



IUT	Robert Schuman	
Institut universitaire de technologie		
	Université de Strasbourg	

EREDEJEUX

RAPPORT DE STAGE

Création d'une API et d'une interface graphique pour le support
d'un jeu de société sur ordinateur

KELHETTER Hugo Etudiant en deuxième année
informatique
06/04/2021 - 18/06/2021

EREDEJEUX : 33 RUE DES ROITELETS 44150 ANCENIS

Maitre de stage : Mme. Isabelle CHARPENTIER – Directrice de recherche au CNRS

Tuteur pédagogique : M. Loic MAZO

Remerciements	3
Introduction.....	4
1 Le contexte de la mission.....	5
1.1 Présentation de l'entreprise	5
1.2 La mission	6
1.3 Description d'une partie	6
2 Mise en œuvre de la mission	8
2.1 Les causes du problème	8
2.2 La méthode choisie et les raisons de ce choix.....	8
2.3 La mise en œuvre	11
2.4 La création de la carte	11
2.5 Les Interfaces.....	14
2.6 Les phases de test.....	14
2.7 Les commentaires.....	14
3 La communication et notre insertion dans l'entreprise	16
3.1 Les visio-conférences.....	16
3.2 La communication écrite	16
3.3 La communication avec Daniil.....	17
4 Les résultats	19
4.1 Les réussites et échecs	19
4.2 La satisfaction cliente	19
4.3 Ce que j'ai appris	19
Conclusion	21
Glossaire	22
Bibliographie	23
Annexes	24

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de ce stage et aux personnes qui m'ont assisté dans la rédaction de ce rapport de stage.

Tout d'abord, je tiens à remercier Mme Isabelle CHARPENTIER, mon maître de stage qui s'est assurée que mon stage se passe dans les meilleures conditions. Lors de ce stage, son aide dans la création de mon projet ainsi que le temps passé avec moi pour m'expliquer les notions d'entreprise ou encore de droit de l'informatique m'ont beaucoup apporté.

Je remercie aussi M. Olivier BARRETEAU, chercheur en science de l'eau à l'INRAE Montpellier et membre de l'équipe d'étude de développement du projet qui a assisté à nombres de nos discussions et qui a permis d'apporter un regard critique extérieur ainsi qu'une aide dans la compréhension du sujet qui était en perpétuelle évolution.

Merci à M. Michael ESSA, administrateur réseau au laboratoire ICUBE à Illkirch qui a toujours été présent pour répondre à mes questions concernant le déploiement du projet sur internet mais aussi les normes de sécurité informatique.

Enfin, je tiens à remercier ma mère qui m'a soutenu tout au long de ce stage et m'a assisté au cours de la rédaction de ce rapport.

Introduction

L'entreprise que j'ai intégrée aux côtés d'un autre stagiaire de mon établissement, Daniil KUDRISHOV, travaillait sur la conception d'un jeu sérieux traitant de la gouvernance de l'eau dans le milieu agricole. Dans ce jeu de société, les joueurs peuvent incarner un agriculteur, un élu ou un gestionnaire de bassin-versant*. Un bassin-versant est un espace drainé par un cours d'eau et ses affluents sur un ensemble de versants. Toutes les eaux dans cet espace s'écoulent et convergent vers un même point de sortie appelé exutoire*. Les agriculteurs se voient partagé le bassin versant en différents sous-bassins pour y effectuer leurs productions agricoles sur différentes parcelles représentées par des cases. Les agriculteurs ont alors le choix de changer leur méthode de production sur une parcelle ou sur l'entièreté des terrains. L'agriculteur peut décider de passer d'une production intensive de viande de bœuf à une production biologique et inversement. Ces changements ont un coût financier, temporel et nécessitent un niveau de formation du joueur. Les élus et gestionnaires ont un rôle de régulation notamment pour veiller aux émissions de polluants dans la terre.

Si en apparence ce jeu peut sembler simple, il devient coûteux en calcul lorsque l'on sait que chaque agriculteur peut effectuer une politique de production différente sur chacune de ses parcelles. Les éventuels pesticides déposés par l'agriculteur sont alors drainés par les cours d'eau vers les parcelles en aval, rendant les calculs rapidement trop compliqués. Il faut aussi prendre en compte la topographie du terrain. En effet, le terrain n'étant pas plat, les eaux ne pouvaient s'évacuer vers n'importe quelle case voisine. Enfin, la perméabilité du sol n'est pas uniforme sur tout le bassin. L'eau qui peut relativement aisément infiltrer le sol en zone agricole ou forestière, aura du mal à traverser la couche de béton dans les villes. En plus de tout le matériel requis pour jouer, il est donc nécessaire d'automatiser les calculs.

Le second problème est la durée nécessaire pour entrer en ligne de commande toutes les données des joueurs par une personne, faisant émerger la nécessité d'une interface pour chaque joueur avec une unité centrale de calcul.

1 Le contexte de la mission

1.1 Présentation de l'entreprise

Le CNRS a fait une demande de création de jeu sérieux dans le but de pouvoir éduquer ses joueurs sur les questions environnementales tout en récupérant les données pour analyser les actions des joueurs. Les pratiques agricoles sont de plus en plus déconnectées du contexte socio-environnemental dans lequel nous sommes. L'agriculture est considérée comme l'un des acteurs principaux impactant la qualité de l'eau des rivières comme dans les régions côtières augmentant ainsi les tensions entre le secteur agricole et certaines organisations, notamment celle pour la protection de l'environnement. Le but est donc de mettre en place des pratiques agricoles plus respectueuses de l'environnement tout en gardant une production suffisante. Il faut aussi noter qu'une production plus respectueuse produit en effet moins, mais est aussi bien plus couteuse.

Pour se faire, le CNRS a fait appel aux zones ateliers. Les zones ateliers (ZA) sont des fédérations de recherches interdisciplinaires, inter-instituts et inter-organismes, portés par le CNRS, depuis 12 ans. Elles constituent un service national d'observation et de recherche en environnement, ancré dans un tissu international. Les zones ateliers mènent des recherches fondamentales et appliquées dans une optique long-termiste dans le domaine environnemental pour apporter des réponses aux demandes de la société. On peut notamment citer les zones ateliers situées en Lorraine et dans le bassin versant de la Seine qui s'intéressent à la qualité de l'eau dans leur milieu respectif sous l'effet des activités humaines.

Eredejeux, une entreprise prestataire des zones ateliers, est située à Ancennis en Loire-Atlantique, spécialisée dans la création et la fabrication de jeux culturels à tirage limité. C'est en tant qu'employés de l'Unistra que nous avons intégré l'entreprise au titre de prestataire. L'entreprise s'adresse à une large variété de clients. Elle propose ses services aux entreprises, institutions ou associations pour la réalisation de jeux à thème divers et variés allant du patrimoine au domaine médical ou environnemental. Pouvant pallier les difficultés de ses clients, l'entreprise travaille de pair avec eux pour fournir la mise en forme si les mécaniques de jeux sont déjà présentes ou encore prendre le rôle de l'équipe graphique si le jeu a déjà été conceptualisé. Leurs différentes créations ont un but commun, permettre l'apprentissage par voie ludique et favoriser la curiosité pour inciter la lecture sur des sites thématiques qui sont en chute libre depuis plusieurs années. L'entreprise a entre autres créé un jeu des 7 familles sur le patrimoine ferroviaire pour la société des chemins de fer de la Vendée ou encore une roue

interactive présentant le cycle de vie du raisin pour appuyer le discours des vignerons lors des visites des caves.

