

Journal de Bord

Date : 25/05/2020

- Lecture de l'énoncé
- Contacte avec le formateur pour clarifier certain points
- Recherche d'exemples du jeu Donkey Kong, pour utiliser comme base
 - Trouvé : <https://www.mysteinbach.ca/game-zone/64/donkey-kong/>
- Rédaction du planning prévisionnel
- Rédaction du début de la documentation
- Recherche et préparation des sprites liés aux jeux

11:36 : Ajout fichier mapping touches bornes dans drive/ressources (s. Garchery)

TODO :

- Finir la préparation des sprites
- Créer le menu d'accueil
 - Sprites
 - Peut-être sons
 - Fonctionnalités -> changement d'états
- Commencer la partie visuel du jeu

Date : 26/05/2020

- Continuation de la préparation des sprites
 - Source sprites
 - <http://www.mariouniverse.com/sprites-nes-dk/>
 - Découpage
 - Fond transparent
 - Découpage en groupe pour les animations
- Regardé un tutoriel sur comment animer les sprites
 - Source :
 - <https://www.youtube.com/watch?v=OLSiWxgONeM>
 - https://github.com/Oyyou/MonoGame_Tutorials/tree/master/MonoGame_Tutorials/Tutorial011
 - Implémentation réussi
- Regardé un tutoriel sur comment avoir plusieurs états dans le projet
 - Source :
 - <https://www.youtube.com/watch?v=76Mz7CIJLoE>

- https://github.com/Oyyou/MonoGame_Tutorials/tree/master/MonoGame_Tutorials/Tutorial013
 - Implémentation réussi
 - Le menu et le jeu sont deux états différents
- Regardé un tutoriel sur comment avoir un button
 - Source :
 - https://github.com/Oyyou/MonoGame_Tutorials/tree/master/MonoGame_Tutorials/Tutorial012
 - Pas utilisé.
- Décision -> Structurer les sprites en 3 types.
 - Sprites génériques -> Non animés
 - Sprites animés
 - Sprites animés qui bouge
 - Quelques problèmes de logique + héritages
 - Majorité des problèmes résolus
 - Dernier problème
 - Solution trouvée mais le code est un peu "bricolé", à changer
- Menu créé et fonctionnel
 - Sauf les sons
- Regardé un tutoriel sur comment implémenter de la musique et des bruitages
 - <https://riptutorial.com/monogame/example/16734/sounds-using-soundeffect>
 - <http://rbwhitaker.wikidot.com/playing-background-music>

TODO :

- Sons menu ou laisser pour après. Faire au même temps que le jeu?
 - Peut être mieux le faire maintenant pour ne pas avoir des problèmes de logique plus tard
 - Rechercher les sons du jeu
- Avancer Documentation
 - Peut être améliorer le planning
 - Avancer la doc technique
 - Procédure de tests avancer commencer le jeu

Observations :

- Planning prévisionnel peut être pas très bien pensé
 - En pratique l'ordre des étapes a changé un peu

Date : 27/05/2020

- Recherche de musiques et bruitages pour le menu et jeu
 - Sources :
 - <https://downloads.khinsider.com/game-soundtracks/album/donkey-kong>
- Implémentation de la musique de fond dans le menu
- Implémentation de la logique pour les bruitages

- A améliorer, action chaque X secondes au lieu de tout le temps
- Page d'informations supplémentaire
 - Recherche de sprites
 - Source :
 - <https://www.pngfuel.com/free-png/nsvqq>
 - Travail sur les sprites
 - Background, etc
 - Transitions entre menu et page d'infos
- Documentation
 - Analyse Fonctionnelle
 - Fonctionnalités
 - Maquettes pour les interfaces
 - Cas d'utilisation
 - Pas sur des cas pour un jeu, demander
- Retard sur le commencement du visuel du jeu, pas les fonctionnalités juste visuel

TODO :

- Visuel jeu
 - Logique pour le terrain

Date : 28/05/2020

- Placement du terrain
 - Non-incliné pour l'instant
- Collisions de Mario
 - Avec la zone de jeu
 - Avec le terrain
 - Problèmes rencontrés
 - Regardé un tuto
 - <https://www.youtube.com/watch?v=CV8P9aq2gQo>
 - https://github.com/Oyyou/MonoGame_Tutorials/tree/master/MonoGame_Tutorials/Tutorial009
 - Problèmes avec la width de Mario, les sprites de l'animation de mario ont des widths différentes
 - Regardé tuto sur la collision par pixel
 - Implémenté, plus tard je me suis rendu compte que cela n'était pas nécessaire car le jeu ignore toutes les collisions avec le terrain sauf les pieds, donc enlevé du programme
 - De nouveau rencontré des problèmes avec la logique
 - Solution : Traiter les "lignes" du sol par niveaux.
 - Exemple : Si Mario n'as pas monté les échelles il est encore au "niveau" 1 donc ignorer le reste du sol
 - Cela pourra créer des problèmes plus tard avec la transition de niveaux, à faire attention
- Échange d'idées avec M.Russo

