



DEPARTEMENT INFORMATIQUE
Institut Universitaire de Technologie

8, rue Montaigne
56000 Vannes cedex

DESORMEAUX—DELAUNAY Tom
COSNIER Quentin
LE BORGNE Antoine
CHOLLET Quentin

Année universitaire 2020-2021

Module 3202
Projet de Synthèse
Spécification premier sprint

Projet : Game Of Math

Tuteur : M.Xavier ROIRAND – xavier.roirand@univ-ubs.fr

Client : M.Frederic LeMellec – flemellec911@gmail.com



Table des matières

Introduction	2
Planning.....	2
Objectif.....	2
Les cas d'utilisations.....	3
Diagramme de séquence élève.....	3
Diagramme de classe de la page de connexion	4
Les différentes maquettes	4
Détail des technologies utilisées.....	6
React JS	6
WebGL.....	6
Node JS.....	6
SQLite	7

Introduction

Ce document s'adresse à notre maître d'œuvre de projet M. ROIRAND, notre client M. LEMELLE ainsi qu'à l'équipe de développement du projet Game of Math.

Les spécifications du premier sprint seront définies ici, et auront pour but de décrire et d'informer sur les fonctionnalités et les éléments de l'application qui seront présents à la fin de ce sprint. Ce premier sprint se concentrera sur le fonctionnement de l'application côté étudiant. C'est-à-dire le développement du login, des QCM et une ébauche de l'interface. Dans ce document, les fonctionnalités primordiales seront illustrées ainsi que les fonctionnalités facultatives grâce à des maquettes graphiques et différents diagrammes.

Planning

Le sprint débutera le 28/11/2020

La fin du sprint et le premier rendu aura lieu le 18/12/2020

Objectif

L'objectif de ce premier sprint est de mettre en œuvre plusieurs interfaces et devront être fonctionnelles. Les interfaces que nous allons développer seront :

- La page de connexion
- La carte interactive (Menu principale)
- La page regroupant les différents chapitres de cours (Quizz)
- La page des quizz

Côté serveur nous allons aussi concevoir et intégrer la base de données ainsi que la gestion des pages citées précédemment.