1.2 La mission

La présence abondante de calculs, qui éclipse tout sens ludique au jeu pour ne laisser place qu'à des traitements longs et rébarbatifs, a rendu indispensable une interface graphique pour entrer les calculs. Pour rendre le tout plus agréable, il a été demandé d'ajouter différents composants du jeu sur l'interface tels que la carte pour mieux sélectionner quelles actions jouer sur quelle case ou encore obtenir des informations concernant ses ressources disponibles ou les résultats de sa production. S'ajoute à cela un animateur faisant partie de l'entreprise qui joue un rôle de maître du jeu. L'animateur possède certains droits uniques. Par exemple, si un agriculteur veut échanger une case avec un autre, il doit passer par l'animateur qui après expertise de la situation acceptera ou non l'échange. De nouvelles actions ont été ajoutées au fur et à mesure de l'avancement du projet. Il est aussi capable de mettre en place des infrastructures d'irrigation ou écologiques. Les infrastructures écologiques ont entre autres pour but d'éviter la propagation des polluants par les cours d'eau. Les infrastructures peuvent être des installations très simples telles que mettre une bande d'herbe autour de sa parcelle. Il lui est aussi possible de pouvoir transformer la carte comme bon lui semble au début de la partie afin de mettre les agriculteurs dans des conditions uniques. Il doit aussi être capable de faire avancer la partie en terminant un tour bien que les joueurs n'aient pas validé leurs actions.

1.3 Description d'une partie

Au lancement de la partie, seul l'animateur a accès à l'interface de jeu. Il lui est possible d'affecter la carte comme bon lui semble. Transformer toutes les forêts en villes et zones agricoles, effectuer des échanges entre zones agricoles des joueurs, il est capable de faire ce qu'il veut. Une fois que l'animateur est satisfait, les autres joueurs peuvent prendre connaissance de leur sous-bassin respectif. S'enclenche alors des phases de dialogue entre les joueurs pour enfin passer à une phase d'action. Les joueurs peuvent alors changer leurs pratiques agricoles, installer des infrastructures, effectuer des échanges. Lorsque la phase des actions est terminée, le serveur enregistre les différentes modifications effectuées par les joueurs, réalise les calculs nécessaires et renvoie à chaque joueur un graphique présentant l'évolution de ses ressources. Pour donner suite à une nouvelle phase de dialogue, les joueurs

peuvent remplir un petit questionnaire où ils indiqueront leur ressenti et leur satisfaction à la vue des résultats sur une échelle allant de -2 à 2. Ceci fait, un nouveau tour peut commencer et les joueurs peuvent apporter leur nouveau lot de modifications.

2 Mise en œuvre de la mission

2.1 Les causes du problème

Nous nous retrouvons donc avec un projet hybride physique et virtuel. Dans la partie physique, les joueurs disposent d'un plateau matériel constitué d'hexagones, de pions pour représenter les différentes propriétés des cases ainsi qu'un jeu de cartes d'action pour les joueurs. Notre mission se dessine alors autour de trois composantes majeures :

- la création d'un serveur auquel il est possible d'entrer les données à calculer et recevoir les données permettant d'avancer dans la partie. Il est nécessaire par celui-ci de pouvoir envoyer et recevoir des informations depuis le client mais aussi de pouvoir lancer un simulateur conçu par notre maître de stage afin de réaliser tous les calculs nécessaires.

- la conception d'une interface personnelle à chaque joueur pour visualiser une carte représentant son bassin versant ainsi qu'un tableau de bord pour y entrer toutes les informations nécessaires. Un système de communication instantanée intégrée à l'application est aussi suggéré pour permettre les échanges lorsque les joueurs sont en distanciel. Il est nécessaire de conserver un historique de toutes les actions effectuées par les joueurs sur le serveur de jeu mais aussi sur un service tiers tel que gitlab ou google drive. Ces fichiers doivent être stockés au format .txt ou .csv pour permettre une lecture facile et effectuer des analyses à posteriori en comparant les actions sur différentes parties.

- et à l'aide d'une machine virtuelle qui nous a été fournie mais aussi de rendre l'application utilisable sans connexion internet en transformant un des ordinateurs des joueurs en serveur local.

2.2 La méthode choisie et les raisons de ce choix

Lors de la première semaine de stage, nous avons assisté à plusieurs réunions dans lesquelles, nous avons pris connaissance de façon plus précise de la forme que devait prendre le jeu et des ressources qui étaient à notre disposition pour mener à bien le projet. Nous avons aussi différentes présentations utilisées par l'équipe d'étude pour conceptualiser le jeu. Ces documents contenaient toutes les informations requises pour comprendre toutes les dynamiques du jeu et l'articulation d'une partie. Nous y retrouvions un glossaire expliquant les mots clés, la présentation des différents rôles, les interactions possibles des joueurs, et le déroulement d'un tour typique d'une partie. Notre première tâche était alors de prendre pleine connaissance des

documents et de créer un schéma représentant les différents échanges au cours d'une partie entre le client et le serveur.

Nous avons aussi pu assister à différentes présentations des outils utilisés pour la génération des cartes avec l'outil Qgis. Il s'agit d'un logiciel permettant entre autres de créer des cartes à l'aide de données satellites où à l'aide d'un jeu de données personnel, notre tâche étant de déterminer dans quelle mesure utiliser ces ressources. Il était possible de générer des cartes sous Qgis que nous pourrions exporter vers notre application ou utiliser un autre outil à l'aide des données brutes de la carte.

Au cours des deux premières semaines, Daniil et moi avons fait nos recherches sur la meilleure manière de mener à bien cette mission. Nous en sommes tous les deux venus à la conclusion qu'il serait le plus simple de créer un site internet en raison de la grande flexibilité dont on peut jouir par l'utilisation de la programmation web.

Nous avons décidé d'utiliser le *framework** React. Un *framework* étant un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. React* est un framework javascript développé par Facebook depuis 2013 et qui a pour but de faciliter la conception d'interface web. Ce *framework* a pour avantage de faciliter la création de composants graphiques sur une page et de les maintenir à jour grâce à des variables d'état qui leur permettent de savoir quel élément doit être rechargé à quel moment. Ce framework a aussi pour avantage de mettre facilement et rapidement à jour des éléments. Je me suis penché sur l'utilisation de React après avoir découvert ce *framework* au cours du semestre 4 dans lequel j'ai pu avoir un avant-goût de la puissance qu'il proposait. Cet intérêt a ensuite été décuplé lorsque Daniil a dit avoir trouvé une bibliothèque permettant la création d'une carte interactive en générant une suite d'hexagones capable de recréer à l'identique le plateau de jeu physique. Une bibliothèque est un ensemble d'outil permettant de traiter des problèmes dans une thématique donnée.

