

Erode

Plan de développement logiciel

Version 1.2

Historique des révisions

Date	Version	Description	Auteur
2017-01-23	1.0	Version initiale du Plan de développement	Mathieu O. Wilson, Alexandre Paquet, Odric Plamondon, Jonathan Sauvé
2017-02-13	1.1	Révision et correction du document en considérant le progrès effectué	Mathieu O. Wilson, Alexandre Paquet, Odric Plamondon, Jonathan Sauvé
2017-04-18	1.2	Revision finale	Mathieu O. Wilson, Alexandre Paquet, Odric Plamondon, Jonathan Sauvé

Table des matières

Table des matières	2
1. Introduction	3
2. Vue d'ensemble du projet	3
2.1 But du projet, portée et objectifs	3
2.2 Hypothèses et contraintes	3
2.3 Biens livrables du projet	4
3. Organisation du projet	4
3.1 Structure d'organisation	4
3.2 Interfaces externes	5
3.3 Responsabilités	5
4. Processus de gestion	6
4.1 Plan de projet	6
4.1.1 Planification des phases	6
4.1.2 Objectifs d'itération & calendrier	6
4.2 Suivi de projet et contrôle	10
4.2.1 Gestion des exigences	10
4.2.2 Contrôle de la qualité	10
4.2.3 Gestion de risque	12
4.2.4 Gestion de configuration	13

1. Introduction

Dans ce document, nous analyserons la vue d'ensemble du projet, l'organisation du projet, ainsi que le processus de gestion. Pour ce qui concerne la vue d'ensemble du projet, nous élaborerons sur le but du projet, sa portée et ses objectifs, les hypothèses et contraintes du projet, ainsi que les biens livrables du dit-projet. Ensuite, en ce qui a trait à l'organisation du projet, nous évoquerons la structure d'organisation de l'équipe de projet, l'interface externe pour communiquer avec les groupes externes et les responsabilités des membres de l'équipe de développement. Finalement, pour la partie sur le processus de gestion, nous caractériserons notre plan de projet et le suivi de projet et contrôle.

2. Vue d'ensemble du projet

2.1 But du projet, portée et objectifs

L'objectif du projet est de produire un jeu vidéo dans le cadre des portes ouvertes de l'école Polytechnique de Montréal. Ce jeu sera joué par les visiteurs de l'événement. Le jeu doit donc être exploré rapidement et présenter une difficulté grandissante au fil du temps. Optimalement, cette dernière doit varier autour de 3 minutes, afin que les visiteurs puissent profiter de leur temps au kiosque au maximum. Il faut aussi que le jeu soit accessible à tous et que sa prise en main se fasse de façon intuitive. La production du jeu sera divisé en plusieurs étapes et il y aura un livrable produit durant chacune d'elle: un prototype, une bêta, puis la version finale.

2.2 Hypothèses et contraintes

Erode comporte quelques hypothèses et contraintes que nous énumérons ci-contre. Tout d'abord, l'équipe de développement du projet est composé de quatre membres, ainsi que d'un client et d'un conseiller/superviseur de projet. Pour ce projet, nous avons comme contraintes physique que le jeu doit s'exécuter sur un ordinateur de bureau et d'être manipulable grâce à un contrôleur de Xbox One. En ce qui a trait à l'échéancier, le prototype du livrable doit être fait pour le 17 février 2017, le livrable bêta doit être remis le 4 avril 2017 et le projet final doit être livré le 18 avril 2017.

2.3 Biens livrables du projet

Plusieurs artefacts seront produits durant le processus créatif de développement du jeu. Il seront présentés au fur et à mesure de leur complétion.