Les cas d'utilisations

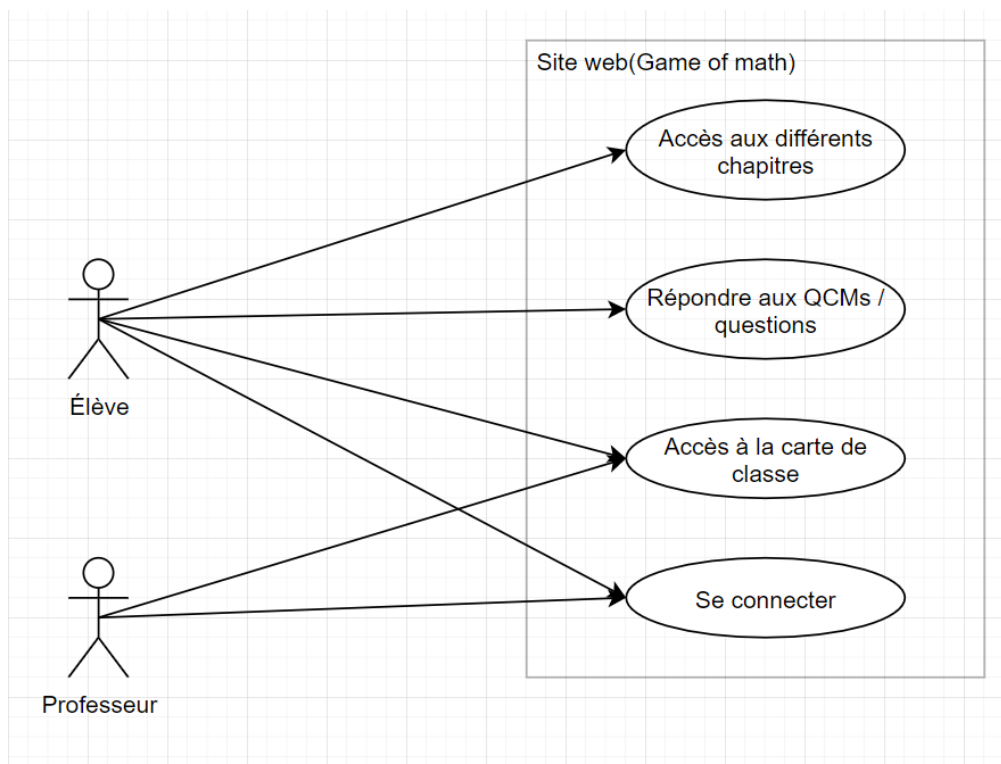


Diagramme de séquence élève