Lors de mes premiers essais, j'ai tenté sans utiliser React de créer la carte à l'aide d'un fichier externe qu'il serait possible de récupérer depuis le serveur. Cette option n'a pas été choisie car le passage à React rendait le tout bien plus compliqué. Il était aisé de récupérer le dit fichier, le problème venait surtout lors de l'affichage. Il n'est pas recommandé par React d'insérer directement des éléments html depuis une source extérieure, et ma modification de la carte posait d'autant plus de problèmes, me faisant décider d'abandonner cette piste pour trouver de nouvelles options.

La partie serveur posait moins de questions. Il avait pour seule fonction de communiquer

avec les bases de données, interagir avec chaque joueur, lancer les calculs une fois les données reçues et renvoyer les résultats. Le *framework* nodejs a été choisi d'une part pour garder de la cohérence en écrivant toute l'application en un seul langage mais aussi car il était enseigné en même temps que React. La seule contrainte étant que les clients doivent pouvoir communiquer avec le serveur, sa mise en place peut être faite de deux manières différentes. La première solution était la plus évidente, il suffisait de publier le site sur internet. L'autre méthode était d'utiliser l'ordinateur de l'animateur pour créer un réseau local privé d'où il peut lancer le serveur. Cette alternative est apparue dans les discussions pour pallier les problèmes de connexion dans les zones blanches.

Un dernier problème concerne la communication entre le serveur et le client. Il est possible d'utiliser des requêtes *http* qui permettent différentes actions simples telles que récupérer des données ou encore envoyer des données et les modifier. Le problème majeur avec cette solution est que le serveur ne peut communiquer avec le client que s'il en fait la demande explicite à l'aide d'une requête au préalable. Pour la soumission des données ce n'est pas un problème bloquant. Cela en devient un pour la récupération des données. Les calculs sont effectués à la fin de chaque tour, ce qui implique qu'un joueur ne peut pas savoir concrètement quand le serveur est prêt à envoyer les données traitées. Il est envisageable de créer une boucle dans laquelle le client demanderait en permanence les résultats tant que la réponse ne serait pas satisfaisante, mais cela présente des problèmes d'utilisation des ressources. Le serveur se retrouverait vite surchargé pour répondre aux demandes incessantes des utilisateurs. D'où le besoin de permettre au serveur de communiquer avec le client de son propre chef.

La bibliothèque Socket.IO permet cela. Cette bibliothèque permet de créer une liste de comportement lors de la réception de requête du client mais il peut aussi s'abonner à des écouteurs pour recevoir les informations lorsque le serveur effectue ses propres requêtes. Le serveur a alors la possibilité de recevoir toutes les informations d'une partie, faire les calculs et renvoyer les résultats sans que les joueurs n'aient à faire des demandes incessantes. Un autre avantage est que les clients n'ont plus besoin d'envoyer les fichiers sur un service tiers car le serveur enregistre directement les fichiers sur l'ordinateur de l'animateur.

Socket.IO fonctionne avec un système d'abonnement. Supposons que je veuille me connecter à une partie, je vais envoyer une demande au serveur en y ajoutant des données telles que mon nom. Le serveur, en s'étant abonné à l'événement "demande de connexion" saura instantanément quoi faire des données reçues.

Au commencement du projet, mes connaissances en React restaient minces : j'avais des

idées de fonctionnalités mais je ne savais pas comment les mettre en place. Au cours de mes recherches, je me suis accordé des périodes lors desquelles j'ai pu regarder des tutoriels vidéo. Les vidéos montrant des développeurs aguerris créer des applications de zéro sont légion et il a été facile de trouver de l'inspiration. Bien que lors de mon premier visionnage, ces vidéos aient été trop avancées ou alors ne répondaient pas à une de mes problématiques, elles m'ont permis de garder certaines notions clés ou de m'ouvrir sur des nouvelles façons de programmer.

2.3 La mise en œuvre

Au cours des deux premières semaines de stage, nous étions chargés de travailler séparément afin de pouvoir nous construire un avis sur la manière d'aborder le sujet. A la fin de ces deux semaines, nos productions étaient différentes concernant la répartition du travail. Daniil avait plus travaillé sur la partie serveur alors que je m'étais concentré davantage sur la partie client. La répartition du travail pour les semaines à venir s'est alors faite naturellement. J'étais chargé de la construction de l'interface et Daniil du serveur.

2.4 La création de la carte

La construction des cartes est réalisée à l'aide de la bibliothèque *react-hexgrid* permettant de réaliser des grilles hexagonales en React. Cette bibliothèque a de nombreux avantages dans ce projet. Il est possible de créer un hexagone en entrant en paramètre la position. Le programme se charge alors de gérer la position des hexagones entre eux en évitant des superpositions et ce en autorisant de changer la taille ou en mettant un écart entre chaque hexagone. Une fois les hexagones tracés pour représenter le bassin, il est possible d'ajouter des chemins entre eux, ce qui permet dans ce projet de dessiner les cours d'eau. Les éléments construits par la bibliothèque sont tous des objets svg. Alors qu'il aurait été plus facile de générer une image que le serveur aurait pu afficher directement, le format svg est à privilégier car il présente beaucoup plus d'avantages : une image classique est composée de pixels de couleur. Pour modifier une image, il est nécessaire de savoir précisément quels pixels changer, de plus, afin de rendre une image cliquable, il faut utiliser la balise html `<area>`. Cependant, pour se faire, il faut fournir les coordonnées de chaque point de chaque zone. Cela revient à dessiner directement soi-même la carte. Le format svg permet de passer outre ces problèmes. Un fichier svg n'est pas une suite de pixels affichant une couleur mais des formes géométriques comprises par le langage html. Un hexagone par exemple est un polygone pour lequel on a donné les coordonnées des 6 arêtes.

Cela permet d'une part de régler les problèmes de couleur en ajoutant des éléments css classiques, mais aussi de gérer l'interactivité de la carte. Ces polygones, comme tout autre élément d'une page web, tel qu'un bouton, peuvent être cliqués indépendamment de leur environnement. Il est donc possible d'identifier quelle case a été cliquée et d'ajuster le contenu de la page en conséquence.