- Saut de Mario
 - Pas très élégant, à retravailler
- Restructuration du code
 - Enlevé le code + nécessaire grâce au système de niveaux mentionné avant
- Retravaillé les sprites de Mario
 - Moins de problèmes avec les collisions

TODO :

- Implémentation échelles
 - Ajouter les animations de Mario
 - Changement de niveaux avec les échelles
- Vérification du niveaux concernant la position Y de Mario
 - Au cas où il descend d'un niveau à l'autre
- Ajouter Kong au mode de jeu
 - Commencer le lancement de tonneaux par Kong
 - Collisions concernés
- Avancer documentation

Date : 29/05/2020

- Continuation saut de Mario
 - Saut = 1 image
 - Marcher = 3 images
 - Problème avec le "AnimationManager"
 - Solution : transformer le saut en 2 images et ignorer la deuxième
 - A retravailler si le temps le permet
- Logique pour le terrain incliné
 - Génération d'une ligne à la fois
 - Point de référence → brique toute en bas à gauche
 - Ajout d'une hauteur à la brique suivant pour créer une inclinaison
 - Ou enlever si l'inclinaison est inversé
- Logique pour les échelles
 - Trop de variables
 - Plusieurs tailles
 - Trous à différents endroits
 - Placements semble aléatoires
 - Difficulté pour avancer
 - Demande de conseils au formateur
 - Décision : Laisser pour demain
- Logique pour les tonneaux
 - Un tonneau qui descend le terrain
 - Problème de logique avec le changement de vitesse (changer de sens)
 - Liste de tonneaux
 - Création avec un interval

- Problème avec la position Y de mario
 - Tout sorte de problèmes avec les collisions
 - Refactorisation du code
- Passage des différents "niveaux" du terrain
 - Pas implémenté complètement, mais les collisions fonctionnent si le niveaux est proprement changé
- Portion de la journée dédié à la documentation n'as pas eu lieu
 - Utilisé ce temps pour déboguer les différents erreurs de mario

TODO :

- Implémenter les échelles
 - Détection de changement des niveaux du terrain
- Rattrapper la documentation
- Ajouter Kong

Date: 02/06/2020

- Ajouté Kong
 - Animations ajouté mais la deuxième n'as pas encore de déclencheur dans le jeu, j'utilise la touche Q pour la déclencher
 - Ajouté le groupes de tonneaux à gauche de Kong
- Ajouté la Princesse et son animation
- Ajouté une "plateforme" pour le placement de la princesse
- Changement de l'inclinaison pour la première plateforme
- Retravail du saut de Mario
 - Plus fluide, moins hardcodé
 - Un système de gravité a été ajouté
- Événement de destruction d'un tonneaux
 - Collision avec le tonneaux bleu en bas à gauche
- Changement de logique définitive pour les collision entre les plateformes et Mario
 - Ce changement a été effectué pour faciliter le codage des échelles que j'estime sera compliqué
 - Système de niveaux, enlevé
 - Fonctionnel, il reste qu'ajouter les échelles pour passer d'une plateforme à l'autre
 - Débogage
 - L'implémentation m'as pris beaucoup de temps
 - Beaucoup d'erreurs de logique que n'avaient pas été des soucis avant
 - Difficulté à trouver des solutions pour ces problèmes
 - Échange d'idées avec M. Russo
 - Solution trouvé
- Implémentation de l'ergonomie pour la borne
 - Tous les boutons peuvent être utilisés
- Changement de la taille de l'écran de jeu

TODO

- Echelles
- Lier animation Kong à la création de tonneaux
- Documenter (proprement) le code

