

ÉPREUVE SYNTHÈSE

MISE EN SITUATION

Pour l'épreuve synthèse du cours 420-5A6-JR – Développement Web, vous devez réaliser une application distribuée. Vous allez devoir réaliser un serveur de services Web, une application mobile et un site Web. L'épreuve synthèse sera découpée en deux parties. La première consistera en la rédaction d'un dossier d'architecture et la seconde aux développements des applications.

MANDAT

En équipe de 5 ou 6 personnes, vous devez tout d'abord réaliser le document d'architecture qui vous permettra de mieux comprendre les systèmes à réaliser et les interactions entre eux. Par la suite, vous serez appelé à développer l'application distribuée (deux clients et un serveur) et les interactions entre eux.

REMISE

PARTIE 1

Vous devez remettre votre dossier d'architecture dans LÉA (une remise par équipe) **avant 23 h 59 le vendredi 1 décembre 2017**.

PARTIE 2

Les deux applications clientes et le serveur devront être remis dans un dossier de remise du cours (une remise par équipe) **avant 23 h 59 le mercredi 20 décembre 2017**.

Application Fureteur	Application Mobile	Serveur
Vous devez remettre le code source de votre application fureteur.	Vous devez remettre le projet Android Studio et vous assurer que votre application est fonctionnelle sur un appareil Nexus 5 utilisé lors du cours.	Vous devez remettre le code source de votre serveur. Vous devez partager avec l'enseignant votre projet Cloud9.io, pour que l'enseignant soit en mesure de redémarrer votre serveur au besoin.

VOCABULAIRE

EXPLORATEUR

L'*Explorateur* est le citoyen d'Andromia voyageant de *Portal* en *Portal* pour collectionner des *Runes* et des *Units*. Il existe d'autres moyens de transport sur Andromia, mais seuls les Explorateurs peuvent utiliser les *Portals*. Un explorateur possède au minimum : un nom d'explorateur (Personnage), un nom d'utilisateur et un mot de passe.

LOCATION

Dans le monde d'Andromia, les *Locations* sont les villes, les villages, les points d'intérêt. Il est possible de voyager de *Location* en *Location* via des *Portals*. On reconnaît une *Location* à son nom.

PORTAL

Les *Portals* sont une façon de voyager sur Andromia, lorsqu'un explorateur scanne la clé d'un Portal, il sera transporté dans une autre *Location*. Lors de son voyage, il pourra récupérer des *Runes* et débloquent une *Unit*. Un *Portal* est identifié de manière unique par sa clé.

UNIT

Les *Units* sont des unités qui voyagent elles aussi à travers les *Portals*, les explorateurs sont en mesure de les capturer en activant leur noyau à l'aide du bon nombre et du bon type de *Rune*. L'objectif de l'*Explorateur* est d'en débloquent le plus possible. On ne sait jamais, un jour, elles pourraient être utiles.

RUNE

Il existe 12 types de *Runes* dans le monde d'Andromia soit : Air, Darkness, Earth, Energy, Fire, Life, Light, Logic, Music, Space, Toxic et Water.

INOX

L'*Inox* est la principale devise sur Andromia, son utilisation est encore inconnue.

CONCEPT DE BASE

Dans le monde d'Andromia pour explorer un *Portal* vous devez à l'aide d'un appareil mobile numériser un code QR. La clé contenue dans le code vous permet de voyager entre deux *Locations*. Ce voyage pourrait vous permettre de récupérer des *Runes* et peut-être même de capturer une *Unit*. Pour activer un noyau et capturer une *Unit*, vous devez mettre le bon nombre de chacun des types de *Runes*, une fois le noyau activé, cette *Unit* sera ajoutée à votre collection.

PARTIE 1 – DOSSIER D'ARCHITECTURE

La première étape est de réaliser un dossier d'architecture. Ce dossier expliquera les services disponibles sur votre serveur et ensuite expliquer les interactions entre le serveur et les clients.