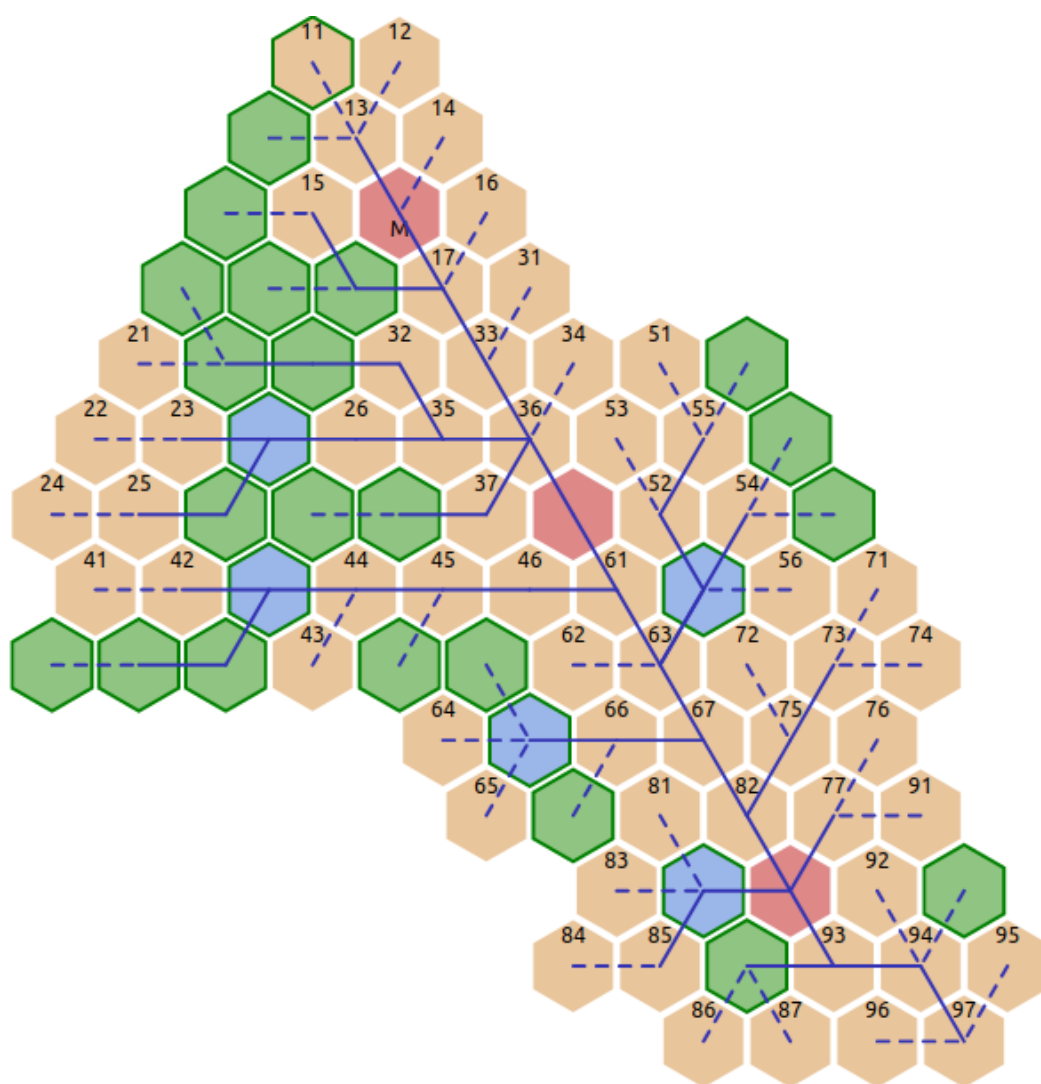


Figure 1 Carte type d'un animateur

Nous retrouvons ici la carte complète (figure 1) visible par l'animateur. Les cases vertes sont des forêts et les cases bleues des étangs. Ces deux types de cases ont un contour vert indiquant qu'ils ont un effet filtrant pour la pollution. Les cases rouges représentent les villes. Nous pouvons voir un 'M' inscrit sur la ville du haut indiquant la présence d'un marché local. Les autres cases sont des parcelles agricoles, appartenant aux différents agriculteurs. Les cases

portant un numéro compris entre 10 et 19 appartiennent à l'agriculteur numéro 1, les cases portant un numéro compris entre 20 et 29 appartiennent à l'agriculteur numéro 2, et ainsi de suite jusqu'à l'agriculteur numéro 9. La case numéro 11 possède elle aussi un contour vert indiquant que l'agriculteur a mis en place une infrastructure écologique sur sa parcelle. Les traits bleus représentent eux les cours d'eau. Ils peuvent être continus ou pointillés, le cas échéant indiquant que la quantité d'eau n'est pas suffisante pour former une rivière à proprement parler.

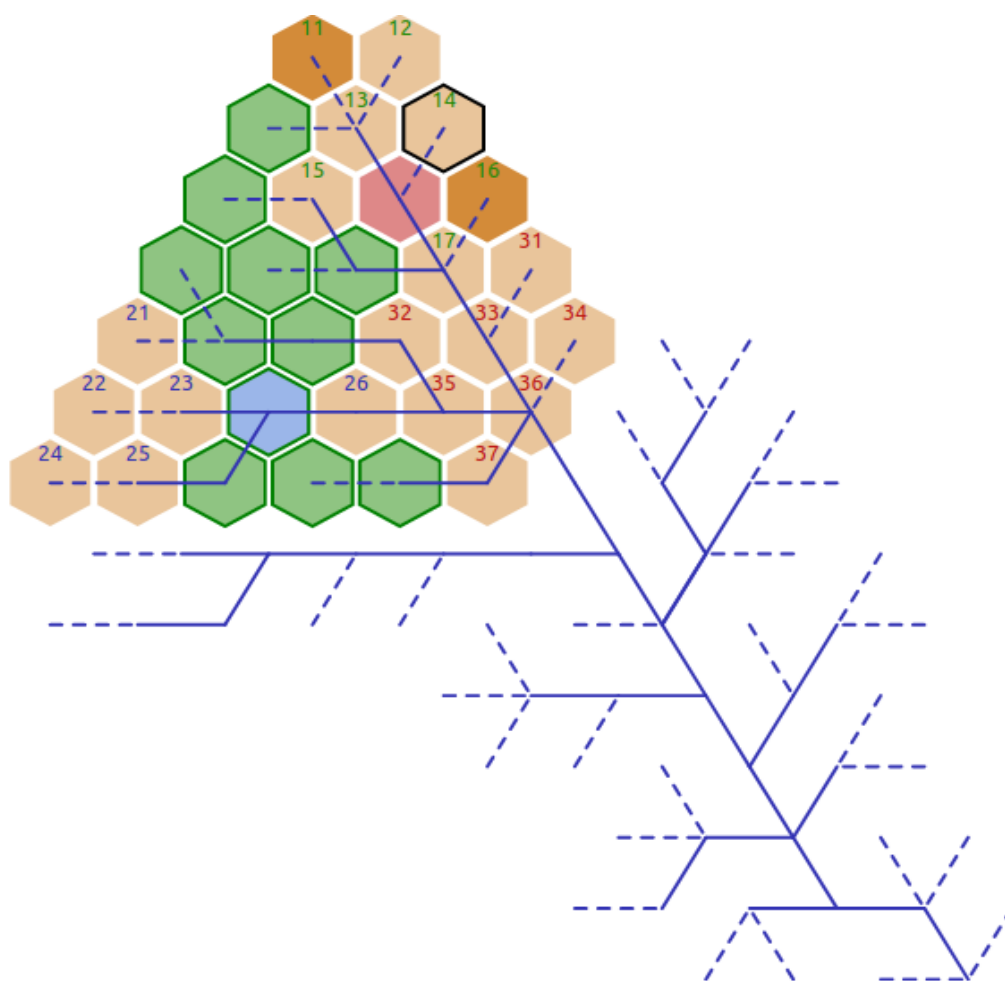


Figure II Carte type d'un agriculteur

Cette carte est celle visible par l'agriculteur numéro 1 (figure II). Nous voyons qu'il ne peut voir ce qu'il se passe au-delà de son sous-bassin qu'il partage avec les agriculteurs numéro 2 et 3. Les cases 11 et 16 ont une couleur plus foncée montrant que le joueur a effectué des modifications durant ce tour sur celles-ci. La case 14 a un contour noir permettant au joueur de mieux visualiser quelle case il a sélectionné et donc sur quelle case ses futures actions vont se

répercuter.