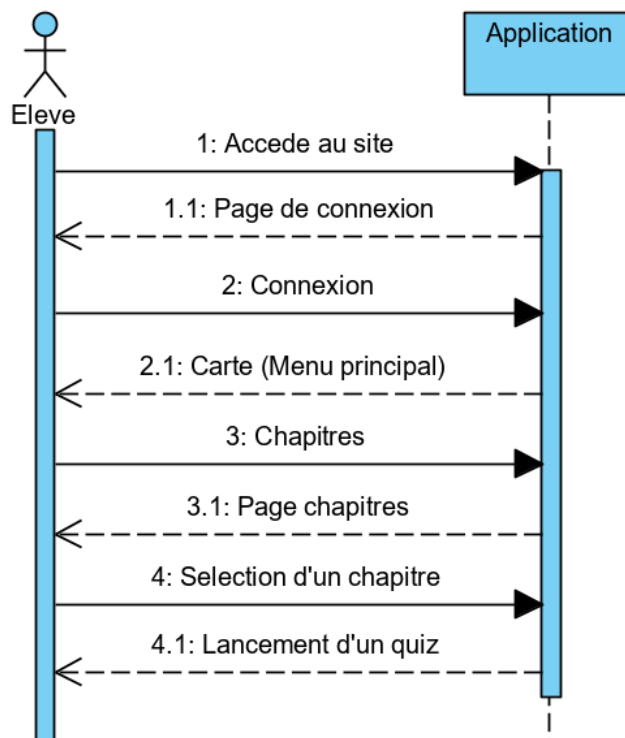
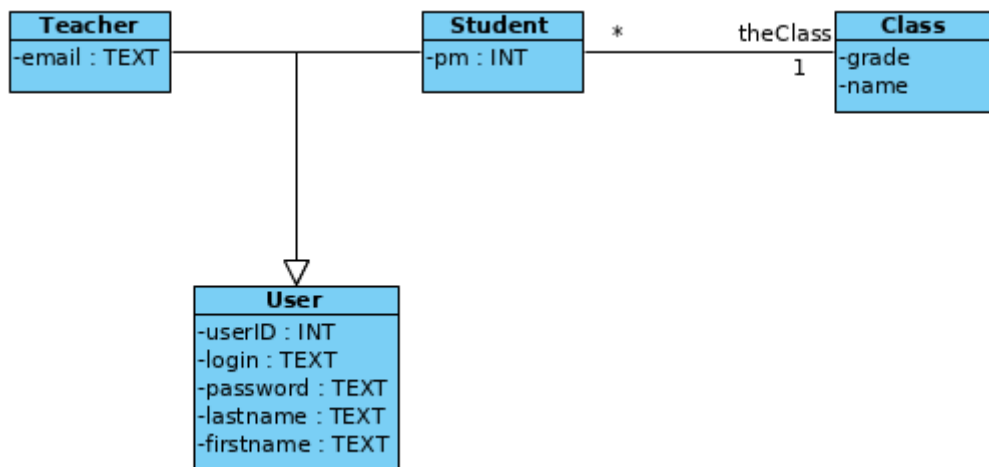
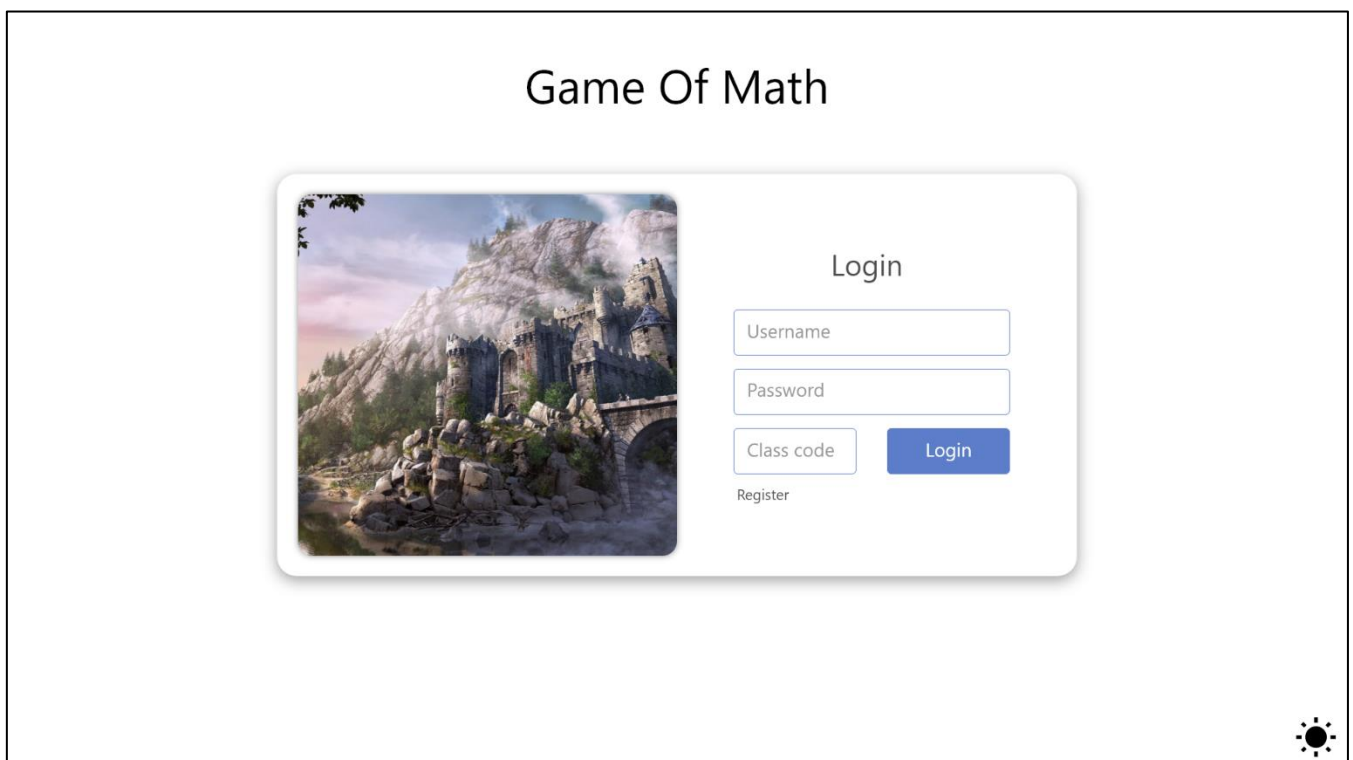


