GAMING CAMPUS

POUR LES PROS

CAHIER DES CHARGES

Formation Développeur de Jeux Vidéo Pro

David MEKERSA / Nicolas POINTET

(Apprentissage par Projet / © Active Learning by Gaming Campus)

NOM DU PROJET : Construire la partie Front de votre projet

Numéro: Project N°1

Discipline(s) / Enseignement(s) couvert(s) par le projet :
Développement de Jeux Vidéo

Nom du professionnel / intervenant :
David Mekersa / Nicolas Pointet

Modalités d'apprentissage :
Cours théoriques et pratiques
Apprentissage par projets
Cours à distance, le soir

Intitulé, contexte et descriptif du projet (intitulé, contexte, périmètre) :

Contexte :

Projet :
Création d'un Jeu vidéo 2D avec un langage de Script. L'apprenant doit réaliser un jeu en vue de dessus, mettant en œuvre 5 concepts fondamentaux de la programmation de jeux vidéo :

1- *Les listes*
2- *Les angles et rotations*
3- *La modularité*
4- *Les fonctions*
5- *Les machines à états*

Exemples de Gameplays :

Hotline Miami
Geometry War
Guncar Arena
Vindicators
Blasteroid

Technologies :
Lua (langage)
Love2D (framework)
Visual Studio Code (IDE).

Contraintes :
Pas d'utilisation de librairies externes.

A l’issue du projet, l’étudiant sera capable de :

- *Implémenter des fonctionnalités de jeu en utilisant un langage de script (Le Lua).*
- *Programmer des déplacements et des rotations à 360° pour les personnages et les objets du jeu en utilisant les fonctions de trigonométries telles que sinus, cosinus, calcul d’angles et de distances.*
- *Définition de conditions de victoire et de défaite pour le jeu, en prenant en compte les objectifs du joueur et les événements qui se produisent dans le jeu.*
- *Mise en œuvre de la modularité dans le code du jeu, en utilisant les modules Lua pour structurer le code de manière logique et réutilisable.*
- *Découpage du jeu en différentes scènes (par exemple, menu principal, gameplay, écran de pause, victoire, défaite), en utilisant des techniques de gestion de l'état du jeu.*
- *Gestion d'une liste d'entités du jeu (par exemple, personnages, objets, ennemis, etc.), en utilisant des listes pour stocker et gérer ces données.*
- *Utilisation de fonctions pour gérer les relations entre différentes parties du code du jeu et structurer / découper le code.*
- *Utilisation d'une machine à état fini (MEF) pour gérer les états et les transitions du comportement des PNJ ou des ennemis dans le jeu. Cela peut inclure la définition de différents états (par exemple, "idle", "marche", "attaque", "fuite") et de transitions entre ces états, en fonction des actions du joueur ou de l'environnement.*

ECHÉANCES ET LIVRABLES

Dates échéances	**Livrables attendus**	**Moyens / formats** (comment on transmet le livrable)
Dernière semaine du module	**Le code du projet**	Dossier du projet sous forme d'archive, ou Github

Planning

4 semaines de coaching
2 semaines d'accompagnement projet
1 semaine de rendu de projet (soutenance)

Ce planning peut être amenés à évoluer en fonction des contraintes des intervenants.

Pré-Requis au projet :

Avoir suivi la formation de mise à niveau.

Atelier : Langage et programmation pour le jeu vidéo

Atelier : Pixels et origine

Atelier : Les bases d'un Lunar Lander en Love2D

Atelier Jeu de Pong : Apprenez à coder votre premier jeu vidéo rapidement et facilement

Atelier : Coder un inventaire de RPG avec une liste

RESSOURCES

Ressources pour le travail préparatoire :

Ateliers à suivre obligatoirement :

- Atelier : Donner vie à une horde de zombis (Les bases de l'Intelligence Artificielle)
- Atelier DLC 115 : Faire spawner des ennemis à intervalles réguliers
- Atelier DLC 116 : Programmer le comportement des ennemis
- DLC 142 – TANK WAR – Tirs orientés
- Atelier : Le secret des Tilemaps pour créer des niveaux
- https://www.gamecodeur.fr/dlc-76-attrape-moi-si-tu-peux-se-faire-poursuivre-par-un-ennemi/

Et tous les cours qui pourraient vous aider sur :
https://www.gamecodeur.fr/toutes-les-categories/

Exercices utiles :
Enoncés et solutions sur :
https://www.gamecodeur.fr/programmation-lua-love2d-10-super-exercices-de-programmation-pour-sexercer-et-progresser/