2.5 Les Interfaces

Il a été nécessaire lors de la conception des différentes interfaces de faire des choix pour fournir une interface la plus ergonomique et facile à prendre en main malgré des compétences en graphisme réduites. Si cela s'est avéré compliqué pour les interfaces de jeu où il fallait afficher la carte ainsi que les différents formulaires, il était bien plus simple de le faire dans les écrans d'accueils qui demandaient très peu d'actions (Figures III à VIII).

2.6 Les phases de test

Au cours du projet, Daniil et moi-même avons effectué un grand nombre de tests pour s'assurer que tout marchait comme il le fallait. Ce qui nous manquait cependant était des phases de test avec des personnes qui ne connaissaient pas le programme. Nous avions pour objectif de faire en sorte que le programme soit jouable lors d'un atelier de test à l'université de Tours les 14 et 15 juin. Cet atelier avait pour but de déceler des bugs restant mais aussi de faire évaluer le jeu auprès de joueurs.

Avant cela, nous avons eu une autre réunion de test le 11 juin avec Mme Charpentier qui était accompagnée d'une collègue qui n'avait aucune connaissance en informatique.

Cette première séance de test a permis de réaliser l'étendu des tests que je n'avais pas réalisés. Les comportements que j'ai observés, pour certains, ne m'étaient jamais venus à l'esprit. Par exemple, lors du début de la partie, un écran d'attente s'affiche le temps que l'animateur fasse ses modifications avant de débiter la partie. D'autre part, un joueur peut se reconnecter à la partie si par exemple il avait fermé sa fenêtre puis retrouvé son interface de jeu. Mais que se passerait-il si un joueur venait à se reconnecter avant que l'animateur ne lance sa partie ? Au moment des tests, je n'avais pas imaginé une seule seconde qu'un joueur puisse se déconnecter à un tel moment. Ce fut une réelle surprise quand ce joueur s'est reconnecté pour arriver sur l'écran de jeu des agriculteurs alors que les autres étaient encore en attente, l'animateur n'ayant pas encore commencé la partie.

2.7 Les commentaires

Ce projet ayant pour but d'évoluer, il nous a été demandé de commenter l'intégralité de

notre code en respectant une convention pour permettre aux prochaines personnes de mieux s'y retrouver. Dans chaque corps de commentaire il nous a aussi été demandé d'inclure le nom de chaque personne ayant travaillé sur la partie de code en question. Cela a été demandé car Mme Charpentier avait pour projet de déposer ses programmes et par la même occasion, faire protéger nos parties de code respectifs. (Figure VII)

3 La communication et notre insertion dans l'entreprise

3.1 Les visio-conférences

Tout au long du stage, nous avons installé une routine d'au minimum deux réunions en visioconférence par semaine. Ces réunions en présence de mon maître de stage Mme Isabelle CHARPENTIER et de M. Olivier BARRETEAU nous permettent d'une part de poser des questions, mais aussi de faire une démonstration du travail effectué, de discuter de ce qui convient ou pas dans le projet et des prochaines tâches à effectuer. Il nous était notamment demandé de créer des schémas pour appuyer nos propos et montrer que nous avions compris ce qu'il nous était demandé (Figure VIII). Au cours de ces discussions il était aussi possible de proposer des modifications du projet en fonction de nos capacités et de notre confiance en la réalisation de certaines tâches. Il arrivait que nos attentes soient revues à la baisse comme à la hausse au fur et à mesure que nous prenions en main le projet.

En plus de ces réunions hebdomadaires il nous a été donné l'occasion d'assister à des réunions avec l'équipe d'étude de la création du jeu physique. Ces réunions où il n'était pas question du développement du jeu informatique, m'ont permis de découvrir une nouvelle manière de communiquer bien différente de ce que j'ai pu expérimenter au cours des réunions de mes projets tutorés à distance.

3.2 La communication écrite

Afin d'avoir accès à un serveur permettant d'héberger l'application, il a fallu définir nos besoins et par quel moyen les satisfaire. Une fois cela fait, nous avons été mis en communication avec l'administrateur réseau afin de discuter de ce qui pourrait nous être fourni. Il nous a aussi été possible de nous renseigner dans des domaines qui nous étaient étrangers tels que la sécurité réseau. Bien que l'utilisation d'un certificat http officiel soit nécessaire pour permettre une sécurité optimale sur le site, l'utilisation d'un certificat auto-signé, c'est-à-dire généré par nos soins et donc gratuit, est suffisant pour les communications avec notre serveur de test. Il est aussi nécessaire dans le cas de l'utilisation d'un serveur local de se déconnecter de tout réseau

internet et de ne permettre l'accès qu'à des personnes de confiance.

3.3 La communication avec Daniil

La communication avec Daniil était devenue primordiale lors de la seconde partie du stage afin de pouvoir comprendre quels étaient chacun nos besoins, quelles données nous avions besoin de recevoir et ce que nous pouvions envoyer. Une question sous-jacente à "ce que je peux envoyer" est "ce que je veux envoyer". Il existe un minimum de données requises pour faire son travail de chaque côté mais aussi un certain nombre de données qui permettent de faciliter le travail de l'autre côté. Il est possible de mettre en place des marchés dans les villes et de les enlever. Cette procédure est simple d'utilisation, il suffit de sélectionner sa case et d'appuyer sur un bouton pour changer l'état du marché dans la ville. D'un côté informatique, cela pose une question sur le fonctionnement de l'application. Le client se contente-t-il de dire au serveur de changer l'état de la ville en indiquant uniquement de quelle case il s'agit ou faut-il aussi indiquer l'action à faire ? Dans ce cas précis, la question est de savoir qui doit s'occuper de retrouver l'information concernant si oui ou non la ville a un marché et ajuster les modifications en conséquence. Le client récupère ladite information pour ajuster l'affichage. La logique voudrait qu'ayant déjà l'information, je l'envoie au serveur pour lui éviter des calculs inutiles. Cependant, ma participation pour la création des fonctionnalités étant minimales, je me sentais détaché et donc non concerné par cette affaire. Une discussion bien que courte sur ce sujet a permis une optimisation de notre programme commun. Les discussions suivantes ont permis d'arriver à un consensus sur des problèmes bien plus complexes.