Date: 03/06/2020

- Documentation méthodes importantes
 - Exemple : GroundLayoutSpawn()
- Vidéo conférence avec les experts
 - Discussion sur le cahier des charges
 - Doutes par rapport au cahier des charges enlevés
- Ajout des bruitages pour Mario
- Recherche des tutos sur comment faire la logique pour les échelles, aucun utilisé mais certains aspect de chacun ont été utilisés
 - Exemple tuto
https://www.reddit.com/r/godot/comments/d9mi3c/dev_blog_01_creating_ladders_in_godot/
 - Difficulté à trouver une bonne logique pour comment procéder
 - Idée
 - Séparé une échelle en plusieurs pour mieux la pouvoir adapté aux différents espacements entre chaque plateforme
 - Avoir une hitbox qui occupe les plusieurs sprites
 - Implémenté
 - Mais problèmes avec la collision
 - Debug pendant toute l'aprem
 - Pas compris le problème
 - Hauteur de l'hitbox est correct avec console.WriteLine, mais la collision s'arrête à la l'hauteur de la première sprite

TODO :

- Trouver le problème avec les échelles
- Sons et bruitages pour kong
- Animation de monter les échelles de mario + bruitage
- Lier animation Kong à la création de tonneaux

Date: 04/06/2020

- Problème des échelles très simple, oublié simplement de compenser la position Y par rapport aux nombre de petites échelles
 - Ajout de la logique pour la création des échelles
 - Changement de la logique des échelles, le soucis mentionné avant n'est plus pertinent.
- Recherche de sons et bruitages pour Kong

- Préparation des sons (convertir en .wav)
 - <https://online-audio-converter.com/>
- Enlever les morceaux non nécessaires
 - <https://audiotrimmer.com/>
- Logique pour la liaison de l'animation de Kong à la création des tonneaux
- Logique, animations et bruitages nécessaires pour que mario puisse monter les échelles
 - Tuto pour les sons :
<https://community.monogame.net/t/how-do-you-make-monogame-play-a-sound-on-a-key-press-and-stop-as-soon-as-you-stop-pressing-that-key/9958/2>
- Retraitement de sprites déjà dans le jeu, certaines animations avaient quelques bugs

TODO:

- Trouver une musique de fond pour le jeu
- Avoir une condition de victoire (collision avec la princesse)
- Système de vies
- Gestion du score
- Sauvegarde du highscore
- Quelques sound effects ont un bruit de fond, chercher des alternatives

Date: 05/06/2020

- Système de vies
 - Logique
 - Affichage
 - Recherche + traitement de sprites
 - Condition de perte
- Collision de Mario avec les tonneaux
- Gestion du score
 - Compteur du temps de jeu, classe GameTimer
 - Tuto suivi : <https://www.youtube.com/watch?v=-2FeSrYT1KE>
- Sauvegarde des highscores en XML
 - Classes : Score, ScoreManager
 - Tuto suivi : <https://www.youtube.com/watch?v=JzEwVCgALuY>
- Affichage du score et highscore
 - Avec une spriteFont
 - Font utilisé : Arcade Classic
 - Source : <https://www.dafont.com/arcade-classic-2.font>
 - Tuto pour mesurer un string dessiné :
<https://gamedev.stackexchange.com/questions/19438/how-do-i-get-the-height-of-an-xna-spritefont>
- Collision avec la princesse
 - Condition de victoire
- Condition de perte + victoire

- Transition vers le menu
- Timer d'attente
 - Si gagnant affiche "You Win" et le score + une musique de fond
 - Si perdant affiche "You Lost" + une musique de fond (différente de celle de victoire)
- Recherche de sons pour remplacer ceux qui le nécessite
- Travail sur certains sons
 - Préparation des sons (convertir en .wav)
 - <https://online-audio-converter.com/>
 - Enlever les morceaux non nécessaires
 - <https://audiotrimmer.com/>
- Te

TODO :

- Avancer Documentation
- Ajouter testes unitaires
- Documenter proprement le code
- Jouer avec le volume des sons, certains sons sont très forts comparés aux autres
- Si le temps le permet, retravailler page d'infos

Date: 08/06/2020

- Vérification de toutes les fonctionnalités du programme pour assurer leur fonctionnement
- Avancement de la doc
 - Méthodologie
 - Planification
 - Analyse Organique jusqu'à explication des classes
- Doc utilisateur

TODO :

- Finir la documentation technique
 - Mise en page de la doc technique
 - Résumé du tpi
 - Assurer rapidement le fonctionnement des fonctionnalités
 - Export code source
-

Date: 09/06/2020

- Export code source
 - Résolution de problèmes
 - Utf8
 - Projet trop grand
- Fin de la doc technique
 - Fin de l'analyse organique
 - Conclusion
 - Bibliographie
 - Source de données
 - Glossaire
- Mise en page de la doc technique
- Résumé du TPI