- Schéma de description du processus (13 janvier 2017): Cet artefact présente le processus de développement utilisé dans le développement du jeu.
- Document de design de jeu (20 janvier 2017): Cet artefact présente entre autre le contexte, l'univers et l'ensemble des mécaniques du jeu.
- Plan de développement logiciel (27 janvier 2017): Cet artefact présente les différents artefacts qui seront produits avec leurs échéanciers, ainsi que les stratégies de gestion de risque qui seront employées.
- Document de traçabilité des exigences (17 février 2017): Cet artefact a pour but de garder la trace de l'état de complétion des différents requis.
- Document d'architecture (17 février 2017): Cet artefact présente l'architecture du jeu. La logique d'application et de déploiement sera détaillée.
- Prototype (17 février 2017) : Version de démonstration du jeu. Cette version sert à présenter au client la base du jeu sur laquelle il sera itéré.
- Version Bêta du jeu (5 avril 2017) : Version bêta du jeu qui doit être remise avant de faire les retouches finales. La version bêta doit contenir toutes les fonctionnalités du jeu.
- Produit final (18 avril 2017) : Remise de la version finale du jeu au client. Celle-ci contient toutes les fonctionnalités du jeu, le son, la musique, de belles animations et des effets de qualité. De plus, il faut remettre les versions finales des différents documents liés à Erode.

3. Organisation du projet

3.1 Structure d'organisation

L'équipe est composée de quatre membres comme il a été dit plus haut: Alexandre, Jonathan, Mathieu et Odris. Quant à la direction, elle est composée du client (Laurent Tremblay) que nous rencontrons chaque semaine. Puis, Olivier Gendreau est l'autorité responsable de la supervision du projet que nous rencontrons aussi chaque semaine. Finalement, Maude St-Cyr Bouchard est notre conseillère en habiletés relationnelles.

3.2 Interfaces externes

Groupes externes	Contact externe	Contact interne
Client	Laurent Tremblay	Mathieu O. Wilson
Superviseur	Olivier Gendreau	Odric Plamondon

L'équipe communique avec le client de façon régulière. Puisque le client se retrouve à l'intérieur de l'école Polytechnique, il est facile de se rencontrer de façon informelle. De plus, une rencontre formelle est organisée une fois par semaine ou deux afin de s'assurer de respecter la vision que le client a du projet.

L'équipe communique avec le superviseur de façon régulière. Une rencontre formelle hebdomadaire est organisée avec le superviseur afin d'assurer un bon suivi des exigences relatives au cours du projet intégrateur 4.

3.3 Responsabilités

Les membres de l'équipe de développement possèdent des responsabilités. Tout d'abord, Alexandre est le "Scrum master", il est responsable d'animer les rencontres et de se charger que l'information se propage entre tous les membres, ainsi que de s'assurer de l'avancement de chaque membre afin de respecter l'échéancier. Jonathan s'occupe majoritairement de la documentation. Il s'assure que le produit développé respecte le plan établi et il s'assure que le projet respecte les requis établis par le client. Mathieu a la responsabilité de communiquer avec Laurent, notre client et analyste, pour toutes questions de projet ou encore pour des demandes d'équipes, ainsi que de lui communiquer l'avancement du projet. Finalement, Odric s'assure que les membres de l'équipe ont le matériel nécessaire à leur disposition pour travailler convenablement. De plus, il gère les différents outils pour la configuration et les versions (comme Git et Redmine).

4. Processus de gestion

4.1 Plan de projet

4.1.1 Planification des phases

Phase	Date de remise
A - Documentation initiale	27 janvier 2017
B - Prototype et documentation	20 février 2017
C - Beta	5 avril 2017
D - Produit final	18 avril 2017

4.1.2 Objectifs d'itération & calendrier

La phase A a déjà été effectuée en date de remise de ce document. La lettre ou les lettres entre parenthèses indiquent dans quelle(s) phase(s) se trouve l'itération.

Numéro d'itération	Date	Tâches
1 (B)	30/01/17	Personnage <ul style="list-style-type: none">- Contrôles du personnage en 2D- Contrôles de Caméra Plateforme <ul style="list-style-type: none">- Faire grille de jeu- Mécaniques de plateforme<ul style="list-style-type: none">- Points de vie- Destruction Obstacles <ul style="list-style-type: none">- Comportement astéroïde Interface <ul style="list-style-type: none">- Menu principal- HUD<ul style="list-style-type: none">- Temps- Score HUD

