

Projet S2 : TheAzureStone

Par le groupe BananaSplit

Adrien Mussat

Jules-Victor Lepinay

Amine Belhadj

Bob De Boisvilliers

Sommaire

1	Introduction	3
1.1	Membres du groupe	3
1.2	Présentation du projet	3
1.3	But du jeu	3
1.4	Contraintes	3
2	État de l'art du projet et points fort	4
2.1	Vue	4
2.2	La carte	4
2.3	Multijoueurs	4
2.4	Système de sauvegarde	4
2.5	L'IA	4
2.6	Gameplay	5
2.7	UI/UX	6
2.8	Site Web	6
2.9	Références et inspirations	7
3	Outils et Ressources	7
3.1	Github	7
3.2	Unity et C#	7
3.3	Photon	8
3.4	Modélisation 3D et effets	8
4	Objectifs	9
4.1	Objectifs pour chaque soutenances	9
4.2	Répartitions des tâches	10
5	Conclusion	10

1 Introduction

1.1 Membres du groupe

Notre groupe *BananaSplit* se compose de 3 membres tous en B1 :

- Chef de projet : Adrien Mussat (adrien.mussat@epita.fr)
- Jules-Victor Lepinay (jules-victor.lepinay@epita.fr)
- Amine Belhadj (amine.belhadj@epita.fr)
- Bob De Boisvilliers (bob.de-boisvilliers@epita.fr)

1.2 Présentation du projet

Pour ce projet de S2, nous avons saisi l'opportunité qui nous était offerte de réaliser un jeu vidéo. Domaine qui nous ai cher étant tous amateur de ces derniers en particulier les jeux de stratégie et de simulation.

Ainsi créer un jeu vidéo en groupe nous permet d'acquérir de nouvelles compétences techniques et développer un esprit de travail d'équipe. Et d'augmenter certaines acquises durant le S1 comme la programmation en C# ou la gestion de logiciel basé sur git.

Nous prévoyons donc ici de faire un jeu de gestion et stratégie basé exclusivement sur une ferme. Le joueur pourra se projeter ainsi pendant quelques heures dans une tout autre vie. Ce jeu s'inspirera sur certains jeux similaires auxquels nous avons joué tel que *Farmville* ou bien *SimCity* mais aussi des jeux mobiles pour leur style à la vue isométrique.

1.3 But du jeu

Dans ce jeu, le joueur intègre donc un fermier dont l'ultime but est de prospérer en gagnant de l'argent via la vente de ses productions.

Pour cela, il a la possibilité : d'embaucher des ouvriers agricoles pour l'aider à travailler dans ses champs et enclos, d'augmenter la taille de sa ferme, de diversifier ses cultures et les rendre plus rentables grâce à un système de recherche, et de choisir de jouer seul ou en coopération.

Comme expliqué plus haut, ce jeu rentre dans les catégories des jeux de gestion et stratégie, il sera donc composé avec une vue à la troisième personne et le contrôle d'un avatar dans le jeu.

1.4 Contraintes

Lors de la production de ce jeu, nous serons confrontés à des problématiques diverses tels que le multijoueurs que nous devons implémenter et faire en sorte que l'expérience des joueurs ne soit pas impactée, nous avons plusieurs pistes de solutions dans la section 2.4) de l'état du projet et points forts ainsi que dans la partie 3, 3.3) Photon. Le jeu bien qu'en vue isométrique, les textures seront modélisées en 3D ce qui implique une gestion des textures plus complexe que des textures plane 2D, mais qui permettent de pouvoir mouvoir des avatar sur une carte plus facilement.

2 État de l'art du projet et points fort

2.1 Vue

TheAzureStone Ce jouera avec une **vue isométrique** du dessus couplée à un environnement 3D : ces caractéristiques vont permettre de donner un rendu stable et net de toute la carte, ce qui s'applique parfaitement au type jeu envisagé. En effet le point de vue isométrique fournit un rendu plus homogène entre les bâtiments qui sont générés en 3D ainsi qu'au joueur de mieux discerner les textures.

2.2 La carte

La carte sera une sorte de plateau pré généré divisé en une multitude de cases où le joueur pourra faire évoluer son exploitation. Il pourra y déplacer son personnage et interagir avec toutes sortes de bâtiments et entités détaillés dans la section Gameplay. C'est également sur cette scène que les divers événements semi-aléatoires auront lieu.

