, mais qui a posé un problème d'adaptation pour d'autres car ce n'est pas un langage « habituel », surtout au niveau de la syntaxe par exemple, d'autant plus que certains avaient l'habitude de coder des jeux dans d'autres langages. Ce fut donc une surprise générale puisque nous nous attendions tous à coder sous Android uniquement.

Au niveau de l'organisation, le but était simple : chacun fait son jeu, et cela s'est avéré très utile puisque la participation de tous les membres devenait inévitable. On pourrait croire que le travail en équipe se perd sous cet angle individualiste. Cependant, la différence de niveau entre les membres se ressentait et cela se reflétait surtout sur les jeux. C'est grâce à ce problème que ce projet est passé d'un projet individuel entre chaque membre à un travail collectif efficace : séances de soutien par ceux qui sont à l'aise avec le langage, réunion pour faire des points sur l'avancement des jeux etc. Il n'y avait donc personne mis à l'écart et la flexibilité du projet fait que chaque personne pouvait avancer à son rythme tout en respectant les délais imposés.

La difficulté du projet s'est fait ressentir lorsque nous voulions établir un système de progression (qu'on rappelle non-obligatoire pour jouer). La question était comment allons-nous procéder pour relier une base de données (supposée) en ligne avec notre jeu sous Solar2D, en LUA ? La méthode serait-elle assez flexible pour qu'elle puisse mettre à jour à n'importe quel moment que l'on souhaite lorsque l'utilisateur joue ? Il a fallu étendre nos connaissances acquis en Web et les appliquer dans ce cas avec Flask car nous avons découvert que Solar2D (LUA) possède sa propre bibliothèque intégrée pour gérer les requêtes HTTP. La difficulté était non pas de l'appliquer mais de trouver cette solution car il y avait effectivement un bon nombre de solutions possibles mais qui ne répondaient pas à nos attentes (apparition de latence, ralentissement du jeu, implémentation catastrophique etc.).

IV. Conclusion

Pour conclure ce projet, il est indéniable que cela a été une expérience enrichissante non-seulement au niveau technique grâce aux différents outils que nous avons maîtrisé (Lua, Solar2D, Flask etc.) mais aussi dans la gestion efficace d'un projet mené par un chef de projet et des membres organisés et passionnés par le sujet. Toutes ces expériences nous seront très bénéfiques pour la suite. Nous avons réussi à faire ce que nous avons prévu de réaliser dans le cahier des charges mais le projet pourrait aller encore plus loin car en effet, nous n'avons traité que 5 matières sur tout ce que l'EFREI propose durant cette année de L3. Nous sommes aussi très fiers d'avoir pu proposer une tentative de réponse sur le thème de jouer en apprenant car nous trouvons que cette problématique n'est pas souvent abordée très sérieusement alors que le potentiel existe.