Sur la période du stage, Daniil et moi étions les seuls à programmer sur la version informatique du projet. Cependant, nous savions qu'un autre étudiant allait reprendre le projet par la suite pour ajouter de nouvelles fonctionnalités. Il était alors impératif de rendre le code le plus compréhensible possible que ce soit sur la lisibilité mais aussi pour permettre la prise en main la plus rapide possible à l'aide de commentaires. Nous connaissions certains éléments que le stagiaire sera chargé d'implémenter et cela m'a permis de me demander avec quelle facilité je pourrais moi-même le faire. Cette question a pour but d'évaluer la réutilisabilité de mon code et de créer des composants plus génériques afin de faciliter son travail. Certains composants tels que la carte permettent une réutilisabilité satisfaisante par les paramètres qu'ils acceptent. Ce composant prend en paramètre une fonction qui est déclenchée lorsqu'on clique sur une des cases de la carte. En affectant une nouvelle fonction à l'appel du composant, il est facile de

mettre en place de nouveaux comportements en fonction du rôle du joueur. Un agriculteur peut alors afficher un menu permettant de changer ses méthodes agricoles alors qu'un gestionnaire peut avoir accès à des informations telles que le taux de pollution. Il est aussi important de créer des composants plus petits pour éviter la multiplication du code. Une case à cocher prend de nombreux paramètres, son nom, sa couleur, si elle est cochée par défaut ou non, ce qu'il doit se passer lorsqu'on clique dessus et bien plus encore. Créer un composant qui génère des cases à cocher avec un certain nombre de paramètres communs à toutes permet de ne pas répéter le code et d'éviter des erreurs ou des inconsistances dans le style du composant.

4 Les résultats

4.1 Les réussites et échecs

Les demandes principales de la mission, à savoir créer une interface graphique permettant de communiquer avec un serveur ont été satisfaites. L'interface permet aux agriculteurs de jouer des actions et à l'animateur d'apporter des modifications à la carte. Les interfaces des élus et administrateurs, bien que très similaires avec celle des agriculteurs, n'ont pas été faites pour privilégier d'autres fonctionnalités en raison de leurs rôles secondaires sur la partie informatique.

Le déploiement du serveur sur internet n'a pas été fait ce qui ne laisse comme seule possibilité d'utiliser un serveur local et donc de posséder les fichiers du jeu sur un des ordinateurs. Cette tâche en incombe à l'animateur membre de l'entreprise sans qui l'intérêt du jeu est grandement réduit. Nous avons réussi à installer le jeu sur l'ordinateur de notre maître de stage qui jouera probablement le rôle de l'animateur dans le futur. Cela peut être considéré comme une réussite étant donné que la condition principale était que le jeu soit fonctionnel lors de rendu de l'application. De plus, la publication sur internet n'avait pour vocation de ne durer que le temps du développement jusqu'à l'atelier de test à Tours.

4.2 La satisfaction cliente

Mon maître de stage étant l'un des futurs utilisateurs du programme, il était aisé de communiquer avec les clients afin de connaître leurs impressions, ce qui leur plaisait ou non et satisfaire leurs attentes. Bien que le stage et la phase de développement ne soient pas finis, il semble que Mme Charpentier est satisfaite de l'avancement ainsi que de la tournure que prend le projet.

4.3 Ce que j'ai appris

Sur le plan technique, ce stage m'a permis de mieux me familiariser avec le *framework* React pour lequel j'avais une attirance et une volonté d'approfondir mes connaissances. Durant ce stage, j'ai appris, à l'aide de mon équipe à conceptualiser un cahier des charges qui, contrairement aux projets tutorés, n'était pas seulement des cases blanches à compléter. Il nous a été demandé de nous plier à des restrictions et de fournir avant tout un projet qui plaisait au

client. Étant donné la durée de projet bien supérieure aux projets tutorés qui étaient soit longs, mais avec peu de temps alloué à celui-ci, ou courts et intenses, un travail de fond sur le projet a été permis. Durant ce travail, il était possible de revoir la forme du projet, de retourner en arrière ou de vagabonder à la recherche de bibliothèques permettant de faire son travail avec une meilleure qualité. Si la recherche de bibliothèque n'était pas dans mes habitudes au début du projet et que je me retrouvais régulièrement à vouloir réinventer la roue, Daniil du fait de son utilisation de multiples bibliothèques m'a permis de découvrir leur puissance. Bien que je sois encore loin de maîtriser React, lors de ce stage, je me dis que j'ai au moins un petit peu avancé.

Ce stage m'a aussi beaucoup apporté sur le plan humain. D'un part, les réunions hebdomadaires m'ont aidé par la présentation de ma progression dans le projet à énoncer de façon synthétique mes pensées ainsi que parler à un comité de plusieurs personnes dont certaines que je ne connaissais pas au début du stage. De plus, j'avais pour habitude au cours de mes années à l'IUT de très peu communiquer sur mes projets de groupe, que ce soit pour informer mon équipe de développement de mon avancement ou pour prendre connaissance du sien. Les communications ponctuelles hors des réunions planifiées m'ont permis de prendre l'habitude d'interagir avec les autres membres du groupe et donc de faire avancer le projet de façon plus efficiente.

L'application étant adaptée aux écrans d'ordinateur mais aussi aux tablettes et téléphones, les seules modifications envisageables sont l'ajout de fonctionnalités pour l'animateur. Il a été notamment discuté de l'ajout d'une commande permettant de créer des événements pour manipuler les éléments tels que la météo. Bien que secondaire, le prochain stagiaire devra s'occuper d'ajouter les fonctionnalités aux interfaces des élus et gestionnaires.

Conclusion

En intégrant l'entreprise, j'avais certaines appréhensions. J'aurais premièrement pu ne pas être à la hauteur pour la conception de l'interface, le graphisme et l'esthétisme ne faisant pas partie de mes qualités et mon expérience en css étant encore très superficielle. Ensuite, il était demandé de se familiariser avec les moteurs de jeu créés par le maître de stage et donc de pouvoir les utiliser. Enfin, j'avais comme crainte d'intégrer l'entreprise et de me sentir déraciné de l'environnement scolaire en général mais aussi de ne tout simplement pas savoir quoi faire et de n'avoir aucune idée par quoi commencer, n'ayant pas un sujet détaillant étape par étape quoi faire. C'était sans compter la préparation durant mes années d'études lors de nombreux projets qui m'ont permis de développer une autonomie avec laquelle j'ai toujours su trouver un moyen de répondre à mes problèmes mais aussi grâce à l'accompagnement de mon maître de stage lors des multiples présentations de ses productions.