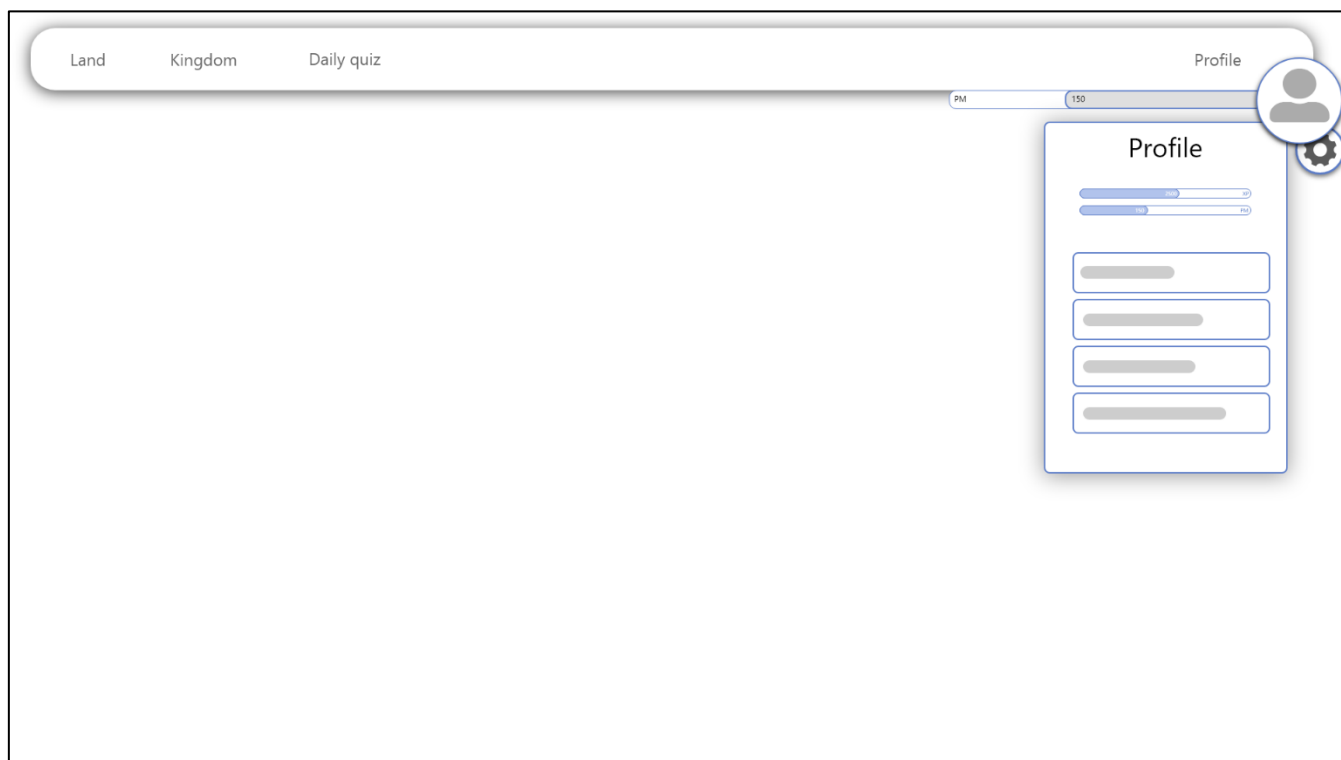
Diagramme de classe de la page de connexion