Vous pouvez vous inspirer du modèle vu en classe pour votre dossier. Voici les sections obligatoires dans votre dossier d'architecture

- Liste des récits utilisateurs avec leur vélocité
- Interface de communication
- Serveur
 - Diagramme de la base de données
 - Routes disponibles (requête et réponse)
- Applications clientes
 - Services appelés
 - Lien entre les vues et le récit utilisateur

PARTIE 2 – APPLICATION DISTRIBUÉE

LE SERVEUR CENTRAL D'ANDROMIA

Une version beta du serveur est disponible à l'adresse suivante : <http://api.andromia.science>

SÉRIALISATION

<p>Exploration</p> <ul style="list-style-type: none"> • L'attribut runes est vide ou contient un objet Rune. • L'attribut unit est vide ou contient un objet Unit. 	<pre>{ "dateExploration": "2017-11-11 21:10:23", "destination": "N'Jast", "runes": { ... }, "unit": { ... } }</pre>
<p>Rune</p>	<pre>{ "air": 1, "darkness": 2, "earth": 0, "energy": 0, "fire": 1, "life": 1, "light": 0, "logic": 1, "music": 2, "space": 9, "toxic": 2, "water": 0 }</pre>
<p>Unit</p>	<pre>{ "name": "Yodarha", "life": 4, "speed": 4, "imageUrl": "http://420.cstj.qc.ca/ycharron/andromia/a2017/12.png", "affinity": "life", "runes": { "weapons": ["staff"], "abilities": ["life", "music"] }, "set": "A2017", "number": 12, "uuid": "NWUwZGUXODEtMDRlNy00ZjFmLWI2NDAtN2U1OGewYzlmOGRm", "kernel": {} }</pre>

Un exemple est disponible dans le répertoire de partage.

VOYAGER VIA UN PORTAL

Votre application mobile devra faire appel à ce service lorsqu'un code QR sera numérisé.

URL	/portals/{key}
Paramètres d'URL	key : la clé du portal, correspondant au code QR numérisé.
Corps de requête	Aucun
Réponse - Réussite	Code HTTP : OK – 200 - 4 scénarios possibles <ol style="list-style-type: none"> 1. Sérialisation JSON d'une exploration sans <i>rune</i> et sans <i>unit</i>. 2. Sérialisation JSON d'une exploration avec <i>rune</i> et sans <i>unit</i>. 3. Sérialisation JSON d'une exploration sans <i>rune</i> et avec <i>unit</i>. 4. Sérialisation JSON d'une exploration avec <i>rune</i> avec <i>unit</i>.
Réponse - Échec	Code HTTP: NOT FOUND – 404

APPLICATIONS CLIENTES

Vous devez développer deux applications clientes qui utiliseront les services Web développés et fourni. Une des applications sera basée Web tandis que l'autre sera une application Android. Voici les récits utilisateurs nécessaires. Vous pouvez inclure d'autres récits utilisateurs, suite à un accord avec l'enseignant.

RÔLES

- Citoyen : Utilisateur ne possédant pas de compte de jeu
- Explorateur : Utilisateur possédant un compte de jeu

APPLICATION FURETEUR

RÉCITS UTILISATEURS

No.	Récit utilisateur
F1	En tant que Citoyen , je veux pouvoir me créer un compte pour devenir Explorateur .
F2	En tant qu' Explorateur , je veux me connecter, pour utiliser les fonctionnalités de l'application.
F3	En tant qu' Explorateur connecté , je veux me déconnecter
F4	En tant qu' Explorateur connecté , je veux afficher ma collection de <i>Unit</i> .
F5	En tant qu' Explorateur connecté , je veux afficher les détails d'une <i>Unit</i> de ma collection
F6	En tant qu' Explorateur connecté , je veux voir mon inventaire de Runes.
F7	En tant qu' Explorateur connecté , je veux voir mon total d'Inox et connaître ma location