Étant libre concernant mon travail, que ce soit concernant mon organisation ou les outils utilisés, j'ai pu évoluer sur le plan technique sans ressentir de contrainte venant de l'entreprise et je n'ai donc pas ressenti d'obstacles autres que mes propres lacunes. J'ai pu travailler et me familiariser avec le langage que je désirais, apprendre celui qui m'intéressait et cela m'a permis d'aller peut-être plus loin qu'avec un autre langage que j'affectionnais moins. J'ai aussi appris à communiquer avec les autres employés, à poser des questions, proposer des suggestions et demander de l'aide, choses qui ne faisaient pas partie de mes habitudes. En définitif, ce stage m'a très peu pris pour me donner énormément au niveau technique, communicationnel et humain.

Si ce projet devait être à refaire à l'identique, la démarche technique changerait notamment concernant la recherche de bibliothèque permettant de gagner du temps sur les petites tâches et donc consacrer plus de temps sur des points plus importants tels que l'architecture du programme qui pourrait être améliorée. Une chose est sûre cependant, la communication interne serait probablement plus importante.

Glossaire

Bassin-versant : Espace géographique alimentant un cours d'eau et drainé par lui.

Bibliothèque : ensemble d'outils permettant de traiter des problèmes dans une thématique précise. Une bibliothèque sur les chaînes de caractère permet par exemple de copier, couper ou modifier une chaîne de caractère.

Éxutoire : Point de convergence des cours d'eau d'un bassin-versant.

Framework : un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.

React : framework développé par Facebook pour simplifier la création d'applications web.

Bibliographie

Coding With Chaim. React Chat App Using Socket.IO | Socket IO Tutorial. Youtube, 30/3/2020 [consulté le 16/04/2021] 1 vidéo 20min 30sec

Coding With Chaim. Share Images Using Socket IO | Binary Data W/ Socket IO. Youtube, 17/01/2021 [consulté le 04/06/2021] 1 vidéo 13min 12sec

Facebook Inc. Documentation officielle de React.2013,2021 [consulté de mai 2021 à juin 2021]

Glass Erin, Comment installer le serveur web Apache sur Ubuntu 20.04, digitalOcean, 2020 [consulté le 27/05/2021]. Disponible sur <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-fr>

WATSON Jed, React Select[en ligne]. Disponible sur react-select.com

Traversy Media. Realtime Chat With Users & Rooms - Socket.io, Node & Express. Youtube, [consulté le 15/04/2021]. 1 vidéo 58min 44sec

Web Dev Simplified. How To Create A Messaging App With Socket.io And React. Youtube, 05/09/2020 [consulté le 16/04/2021]. 1 vidéo, 1h 44min 07sec

Annexes

Figure I Carte type d'un animateur	13
Figure II Carte type d'un agriculteur	14
Figure III Écran d'accueil	25
Figure IV Écran de création de partie	25
Figure VI Interface type de jeu d'un agriculteur	26
Figure VII Exemple de code commenté.....	27
Figure VIII Schéma des différentes communications	28



