



G A M I N G C A M P U S

CAHIER DES CHARGES

Formation Développeur de Jeux Vidéo Pro

David MEKERSA

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

NOM DU PROJET : Construire la partie Front de votre projet

Numéro: Project N°2

Date:

Formation / Promotion : Dev JV Durban

- **Discipline(s) / Enseignement(s) couvert(s) par le projet :**

Développement de Jeux Vidéo

- **Nom du professionnel / intervenant :**

David Mekersa

- **Profil LinkedIn :**

<https://www.linkedin.com/in/dmekersa/>

- **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 Orienté Objet. L'apprenant doit réaliser un jeu de casse brique :

Le jeu doit proposer :

- 1. Plusieurs niveaux (chargés depuis un/des fichiers externes)*
- 2. Une architecture objet avec héritage*
- 3. Un Service Locator maison (pas celui de Monogame)*
- 4. Un gameplay très éloigné du casse brique classique*
- 5. Optionnellement : un éditeur de niveaux*

Exemples de Gameplays :

[Breakoid](#) de Ced30 (membre de la 1ère promo)

[Time Breaker in Space](#) de Raphaël (membre de la 1ère promo)

[Arkanoid \(arcade\)](#) La base

[Crystal Hammer \(Amiga\)](#) L'age d'or

[Bricks of Egypt](#) Génial casse brique thématique (égypte, existe aussi médiéval (Bricks of Camelot) et océan (Bricks of Atlantis))

[Un casse brique original sur TIC-80](#)

[Breakout Hero \(Pico-8\)](#)

[Wizorb](#)

Technologies :

C# (langage), Monogame (framework), Visual Studio 2022 (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 Orienté Objets (Le C#).
- Définir des 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.
- Mettre en œuvre une architecture Orientée Objet
- Découper le 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.
- Gérer d'une liste d'entités du jeu (par exemple, personnages, objets, ennemis, etc.), en utilisant des listes pour stocker et gérer ces données.

Ces objets pouvant être différents tout en utilisant une hiérarchie de classes afin de gérer un tronc de comportements en commun.

- Utiliser de l'injection de dépendance pour gérer les relations entre différentes parties du code du jeu et limiter la portée des variables (encapsulation).
- Utilisation du pattern de Services (Service Locator) afin de permettre à différentes entités du jeu d'accéder à des fonctionnalités distribuées (le pattern devra être codé from scratch et ne pas utiliser les services proposés par Monogame).

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

Semaine 1

Kickoff

Coaching

Semaine 2

Coaching

Coaching

Semaine 3

Coaching

Coaching

Semaine 4

Accompagnement

Accompagnement

Semaine 5

Accompagnement

Accompagnement

Semaine 6

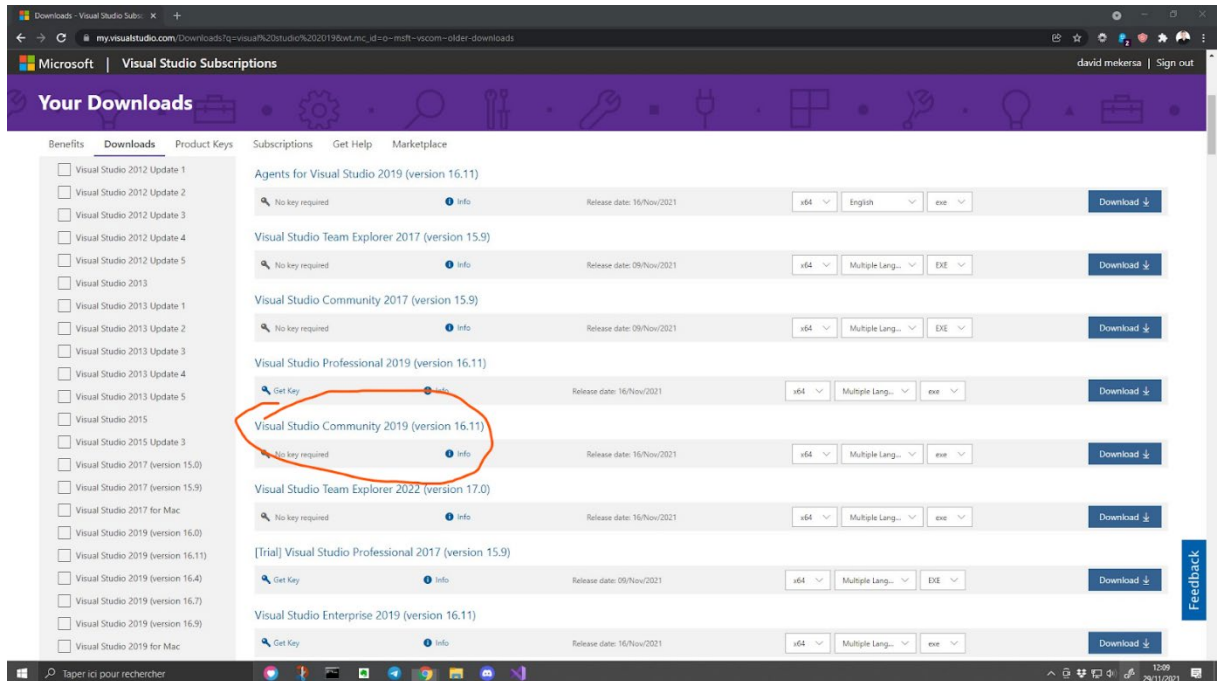
Soutenance sur RV

Ces dates peuvent être amenées à évoluer en fonction des contraintes des intervenants.

Prérequis

Vous devez installer :

- Visual Studio 2022 Community et Monogame



https://docs.monogame.net/articles/getting_started/0_getting_started.html

La procédure est décrite sur ma page :

<https://www.gamecodeur.fr/parcours-intermediaire-csharp/>

RESSOURCES

<https://www.gamecodeur.fr/parcours-intermediaire-csharp/>

En particulier les 7 premiers modules :

- Programmation de jeux en C# avec Monogame : Apprenez les bases de C# en 1 jour
- La Programmation Orientée Objet facile en C#
- Atelier : Concepts avancés et astuces en C#
- Atelier : Maitriser les images avec Monogame
- Atelier : Les bases d'un Lunar Lander
- Atelier : Sons et musiques en C# avec Monogame
- Atelier : Construire un jeu en C# avec Monogame

Un replay de stage qui vous forme de A à Z à la programmation d'un casse brique en POO en C# :

<https://gamecodeur.podia.com/stage-poo-sans-prise-de-tete?coupon=GAMINGCAMPUS>

Ou en Lua pur :

<https://gamecodeur.podia.com/stage-programmation-casse-brique-extreme?coupon=GAMINGCAMPUS>

Concernant les Design Patterns :

- [Le replay du DLC 87](#) – Programmer mieux c'est bien : Programmation d'un Service Locator
- <https://gameprogrammingpatterns.com/>
- <https://cdiese.fr/design-pattern-service-locator/>

Grille d'évaluation du projet

Critères d'évaluation
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
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.)
POO : Le code est découpé en objets et met en pratique les grands principes de la POO qui sont l'encapsulation, l'héritage et le polymorphisme.
Design Pattern de Services : Le jeu utilise un Service Locator
Présentations claire du projet : L'apprenant peut 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 maîtrise les concepts (fondamentaux : boucles, expressions, structures de contrôles, fonctions) et le langage utilisé dans son projet