2.3 Multijoueurs

Le multijoueurs constitue dans le projet une des parties les plus importantes, c'est une tâche qui devra donc être traitée en priorité pour assurer une base stable durant le développement. Ici un joueur dit "host" aura la possibilité de créer une nouvelle ou de charger sa partie. Puis d'être rejoint par d'autres joueurs sur cette même partie. De plus sur chaque partie un **chat** textuel sera intégré pour communiquer de façon rapide entre les joueurs sur une même instance.

2.4 Système de sauvegarde

L'implantation d'un tel système est nécessaire pour que le joueur puisse continuer sur son avancée à chaque fois qu'il revient dans le jeu. Sa carte et tout autre avancement doivent donc pouvoir être sauvegardés dans un fichier (local ou externe). C'est pour cela que le joueur sauvegardera sur son ordinateur une sauvegarde la plus récente de la carte. Pour le multijoueur, il y a un système d'hôte et d'invité, ce qui implique que le serveur met à jour la sauvegarde de l'invité.

2.5 L'IA

L'intelligence artificielle sera implémentée autour de 3 axes principaux :

- Le cours du marché : en effet, le prix des ventes des différentes ressources va varier grâce à un système qui prévoit en fonction de l'activité des joueurs. C'est-à-dire de la production en temps réel de la ferme du joueur.
- Les path findings des différentes entités. En effet, pour que le joueur puisse déléguer son travail, il faut avoir différentes entités qui le remplaceront. Elles devront être optimisées dû à leur nombre nécessaire lorsque la ferme se développe.

2.6 Gameplay

Voici une liste de toutes les actions possibles que devrait pouvoir faire le joueur.

- Production :
 - Le joueur doit se déplacer de la ferme au champ ou enclos
 - Attendre dans le champ/enclos un temps donné
 - Récupérer les récoltes obtenu après le temps
 - Mettre les récoltes dans la ferme qui sert d'inventaire
- Construction
 - Avoir dans l'inventaire les ressources nécessaires à la construction
 - Vérifier si l'emplacement choisi est valide
 - Apparition du bâtiment à l'emplacement
- Vendre / acheter
 - Se rendre au bâtiment de l'hôtel des ventes sur la carte
 - Consulter les produits disponibles
 - Pour vendre :

Le joueur devra sélectionner le bouton vendre sur l'interface de l'hôtel des ventes et choisir la ressource à vendre puis déterminer la quantité à vendre. Ce qui en résulte à une augmentation de l'argent présente dans la balance du joueur.
 - pour acheter :

Le joueur devra sélectionner une des ressources dans l'interface de l'hôtel des ventes
choisir la quantité à acheter
vérifier si la balance du joueur est supérieur ou égale au prix affiché
si la balance n'est pas suffisante ne pas acheter et un pop-up arrive
si la balance est suffisante la ressource dans la quantité demandée est transféré dans l'inventaire du joueur la balance du joueur est décrétementée de la valeur affichée dans l'hôtel.
- Bâtiments

Pour les bâtiments, le joueur pourra interagir avec tous ceux qui sont nécessaires au bon fonctionnement d'une ferme et plus avec le système de recherche. Ce système permet de débloquent de nouvelles graines, mais surtout de nouveaux bâtiment de production. Donc nous aurons le bâtiment principal, la ferme mais aussi les champs et enclos ainsi que les ressources avec comme par exemple le blé qui est différent du riz.

- Terrain

Le terrain représente toutes les différentes tuiles qui vont être implémentées pour définir la carte qui sera générée au préalable, ces tuiles permettent aux joueurs ainsi qu'aux IA de pouvoir se déplacer et de se repérer sur la carte. Le terrain sera généré en 3D qui seront homogènes avec la vue isométrique. On aura donc pour tuile, des sources d'eau (rivières/point d'eau), avec aussi les bordures de la carte qui sont représentées par des tuiles spéciales qui ne sont pas utilisées ailleurs pour que le joueur comprenne rapidement leur sens (montagne/océan), enfin les tuiles les plus utilisées avec des couleurs qui diffèrent, les plaines.