2 (B)	06/02/17	Personnage <ul style="list-style-type: none"> - Attaque de base - Armes personnage Plateforme <ul style="list-style-type: none"> - Réparation de la plateforme Ennemis <ul style="list-style-type: none"> - Implémenter Ennemi Hunter <ul style="list-style-type: none"> - AI - Attaque - Effet sur le joueur - Points de vie Interface <ul style="list-style-type: none"> - Menu in game Général <ul style="list-style-type: none"> - Tomber de la plateforme - Effet de défaite Obstacles <ul style="list-style-type: none"> - Astéroïde <ul style="list-style-type: none"> - Effets sur le joueur - Effets sur ennemis - Comportement - Collision Marteau
3 (B)	20/02/17	<ul style="list-style-type: none"> - Finaliser prototype - Compléter documents Personnage <ul style="list-style-type: none"> - Attaque chargée - Blitz - Spin Obstacles <ul style="list-style-type: none"> - Comète <ul style="list-style-type: none"> - Comportement - Avertissement - Collision plateforme - Trou noir <ul style="list-style-type: none"> - Comportement - Vent Solaire <ul style="list-style-type: none"> - Animation - Effets sur les objets - Effets sur les ennemis - Effets sur le joueur? Power-ups <ul style="list-style-type: none"> - SuperSpeed <ul style="list-style-type: none"> - Déplacement du joueur modifié

		<ul style="list-style-type: none"> - Effets visuels - Multiplicateur de score - Effets visuels - Effet sur le score Interface <ul style="list-style-type: none"> - Leaderboard
4 (C)	27/02/17	Ennemis <ul style="list-style-type: none"> - SHOOTER <ul style="list-style-type: none"> - Trouver modèle - Faire AI - Trouver animations - Attaque (projectile) - Effet sur le joueur - Points de vie - HUNTER <ul style="list-style-type: none"> - Trouver modèle - Trouver animations Power-ups <ul style="list-style-type: none"> - Ralenti du temps - Logique - Visuel HUD (Vignette)
5 (C)	06/03/17	Plateforme <ul style="list-style-type: none"> - Texture - Effets Visuels Obstacles <ul style="list-style-type: none"> - Champ électromagnétique <ul style="list-style-type: none"> - Modèle - Animation - Points de vie - Effet sur le joueur - Comète <ul style="list-style-type: none"> - Effets sur le joueur - Effets sur ennemis - Effets sur obstacles Power-ups <ul style="list-style-type: none"> - Super marteau <ul style="list-style-type: none"> - Effets visuels - Effets sur ennemis - Effets sur astéroïdes - Personnage <ul style="list-style-type: none"> - Recharge Laser / munitions

6 (C)	13/03/17	Obstacles <ul style="list-style-type: none"> - Comète <ul style="list-style-type: none"> - Modèle - Animation - Astéroïde <ul style="list-style-type: none"> - Modèle - Effets sur ennemis - Effets sur Obstacles - Condition de destruction - Trou noir <ul style="list-style-type: none"> - Modèle - Animation Ennemis <ul style="list-style-type: none"> - CHARGER <ul style="list-style-type: none"> - Trouver modèle - Faire AI - Trouver animations - Attaque - Effet sur le joueur - Points de vie
7 (C)	20/03/17	Obstacles <ul style="list-style-type: none"> - Barrière <ul style="list-style-type: none"> - Animation - Déplacement - Effets sur le joueur - Effets sur ennemis - Effets sur Obstacles Ennemis <ul style="list-style-type: none"> - Onyx <ul style="list-style-type: none"> - Trouver modèle - Faire Comportement - Trouver animation - Mécanique du mouvement en S vertical - Points de vie - Gérer évènement pour apparition
8 (C)	27/03/17	Ennemis <ul style="list-style-type: none"> - Logique de spawn Général <ul style="list-style-type: none"> - Choisir modèles (perso) - Mécanique de progression (basique) - Musique de background - Calcul de score

		Personnage - Animations du personnage
9 (D)	05/04/17	Ajuster mécanique de progression Ajuster Logique de Spawn Sons
10 (D)	10/04/17	Ajuster mécanique de progression Ajuster Logique de Spawn Sons
11 (D)	17/04/17	Finaliser remise

4.2 Suivi de projet et contrôle

4.2.1 Gestion des exigences

Les rencontres avec le client se feront environ à chaque deux semaines, de façon à connaître les exigences qui seront à faire rapidement. Les changements commenceront à être appliqués la semaine même ou la semaine suivante. Si la nouvelle exigence demandée nous semble trop majeure, nous négocierons le changement de celle-ci et nous trouverons un compromis avec le client. Sinon, nous l'intégrerons dans notre planification de projet, et l'ajouterons comme tâche à faire dans l'outil Trello.