Figure III Écran d'accueil

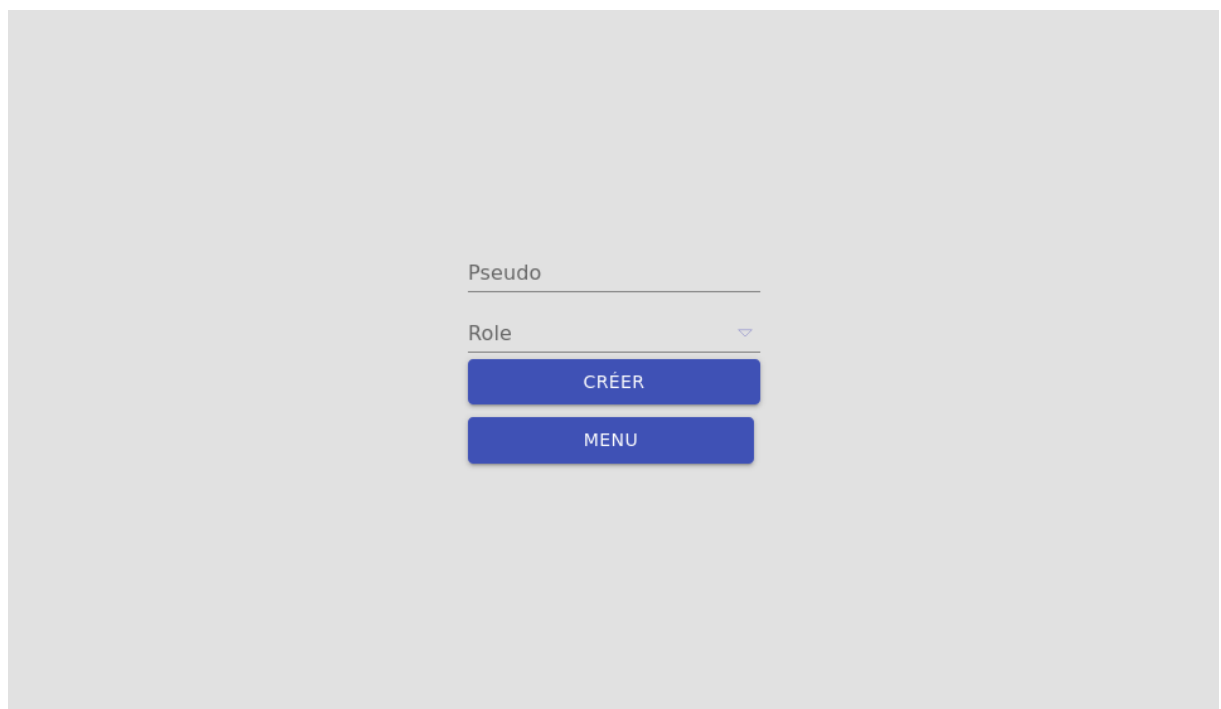


Figure IV Écran de création de partie

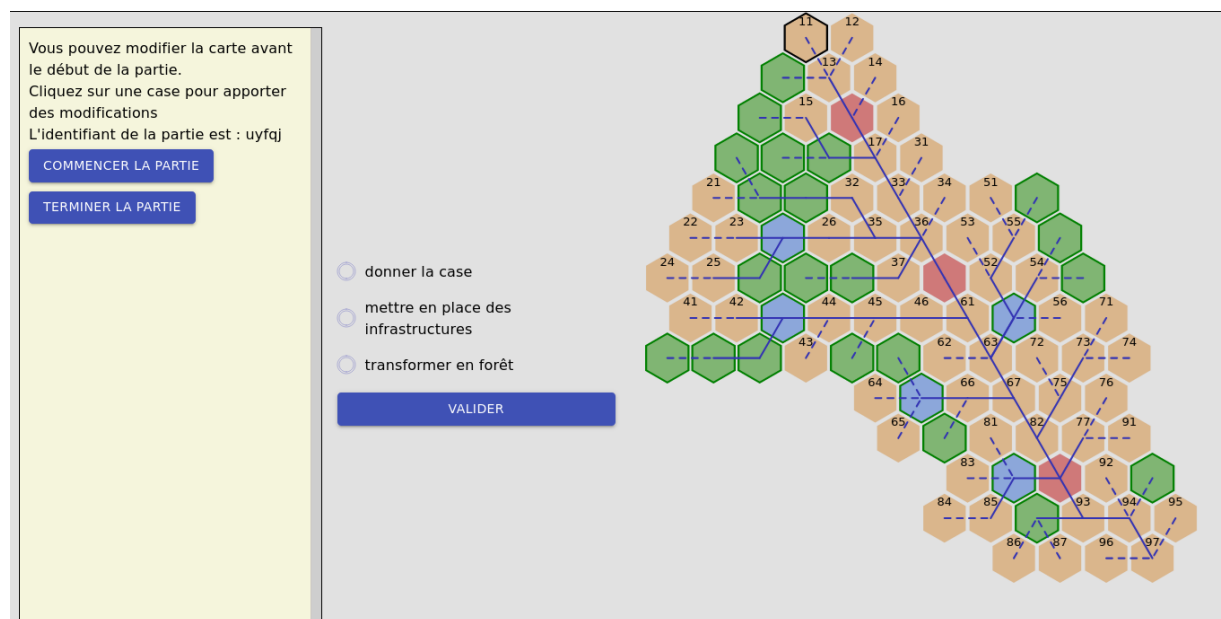


Figure V Interface de jeu type de l'animateur

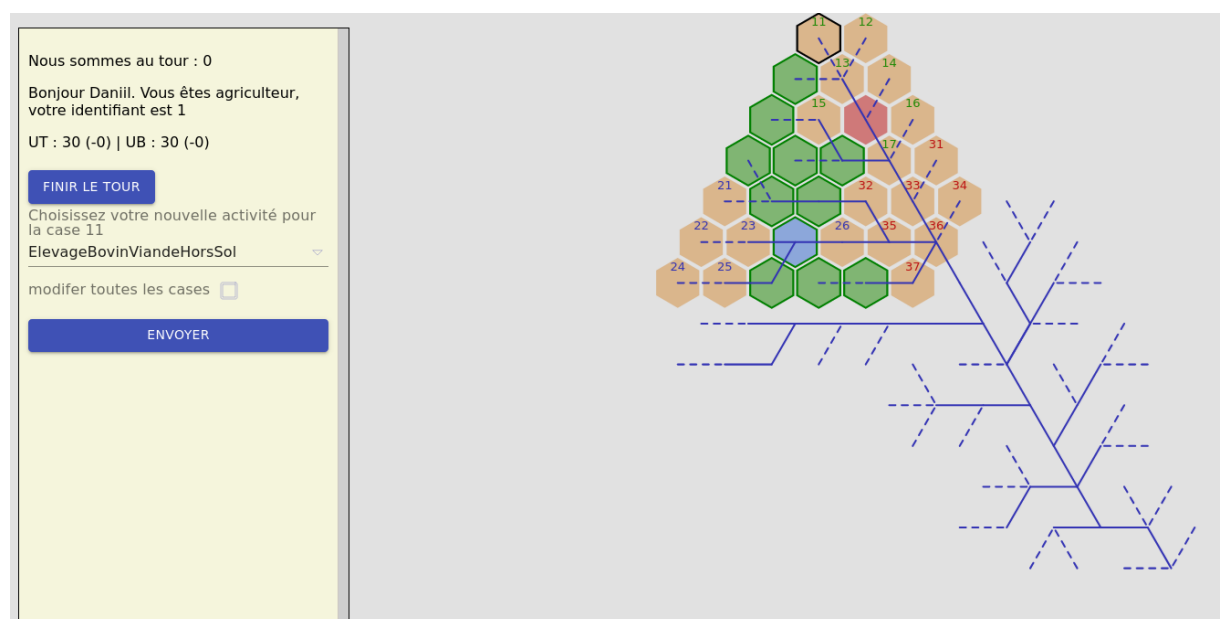


Figure VI Interface type de jeu d'un agriculteur

```

/*
    Function : createCheckbox

    Syntax  : checkbox=createCheckbox(name)

    Input : name : the checkbox's name. it will link the checkbox to
the parent's state

    Output : checkbox : <CheckBox>

    Description : return a checkbox with a set a predefined options

    Author : Hugo KELHETTER

*/
function createCheckbox(name) {
    return <Checkbox
        onChange={this.handleChange}
        name={name}
        color="primary"
        checked={this.state[name]}
    />
}

```

Figure VII Exemple de code commenté

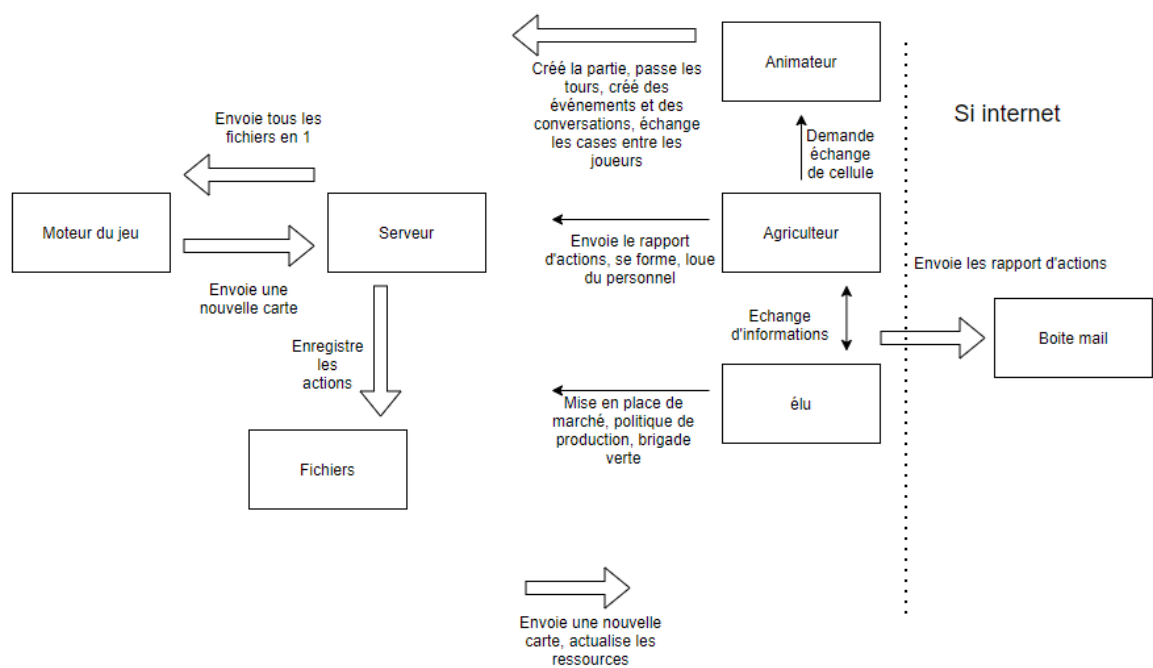


Figure VIII Schéma des différentes communications