APPLICATION MOBILE

RÉCITS UTILISATEURS

No.	Récit utilisateur
M1	En tant que Citoyen , je veux pouvoir me créer un compte pour devenir un Explorateur
M2	En tant qu' Explorateur je veux me connecter, pour utiliser les fonctionnalités de l'application.
M3	En tant qu' Explorateur connecté , je veux me déconnecter
M4	En tant qu' Explorateur connecté , je veux scanner une clé d'accès à un Portal pour voyager entre deux Locations.
M5	En tant qu' Explorateur connecté , je veux voir mon inventaire de Runes.
M6	En tant qu' Explorateur connecté , je veux afficher la liste des explorations qu'il a réalisées.
M7	En tant qu' Explorateur connecté , je veux afficher ma collection de <i>Unit</i> .
M8	En tant qu' Explorateur connecté , je veux voir mon total d'Inox et connaître ma location

SERVEUR

Vous devez réaliser un serveur de services web à l'aide du langage **NodeJS** et des outils de votre choix. L'architecture de vos services vous appartient. Vous devez offrir les services nécessaires aux récits utilisateurs des applications clientes.

BASE DE DONNÉES

Votre serveur devra utiliser une base de données pour conserver les données importantes du jeu. Une analyse préliminaire sera faite en classe pour partir du bon pied.

FONCTIONNALITÉS

En plus de répondre aux demandes des deux applications clientes, le serveur doit réaliser les tâches suivantes :

No.	Récit utilisateur
S1	À chaque multiple de 5 minutes de chaque heure, ajouter 2 Inox à chaque Explorateur .
S2	À chaque heure, ajouter aléatoirement entre 2 et 5 runes de chaque type à chaque Explorateur

SCÉNARIOS

Voici des détails à propos de récits utilisateur qui vous seront grandement utile.

M4	En tant qu'Explorateur connecté, je veux scanner une clé d'accès à un Portal pour voyager entre deux Locations.														
Condition	L'explorateur doit être connecté pour avoir accès à cette fonctionnalité														
Détails	<ol style="list-style-type: none"> L'explorateur numérise le code QR d'un Portal, à l'aide d'un appareil mobile. L'application mobile fait la demande au serveur global d'Andromia des informations de ce Portal. <ol style="list-style-type: none"> Ce Portal n'existe pas: Affichage d'un message à l'utilisateur Ce Portal existe: Réception des informations à propos de cette exploration (Suite du scénario). Affichage d'un écran avec les informations de l'exploration <ol style="list-style-type: none"> Si le Portal contenait une <i>Unit</i>, l'explorateur peut la débloquent en activant le noyau ce qui termine le voyage Dans tous les cas, il peut terminer le voyage Suite à la fin du voyage, voici les situations possibles <table> <tr> <th>Scénario A</th><th>Scénario B</th><th>Scénario C</th><th>Scénario D</th></tr> <tr> <td>Pas de Rune Pas d'Unit</td><td>Contient des Runes Pas d'Unit</td><td>Pas de Rune Contient une Unit</td><td>Contient des Runes Contient une Unit</td></tr> <tr> <td>A.1. Ajout de l'exploration aux explorations de l'explorateur</td><td>B.1. Faire A.1 B.2. Ajouter les runes à l'inventaire de l'explorateur</td><td>C.1. Faire A.1 C.2. Si l'explorateur choisi de débloquent la <i>Unit</i>, on ajoute cette <i>Unit</i> à sa Collection</td><td>D.1. Faire A.1 D.2. Faire B.2 D.3. Faire C.2</td></tr> </table>			Scénario A	Scénario B	Scénario C	Scénario D	Pas de Rune Pas d'Unit	Contient des Runes Pas d'Unit	Pas de Rune Contient une Unit	Contient des Runes Contient une Unit	A.1. Ajout de l'exploration aux explorations de l'explorateur	B.1. Faire A.1 B.2. Ajouter les runes à l'inventaire de l'explorateur	C.1. Faire A.1 C.2. Si l'explorateur choisi de débloquent la <i>Unit</i> , on ajoute cette <i>Unit</i> à sa Collection	D.1. Faire A.1 D.2. Faire B.2 D.3. Faire C.2
Scénario A	Scénario B	Scénario C	Scénario D												
Pas de Rune Pas d'Unit	Contient des Runes Pas d'Unit	Pas de Rune Contient une Unit	Contient des Runes Contient une Unit												
A.1. Ajout de l'exploration aux explorations de l'explorateur	B.1. Faire A.1 B.2. Ajouter les runes à l'inventaire de l'explorateur	C.1. Faire A.1 C.2. Si l'explorateur choisi de débloquent la <i>Unit</i> , on ajoute cette <i>Unit</i> à sa Collection	D.1. Faire A.1 D.2. Faire B.2 D.3. Faire C.2												