4.2.2 Contrôle de la qualité

Pour s'assurer de la qualité de notre jeu, nous suivrons le modèle de qualité de [McCall's](#) et tenterons de respecter chacune des facettes de ce dernier. Voici donc les différents critères ainsi que la ou les actions à entreprendre lorsqu'il n'est pas respecté. Il faut savoir que chaque critère sera vérifié à la fin de chaque itération.

Critère	Description	Actions à entreprendre
Maintenabilité	Il est facile de repérer et de réparer un bogue.	- Refactoriser le code - Documenter le code
Flexibilité	Il est facile d'effectuer des changements rapides, à la demande du client	- Effectuer un meilleur découplage du code source
Testabilité	Il est facile de vérifier que le comportement du logiciel correspond aux exigences	- Effectuer un meilleur découplage du code source - Revoir la structure interne

		du logiciel
Portabilité	Il est facile de démarrer le logiciel sur différente plateforme	- Tester le logiciel sous différentes machines et versions de Windows
Réutilisabilité	Chaque module est facilement transférable à un autre environnement	- Éliminer l'interdépendance entre les modules
Interopérabilité	Il est facile d'ajouter des fonctionnalités ou des modules au logiciel	- Refactoriser le code - Réduire le couplage entre les modules
Exactitude	Les fonctionnalités correspondent aux exigences	- Refactoriser le code - Modifier la fonctionnalité - Ajuster l'exigence
Fiabilité	Le logiciel n'arrête pas de fonctionner dans des conditions d'utilisation normales	- Effectuer des "edge cases" plus poussés
Efficacité	Le logiciel ne dépense pas plus de ressources qu'il ne le devrait	- Refactoriser le code pour identifier les fuites de mémoire éventuelle - Limiter le nombre d'opérations effectuées à chaque trame (frame)
Intégrité	Non-applicable	-
Utilisabilité	Le logiciel est facile à utiliser par un utilisateur	- Refactoriser l'interface utilisateur - Réduire la charge cognitive

4.2.3 Gestion de risque

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Travail mal réparti	E	Retard du projet, éléments du jeu incomplets	Tenter de prendre de l'avance lorsque possible, et regarder à chaque semaine la progression du projet et des tâches restantes.
6	Sous-estimation du temps requis pour certaines tâches	E	Retard du projet, Diminution de la qualité du jeu	Tenter de prendre de l'avance lorsque possible, et regarder à chaque semaine la progression du projet et des tâches restantes.
2	Disparition d'un coéquipier	C	Retard du projet, nouvelle planification de tâches nécessaire	Avertir les autres coéquipiers d'avance, ainsi que le professeur si possible.
6	Changement des exigences du client	M	Nouvelle planification de tâches nécessaire	Avoir des rencontres régulières avec le client afin d'éviter les surprise et pour prévoir à l'avance.
3	Dédoubllement du travail	M	Ralentissement de la production du jeu	Assigner les tâches à une personne en particulier dans un document.

4.2.4 Gestion de configuration

Pour ce qui est de la gestion de configuration, nous utiliserons principalement [Trello](#) pour gérer le développement de notre projet. Cet outil est parfait pour les projets suivant un processus agile, puisqu'il est simple d'utilisation et il est facile d'y représenter nos itérations. Lorsque nous rencontrerons des problèmes ou bien des changements, il suffira de les ajouter à l'itération en cours ou à la suivante. De cette façon, toute l'équipe sera au courant et la tâche sera ajoutée aux tâches courantes.

Les artefacts du projet seront nommés de façon courte et représentative. Avec le nom de l'artefact, il doit être possible de savoir exactement de quoi on parle sans aller voir la documentation. Pour ce qui est de la nomenclature elle-même, le nom de chaque artefact commencera avec une majuscule. S'il est composé de plus d'un mot, nous suivrons la règle du "Camel Case", où chaque début d'un mot commence avec une majuscule. Les noms de méthodes, quant à elle, commenceront avec une minuscule et suivront ensuite la même règle. Il en va de même pour les attributs de classe.