- PNJ/animations/IA

Pour tout ce qui est de l'interaction entre le monde et le joueur nous avons différents types d'assets qui seront modélisés :

- cochons/vaches/moutons lors d'événements spéciaux s'échappent, cherchent les cultures les plus proches et les détruisent
- tracteur du garage aux champs
- ouvriers agricoles
- hôtel de ventes avec une IA de régulation des prix peut déclencher des événements (à voir pour le lié au réel)

- Events

Le jeu sera doté d'événements aléatoires se déclenchant au cours de la partie pour pouvoir casser la routine que le joueur pourrait subir. C'est un élément de gameplay qui rajoute un nouvel élément non-prévisible et donc qui requiert de l'attention et des contres mesures à implémenté par le joueur. Cette liste non-exhaustive présente les différents type de d'événements. donc nous avons comme type, les inondations, les mauvaises récoltes, des changements brusque dans le stock market ou bien la panne de machines importantes.

2.7 UI/UX

Voici une liste de toutes les interfaces attendues pour rendre le gameplay accessible et agréable.

- Menu de connexion et de chargement de la carte et liens externes
- Interface en jeu (argent, levels, etc.)
 - Bandeau en haut de l'écran affichant toutes les informations importantes pour le joueur en un coup d'œil affichant tel que l'heure et la production moyenne.
- Interface pour chaque bâtiment avec des informations pertinentes, propre à chacun.

2.8 Site Web

Le Site Web permettra d'avoir une vue d'ensemble du projet, une liste des nouveautés intégrés, ainsi que d'héberger les dernières versions publiées. Il est cependant à prévoir un éventuel budget pour l'hébergement et l'achat d'un nom de domaine bien qu'Epita fournisse des ressources pour l'hébergement via ses partenariats.

2.9 Références et inspirations

Ce projet prend ses inspirations dans plusieurs jeux de gestions et de stratégies tels que SimCity ou bien Farmville. Les deux jeux ont des principes de jeux que nous allons utiliser qui sont l'utilisation de tuile, ainsi que des bâtiments de tailles variables usant les tuiles comme moyens d'unité de mesure.

SimCity, étant un jeu de gestion à l'échelle d'une ville, est à un niveau de gestion plus poussé que Farmville, en effet, le jeu demande au joueur de calculer chacun de ses agissements, en effet chaque bâtiment à des besoins spécifiques. Les bâtiments représentent des éléments de la ville bien précis comme les habitations, les bâtiments de services avec la caserne des pompiers, la gendarmerie ou bien l'hôpital. Les joueurs de SimCity doivent relier tous leurs bâtiments entre eux via les routes ce qui demande en plus de gérer le trafic.

Farmville est lui orienté vers la production de ressources et le développement de son entreprise. Il implémente une vue symétrique 2D. Ce jeu en premier lieu sur navigateur web en tant qu'Adobe Flash Game puis mobile dans les versions ultérieures est orienté sur le fait de garder le joueur le plus possible sur le jeu pour dépenser de l'argent dans le jeu. En effet, le système du jeu est basé sur l'attente plus ou moins raisonnable de la poussée de certaines graines

De plus, SimCity bien que plus vieux, le premier est sorti en 1989, comparé à Farmville sorti en 2009, est plus complet. Mais il prend aussi ces inspirations d'Animal Crossing avec sa caméra qui suit un avatar sur une carte et son mode coopération où des joueurs peuvent venir sur la carte d'un autre pour y faire des échanges ou bien l'aider dans diverses tâches. Ainsi que du multijoueur de Stardew Valley avec le fait que les joueurs interagissent sur la même carte et peuvent donc s'entraider.

3 Outils et Ressources

Afin de réaliser un projet efficacement et de qualité nous avons listé ici les outils dont nous somme certains de nous servir durant le développement du projet.

3.1 Github

Un projet a plusieurs nécessite obligatoirement un logiciel de gestion de version. Les serveurs Git que propose Github nous semble l'idéal pour notre projet, car il permettent en plus de leur interface graphique la possibilité de déployer et d'automatiser du code (cf. github page & github actions).

3.2 Unity et C#

Le moteur Unity qui est vivement recommandé pour ce projet convient parfaitement par son interface pratique et son potentiel pour exporter rapidement. Le langage C# natif à Unity est donc nécessaire pour développer le jeu.

