

# LOGICIEL AVANCE

Type :	PROJET
Formations :	Ynov Informatique
Promotions :	Bachelor 2
UF :	Développement Logiciel et Base de données

---

## 1. CADRE DU PROJET

Ce projet permet l'évaluation des compétences acquises grâce aux modules de l'UF « Développement Logiciels » et « Base de données ». Pour ce faire, ce projet pourra être réalisé en groupe de 1 à 3 max.

Vous pouvez soumettre un projet personnel dont le contenu et les fonctionnalités devront respecter des conditions décrites dans la partie « Projet personnel ». Ce projet devra être validé par l'établissement.

Si vous n'avez pas d'idée de projet, vous avez le choix parmi une liste de projets proposés dans la partie « Projets au choix ».

Un bonus sera apporté aux projets personnels et aux groupes qui se challengent en proposant des fonctionnalités plus poussées.

Vous êtes totalement libre quant aux choix technologiques. Nous vous conseillons d'utiliser les langages de programmations et les outils vus avec vos intervenants mais acceptons toute technologie de Développement logiciel. Ex : C++, Java, C#, Python, NodeJS + Electron...

Seul l'aspect fonctionnel sera pris en compte et non l'aspect graphique.

Date de début : 22/03/2022

Date de rendu : 15/05/2022

## 2. OBJECTIFS DE FORMATION VISÉS

Vous serez évalués sur les compétences suivantes : *UF Développement Logiciel et base de données*

### POO

- maîtriser la programmation orienté objet
- mettre en place un algorithme complexe

### ARCHITECTURE LOGICIELLE

- effectuer des choix technologiques et respecter leurs conventions
- maîtriser la mise en place d'actions CRUD
- mettre en place une architecture de code

### UML

- traduire un besoin fonctionnel en modèle de donnée
- concevoir une architecture logiciel en amont du code

## 3. PREREQUIS

Voici les modules d'enseignement prérequis à ce projet :

- UML
- langage objet / architecture logicielle
- programmation bas niveau

## 4. LIVRABLES

- dépôt GIT de votre logiciel à jour :
  1. exécutable de vos logiciels
  2. un document de présentation de votre projet (rôles de chacun, technologies utilisées, structure algorithmique, fonctionnalités majeures, captures d'écran...)
  3. Tout le code de votre logiciel

## 5. MODALITÉS D'ÉVALUATION DU PROJET

Vous serez évalué sur l'ensemble des productions. L'évaluation prendra aussi la forme d'une présentation orale de synthèse d'environ 15 minutes accompagnée d'un support de présentation et d'une démonstration des fonctionnalités du site mises en place.

Le jury sera composé potentiellement d'une partie des intervenants des cours de l'UF « Développement Logiciels » ainsi que de mentors.

Un temps de questions-réponses d'une durée de 5 minutes sera prévu à l'issue des 15 minutes.

Des jalons auront également lieu au cours du déroulement du Projet pour bien suivre l'avancement de chacun des groupes.

## RÉCAPITULATIF DE LA GRILLE DE NOTATION

Le projet personnel décrit ci-dessous comporte 18 points de difficulté. A vous d'y ajouter des fonctionnalités pour atteindre les 28 points minimums requis. Vous pouvez vous référer aux projets proposés pour vous faire une idée des fonctionnalités et des points qui leurs sont attribuées. Les fonctionnalités devront être validées par l'établissement.

Un point bonus sera ajouté à la note finale si vous choisissez de réaliser un projet personnel.

Selon le nombre de points de difficulté total mis en place sur votre projet, des points bonus seront accordés selon les paliers suivants :

<i>Points de difficulté : Entre 28 et 35</i>	<i>0 point bonus</i>
<i>Points de difficulté : Entre 35 et 40</i>	<i>1 point bonus</i>
<i>Points de difficulté : Plus de 40</i>	<i>2 points bonus</i>

Les projets proposés devront être réalisés dans leurs intégralité pour obtenir la totalité des points.

## 6. DESCRIPTIF DU PROJET

Vous êtes totalement libre quant à l'apparence de vos interfaces. L'utilisation de librairies est autorisée et encouragée afin de gagner en rapidité de développement.

Vous avez la possibilité de choisir entre un projet personnel ou un projet proposé.

### LISTE DES PROJETS AU CHOIX :

#### PROJET PERSONNEL :

Le projet personnel devra atteindre ou dépasser les 28 points en degré de difficulté. Les fonctionnalités obligatoires représentent déjà 18 points en degré de difficulté, vous devez alors trouver des fonctionnalités valant au minimum 10 points. Vous pouvez vous référer à la liste de « Projet au choix » pour vous faire une idée des points alloués par degré de difficulté. **Le projet personnel devra être validé avant de pouvoir commencer.**

Le projet choisi devra avoir au minimum :

- soit : *Difficulté : 6*
  - une communication entre deux logiciels de votre création (deux fois le même ou deux différents, en local ou en réseau) : machine to machine
- un lien avec une base de données de votre création, comprenant au moins 3 tables  
*Difficulté : 3*
- au moins un algorithme avancé (ex : génération, IA, analyse, ...) *Difficulté : 3*
- de l'interaction avec l'utilisateur (ex : bouton, champ texte, événement clavier/souris...). *Difficulté : 2*
- un CRUD sur au moins une donnée utile au logiciel *Difficulté : 4*

*Degré de difficulté total : 18 points*

*(Bonus si le projet personnel est choisi)*

## 1<sup>er</sup> PROJET : SERVEUR DE MATCHMAKING

### Présentation

Ce projet contient 3 composantes : un serveur de Matchmaking, un logiciel client et une base de données.