INFORMATIONS TECHNIQUES

AUTHENTIFICATION

Pour l'authentification, nous devons utiliser un système de jeton. Ce qui veut dire que lors de la connexion, nous allons générer un jeton à l'Explorateur si l'authentification est réussie. Par la suite, pour chacune des requêtes au service Web l'application cliente devra fournir ce jeton dans les entêtes de la requête HTTP, pour authentifier la demande. Nous avons fait quelque chose de similaire en classe.

NODE.JS

- express
- base64-url
- cors
- crypto
- dotenv
- express-jwt
- halson
- jsonwebtoken
- knex
- passport
- passport-local
- uuid
- node-schedule :
 - <https://github.com/node-schedule/node-schedule/wiki>
- *moment* : Gestion des dates
 - <https://www.npmjs.org/package/moment>
- Les modules utilisés et développés lors d'un travail pratique 1

FURETEUR

- jQuery et la fonction `$.ajax`
 - <http://stackoverflow.com/questions/3258645/pass-request-headers-in-a-jquery-ajax-get-call>
- Local Storage
 - http://www.w3schools.com/html/html5_webstorage.asp
- Le modèle CSS fourni

ANDROID

- Sauvegarder des données
 - <https://developer.android.com/training/basics/data-storage/shared-preferences.html>
- Fuel
 - <https://github.com/kittinunf/Fuel>
- ZXing
 - <https://github.com/journeyapps/zxing-android-embedded>

PONDÉRATION

Pour ce travail l'évaluation sera faite un peu comme une évaluation de projet, chaque équipe devra faire avancer le projet du mieux de ses connaissances et du nombre de membres dans l'équipe. Les méthodes agiles permettront à l'enseignant de faire le suivi du projet et d'évaluer chacune des membres des équipes de manière individuellement.

Ce travail compte pour 50% de la note final du cours, les points seront distribués de cette manière

PARTIE 1 – DOSSIER D'ARCHITECTURE

La partie 1 de l'épreuve synthèse consiste en la rédaction du dossier d'architecture. Cette partie correspond à 15% de la note du cours, les points seront distribués de cette façon.

Date de remise : **avant 23 h 59 le vendredi 1 décembre 2017**

Élément de correction	Points
Justification adéquate des choix architecturaux	1,5
Définition appropriée des clients	3
Définition appropriée du serveur	3
Définition juste des interactions entre les clients et le serveur	7,5

PARTIE 2 – APPLICATION DISTRIBUÉE

La partie 2 consiste au développement des trois applications (2 clients et 1 serveur). Cette partie correspond à 35% de la note du cours, les points seront distribués de cette façon.

Date de remise : **avant 23 h 59 le mercredi 20 décembre 2017**

Élément de correction	%
Qualité du code	20 %
Fonctionnalités des clients conformes aux exigences	35 %
Fonctionnalités des services conformes aux exigences	25 %
Respect des bonnes pratiques	20 %
• Performance	
• Maintenance	
• Évolution des applications	