3.3 Photon

Plusieurs modules d'Unity offrent la possibilité de mettre en place un Network pour jouer. Nous avons choisi d'utiliser l'intégration Bolt ou PUN 2 (Photon Unity Networking) de Photon. Ce dernier permet d'héberger gratuitement des serveurs de jeux et d'y connecter jusqu'à 20 joueurs en même temps. Il est à prendre en compte que dans le jeu l'ensemble des éléments sur la carte doivent être synchronisé, ce qui peut représenter une quantité importante d'informations à transmettre. Il sera donc nécessaire d'optimiser cette tâche le plus possible pour éviter toute latence, qui rendrait l'expérience utilisateur fort désagréable.

Cf. <https://www.photonengine.com/>



Cependant il n'est pas exclure la mise en place d'autres modules pour simplifier ou améliorer l'expérience.

3.4 Modélisation 3D et effets

Pour pouvoir créer/modifier les asset nous utiliseront principalement l'outil **Blender** qui est un projet open-source dont certains membres du groupe sont déjà familiers avec ce qui permettra l'accélération de la mise en place d'un produit correct.

Mais nous feront également appel ponctuellement à d'autres logiciels pour les divers effets, notamment ceux de la **suite Adobe** (Photoshop, Premiere pro, After Effects...) pour la création d'effets et d'animations qui sont indispensables à l'expérience utilisateur ainsi qu'à l'immersion du joueur dans l'univers créé. Et ici encore certains membres du groupe se sont déjà familiarisés avec ces logiciels.



4 Objectifs

4.1 Objectifs pour chaque soutenances

Soutenances et parties	Première soutenance (du 7 au 11 mars 2022)	Soutenance intermédiaire (du 25 au 29 avril 2022)	Dernière soutenance (du 6 au 17 Juin 2022)
Multijoueurs, RSS et Chat	30 à 40 % (Connexion possible et/ou liaison serveur-client)	70 à 80 % (Chat et début de sync. carte)	100%
IA	< 20%	40% (PathFindings)	100%
Gameplay	30 % (Map, premières interactions)	Inventaire, actions (cf 2.4) 75%	100 %
Level Design (Assets, graphismes..)	20 % (Static et non définitive)	60 % (Ajout des animations)	100%
UI	20 %	50%	100%
SiteWeb	20 % (Bêta site de présentation du projet)	60 %	100%

Ce tableau présente une avancée escompté, il sera ré-estimé et comparé avant chaque soutenance. Comme précisé le multijoueur et une base stable sont les priorités du projet pour ne pas à avoir a retravailler ses parties à chaque futur ajout.

4.2 Répartitions des tâches

Tableau de répartition des éléments à développer entre chaque membre. Chaque tâche a été assignée un membre principal à un ou plusieurs membres secondaires, cependant cela n'empêche pas l'apport de travail sur une partie de toute l'équipe. Ce tableau sera d'ailleurs lui aussi comparé à chaque soutenance pour voir la quantité et les facilité/difficulté de chacun pour chaque tâche.

Membres et parties	Jules-Victor	Adrien	Amine	Bob
Multijoueurs, RSS et Chat	Suppléant	Responsable		
IA	Suppléant			Responsable
Gameplay	Responsable	Suppléant		Suppléant
Level Design	Suppléant		Responsable	
UI		Suppléant	Responsable	
SiteWeb		Responsable		Suppléant

5 Conclusion

Ce cahier des charges tente de présenter tous les éléments nécessaires au jeu *TheAzureStone*. Cela permet également de mettre en valeur les éléments importants et les difficultés que nous allons rencontrer lors de son développement, notamment en ce qui concerne le multijoueurs. Cependant rassembler tous nos objectifs va nous permettre d'être motivées à la création de ce projet.

Nous espérons respecter au maximum les délais prévus dans la section 4.

Comme vous avez pu le constater ce projet ne prévoit que très peu de dépense, en effet nous pensons pouvoir le réaliser uniquement via des outils gratuits (du moins pour les étudiants), cependant dans le cas où il faudrait financer un contenu, le budget ne devrait pas excéder les 25 euros en tout.