### Fonctionnalités

- modèle de données :
  - une file d'attente, contenant pour chaque attendant : *Difficulté : 1*
    - le moyen de communiquer avec lui (IP et port par exemple)
    - un pseudo
    - la date à laquelle il est entrée dans la file
  - des matchs, contenant pour chacun : *Difficulté : 1*
    - le moyen de communiquer avec le joueur 1
    - le moyen de communiquer avec le joueur 2
    - le plateau de jeu
    - si le match est fini
    - s'il y a eu victoire du joueur 1, du joueur 2 ou égalité
  - des tours, contenant pour chacun : *Difficulté : 1*
    - la liaison avec le match
    - qui a joué : le joueur 1 ou le joueur 2
    - l'information du coup joué (en fonction du jeu choisi)
- le serveur de Matchmaking contient :
  - le lien avec la base de données *Difficulté : 3*
  - un système de socket avec les actions suivantes : *Difficulté : 5*
    - arrivé d'un client dans la file d'attente (réception)
    - début d'un match (envoi)
    - réception d'un tour (réception puis envoi)
    - fin d'un match (envoi)
  - une vérification constante de la file d'attente et création de matchs en fonction *Difficulté : 2*
  - une logique de jeu. Vous êtes libre quant au choix du jeu. Cela doit cependant être un jeu de plateau au tour par tour (Ex : puissance 4, dames, morpion, ...) *Difficulté : 5*
- le logiciel client contient :
  - un système de socket avec les actions suivantes : *Difficulté : 5*
    - entrée en file d'attente (envoi)
    - début d'un match (réception)
    - jouer un coup (envoi)
    - prendre en compte le coup adverse (réception)
    - fin d'un match (réception)
  - une partie de la logique du jeu choisis. *Difficulté : 2*

- soit : *Difficulté : 3*
  - une IHM pour pouvoir jouer
  - une CLI avec IA

*Degré de difficulté total : 28 points*

## 2<sup>ème</sup> PROJET : BORNE D'ARCADE

### Présentation

Ce projet consiste à créer une mini borne d'arcade contenant un jeu simple de 1 VS 1, à l'aide d'une Raspberry pi et de boutons/joystick à assembler.

### Fonctionnalités

- cinq écrans : *Difficulté : 3*
  - menu principal avec accès aux autres écrans et présentation des meilleurs scores
  - écran de jeu
  - écran de fin de partie
  - écran d'instructions avec possibilité de retour
  - écran d'option avec possibilité de retour
- l'écran d'instructions devra présenter à l'utilisateur toutes les données nécessaires à la bonne utilisation du logiciel *Difficulté : 1*
- modèle de données :
  - deux joueurs possédant : *Difficulté : 2*
    - une différenciation joueur 1 / joueur 2
    - une vitesse de rotation
    - une vitesse de déplacement
    - des points de vie
    - une puissance de tir
    - un délai de tir
    - une vitesse de projectile
  - des Scores possédants : *Difficulté : 1*
    - un pseudo
    - un score
- branchement des capteurs d'action et réception de leurs informations : *Difficulté : 5*
  - deux Joystick : un par joueur
  - quatre boutons poussoir : deux par joueur
- toute la navigation UI, le jeu ainsi que le remplissage des champs texte doivent pouvoir se faire par le biais des capteurs d'action *Difficulté : 5*

- déroulement d'une partie : *Difficulté : 5*
  - les deux joueurs apparaissent de part et d'autre de l'écran
  - l'espace de jeu se limite à l'écran, pas de physique, la vue caméra est dite "top-down"
  - le Joystick doit permettre au joueur de pivoter à 360° et d'avancer vers l'avant
  - un des boutons doit permettre au joueur de tirer un projectile
  - les données de la base doivent être utilisées pour gérer les capacités des joueurs
  - un Joueur perd des points de vie quand un projectile le touche
- fin d'une partie : *Difficulté : 3*
  - la partie se termine quand un joueur n'a plus de points de vie
  - le logiciel bascule alors sur l'écran de fin de partie, où le score du gagnant s'affiche ainsi que sa position dans les meilleurs scores
  - pour le score, vous devez créer une formule prenant en compte la différence de points de vie, le temps passé sur la partie ainsi que la différence sur les données de BDD du joueur 1 et 2 (un joueur peut gagner un plus gros score en baissant sa vitesse de rotation par rapport à son adversaire par exemple)
- écran d'option :
  - il devra permettre aux joueurs de modifier toutes leurs données (puissance de tir, vitesses ...) *Difficulté : 3*
  - ces données devront être stockées dans une base de données, afin de pouvoir les réutilisées même après fermeture du logiciel *Difficulté : 2*

*Degré de difficulté total : 30 points*

## 7. RESSOURCES COMPLEMENTAIRES

Exemple de boutons/joystick pour le projet n°2 :

[https://www.amazon.fr/Fosiya-Joueurs-Joystick-poussoirs-Raspberry/dp/B08HLZ4NJ9/ref=sr\\_1\\_5?keywords=joystick+arcade&qid=1647871779&sr=8-5](https://www.amazon.fr/Fosiya-Joueurs-Joystick-poussoirs-Raspberry/dp/B08HLZ4NJ9/ref=sr_1_5?keywords=joystick+arcade&qid=1647871779&sr=8-5)

A vous de trouver les ressources nécessaires en fonction des technologies choisies.

## 8. BESOINS MATERIELS ET LOGICIELS

- un IDE
- un langage de programmation orienté objet
- GIT
- un serveur (pour le projet n°1)
- un Raspberry et des boutons/joystick (pour le projet n°2)
- A noter que l'école ne fournit pas le matériel