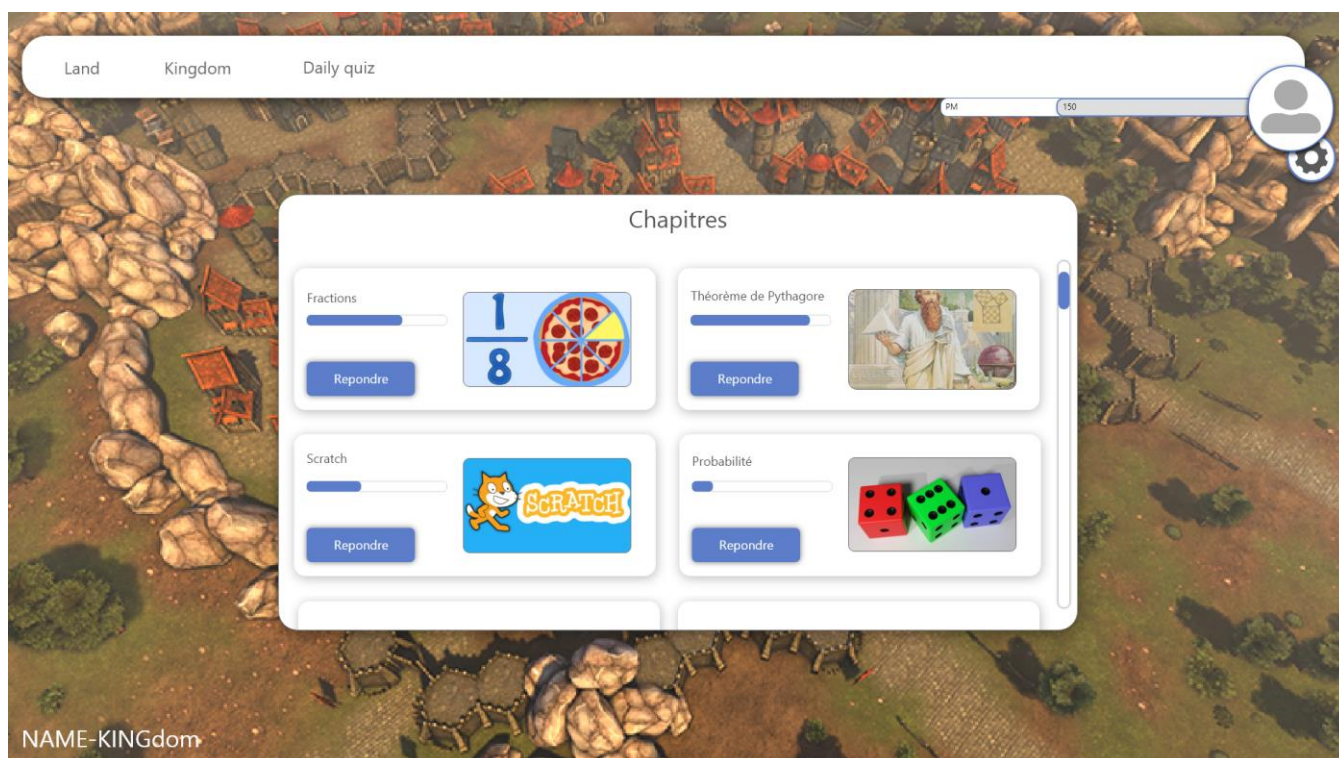
Les différentes maquettes



Exemple de rendu de la page de connexion



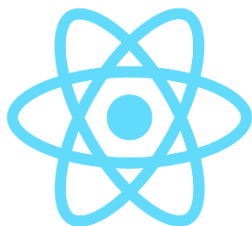
Exemple de rendu de la page principale du jeu



Exemple de rendu du menu des différents chapitres

Détail des technologies utilisées

React JS



La rapidité du rendu fait partie des fonctionnalités-phares de React et lui donne un avantage certain sur Angular. Cette technologie embarque des méthodes de réduction du nombre d'opérations sur le DOM, ainsi que d'optimisation et d'accélération du processus de mise à jour. Le DOM (Document Object Model) est d'une grande utilité en cas de gestion de grosses bases de données. JSX est une syntaxe JS qui autorise les citations en HTML et l'utilisation du balisage HTML lors du rendu des sous-composants. Il favorise l'élaboration d'un code lisible par la machine et offre la possibilité de constituer des composants en un seul fichier vérifié au moment de la compilation. C'est pour cela que nous n'utiliserons pas Angular ou Vue JS qui sont plus compliquer à prendre en main et utilisé pour des plus grosses productions.

WebGL

WebGL est l'unique solution disponible pour effectuer le rendu d'images 3d directement côté client. L'implémentation disponible aujourd'hui dans les navigateurs s'appuie sur Javascript afin de passer les données au pilote WebGL du client. Il représente un large support sur la majorité des plateformes, ce qui en fait un bon choix malgré la difficulté d'approche de l'interface de programmation. Pour le projet, l'utilisation de WebGL est expliquée par le fait que cela permet d'effectuer un rendu simple en début de projet avec des évolutions très poussées possibles grâce aux possibilités de l'interface. Afin de simplifier l'utilisation de l'interface et d'éviter d'avoir à recommencer une grande partie du code inutile, le choix d'utiliser l'API Three.js a été faite.



Node JS

Pour créer cette application web nous utiliserons Node JS accompagné d'Express et Socket IO. L'autre possibilité aurait été PHP. Nous n'utiliserons pas cette technologie, en effet pour notre utilisation il n'est pas adapté. De plus en PHP le serveur, est un serveur Apache, or nous avons besoin d'un grand contrôle sur cette partie. Node JS est alors plus adapté pour nous.



SQLite

Nous utiliserons pour ce projet SQLite. Nous aurions pu utiliser MySQL qui offre énormément d'avantage pour des grosses base de données comme la gestion de multiple connexion. Cependant notre base de données n'a pas besoin d'être accessible par plusieurs applications et ne sera pas très lourd. Les avantages de SQLite sont, qu'il est léger, et nomade, il peut être déplacé et copier facilement, ce qui facilitera le développement.