👉 Semaine 1 :

- Ping Pong (pour les déplacements)
- Les cœurs (pour les boucles)

👉 Semaine 2 :

- La sulfateuse (pour les tirs)

👉 Semaine 3 :

- Cap Canaveral (pour les états)

Outils :

https://code.visualstudio.com/

ACQUIS D’APPRENTISSAGE DÉTAILLÉS

CONNAISSANCES À ACQUÉRIR :

- *Connaissance du vocabulaire professionnel technique*
- *Connaissance d’un environnement de développement*
- *Connaissance des bonnes pratiques de la programmation*
- *Connaissance de l’architecture d’un programme de jeu vidéo*
- *Connaissance d’un langage de script*
- *Connaissance d’un Framework de programmation de jeux vidéo*
- *Connaissance des formats de fichiers appliqués au jeu vidéo*
- *Connaissance des concepts fondamentaux de programmation de jeux vidéo*

COMPÉTENCES À ACQUÉRIR :

- *Mettre en place un environnement de travail pour programmer un jeu vidéo*
- *Analyser et formaliser les fonctionnalités à programmer pour créer le jeu vidéo*
- *Programmer l'ensemble des algorithmes du jeu vidéo*
- *Déployer le jeu vidéo et contrôler son fonctionnement, corriger et faire évoluer ses fonctionnalités*
- *Utilisation de techniques et outils de débogage*

Grille d’évaluation du projet

Critères d’évaluation obligatoires
Découpage en modules : Le code est découpé en modules pour faciliter la compréhension, la réutilisation et la maintenance du code
Utilisation des machines à états : L’apprenant intègre une ou plusieurs machines à états pour gérer les comportements des éléments et/ou les états du jeu (Menus, gameplay, etc.)
Rotations et visées : L’apprenant maitrise les fonctions trigonométriques de bases pour le déplacement et la rotation des éléments de gameplay (Sinus, cosinus, distance, calcul d’angles)
Utilisation des listes : Le jeu intègre l’utilisation des listes pour gérer les différents éléments du jeu (objets, ennemis, etc.) et pour faciliter la création de mécaniques de jeu complexes
Multiples scènes : Le jeu comporte plusieurs scènes (Menu, gameplay, pause, etc.)
Fonctions : L’apprenant utilise des fonctions pour structurer et découper son code, et utilise les paramètres (valeurs, références).

Critères d’évaluation secondaires
Clarté et commentaires du code source : Le code est organisé et commenté de manière à faciliter la lecture et la compréhension
Fonctionnalité et stabilité : Le jeu fonctionne sans bugs ni plantages, permettant une expérience fluide
Présentions claire du projet : L’apprenant peu faire une description structurée de son projet et expliquer les choix de conceptions
Maitrise technique et conceptuelle : L’apprenant démontre, en répondant à des questions, qu’il maitrise les concepts (fondamentaux : boucles, expressions, structures de contrôles, fonctions) et le langage utilisé dans son projet

Concernant la Machine à Etats (MAE) :

La MAE est un critère de validation de la certification, vous devez donc l'implémenter obligatoirement et adapter votre gameplay en conséquence.

- Implémentez au moins 4 états au sein de la même MAE et elle ne doit pas être linéaire.
- Une MAE est un concept formel, pas un truc qu'on improvise. La formation "https://www.gamecodeur.fr/liste-ateliers/atelier-ia-love2d" est FONDAMENTALE pour apprendre à implémenter la MAE de manière SIMPLE tout en étant conforme au concept. Suivez cette formation obligatoirement.
- Conceptualisez votre MAE par écrit, en réalisant le schéma qui vous est enseigné dans la formation.

Une MAE ce n'est pas que pour des ennemis. On peut tout automatiser avec une MAE. C'est ça qui en fait un outil de programmation indispensable dans les jeux vidéo.

Voici un exemple de MAE pour des moissonneuses, en 50 LIGNES DE CODE :
https://www.dropbox.com/t/SUVss0HLzoAPWnHV
(je ne vous donne volontairement pas le projet complet)

- La moissonneuse cherche une cible à moissonner (SEEK)
- Si elle trouve une cible elle va se diriger vers elle (GOTO) sinon elle rentre à la base (RETURN)
- Si elle a atteint la cible, elle moissonne (PICK)
- Si elle est pleine ou si la cible est vide, elle rentre à la base (RETURN)
- En mode RETURN elle livre les ressources dans le silot (vault) et repart au travail (SEEK)