



# 42

| **PERPIGNAN**  
Occitanie

PROJETS RÉALISÉS  
ET SOFTSKILLS ACQUISES  
**TRONC COMMUN**

# TECHNOLOGIES UTILISÉES



**Technologies utilisées pour les bonus  
des projets suivants : Born2beroot et Inception**

# LES PROJETS

**1**

**LIBFT**

Ce premier projet introduit les étudiants de 42 à la programmation C. Il vise à la fois à renforcer leurs acquis de la phase "piscine" en recodant des fonctions classiques de la bibliothèque standard, et à développer des fonctions utilitaires qui leur seront précieuses tout au long de leur cursus. En s'appropriant ces outils de programmation, les étudiants consolident leurs bases et acquièrent une méthodologie de travail efficace.

**2**

**GET\_NEXT\_LINE**

Que ce soit un fichier, stdin (entrée standard), ou même plus tard une connexion réseau, un développeur a toujours besoin d'un moyen de lire du contenu ligne par ligne. L'objectif de ce projet est de travailler sur un programme permettant cette lecture, programme qui sera utile aux étudiants tout au long de leur cursus.

**3**

**BORN2BEROOT**

Ce projet ambitieux vise à initier les étudiants au fascinant univers de la virtualisation et des machines virtuelles. En offrant une plongée immersive dans ce domaine en constante évolution, nous cherchons à élargir les horizons des étudiants en les familiarisant avec les concepts fondamentaux ainsi que les applications pratiques de la virtualisation.

**4**

**FT\_PRINTF**

Pour ce projet, les étudiants doivent recoder la fonction printf de la librairie standard. Ils auront dès lors la possibilité de la réutiliser dans leurs futurs projets. Ce projet porte principalement sur les arguments à taille variable.

# LES PROJETS

**5**

**PUSH-SWAP**

Ce projet demande de trier des données sur une pile, avec un set d'instructions limité, en moins de coups possibles. Pour le réussir, les étudiants doivent manipuler différents algorithmes de tri (par pivot, par insertion, par sélection) et choisir la solution la plus appropriée pour un classement optimisé des données.

**6**

**MINITALK**

L'objectif de ce projet est de coder un programme d'échange de données en utilisant des signaux UNIX. Il s'agit d'un projet d'introduction aux projets UNIX plus importants qui seront abordés plus loin dans le cursus.

**7**

**PIPEX**

Ce projet vise à approfondir la compréhension de deux concepts que les étudiants connaissent déjà à ce stade de leur progression : les redirections et les pipes. Il s'agit d'un projet d'introduction aux projets UNIX plus importants qui sont abordés plus tard dans le cursus.

**8**

**SO\_LONG**

Ce projet passionnant consiste en un jeu en 2D réalisé avec la librairie graphique minilibx. Les étudiants y explorent la manipulation des textures, des sprites et des tiles, enrichissant ainsi leurs compétences pratiques en développement de jeux.



# LES PROJETS

**9**

**FDF**

Tous les programmes que les étudiants ont développé jusqu'ici s'exécutaient uniquement en mode texte dans le terminal. Il s'agit ici de faire plus complexe : comment ouvrir une fenêtre graphique et dessiner à l'intérieur ? Ce projet a pour vocation d'introduire la programmation graphique, le projet FdF propose de représenter le maillage en fils de fer d'un terrain en 3D.

**10**

**FRACT-OL**

Cette initiative invite à explorer la programmation 2D et à plonger dans l'univers envoûtant des fractales grâce à un projet réalisé avec la librairie graphique minilibX. Les participants découvriront les subtilités de la création graphique, fusionnant ainsi l'art et la programmation dans une expérience captivante

**EXAM RANK 02**

Cet examen valide les compétences acquises lors des projets précédents.

**11**

**PHILOSOPHERS**

Ce projet permet aux étudiants de comprendre l'utilisation des "threads" et est une introduction au monde de la parallélisation. Il s'agit ici de comprendre les subtilités de la mémoire partagée et des zones d'exclusion mutuelles : "Parmi un groupe de philosophes, à vous d'anticiper de façon précise le moment où chaque philosophe du groupe devra prendre des fourchettes et manger des spaghettis, sans qu'aucun d'eux ne meurt de faim."

# LES PROJETS

## 12 MINISHELL

Ce premier projet de groupe a pour but de concevoir un clone simplifié du terminal, mettant en pratique les concepts explorés lors des projets précédents sur UNIX. Les participants seront initiés à la collaboration en équipe tout en consolidant leur compréhension des principes fondamentaux de la programmation système.

## EXAM RANK 03

Cet examen valide les compétences acquises lors des projets précédents.

## 13 NETPRACTICE

NetPractice offre une approche pratique pour explorer les réseaux, mettant l'accent sur la manipulation d'adresses IP et de composants virtuels. Les participants acquièrent ainsi une expérience concrète et précieuse dans ce domaine essentiel de l'informatique.

## 14 CUB3D

Ce projet est inspiré du jeu éponyme mondialement connu, considéré comme le premier FPS (Jeu de tir à la première personne) jamais développé. Il permet d'explorer la technique du ray-casting. L'objectif est de faire une vue dynamique au sein d'un labyrinthe, dans lequel le protagoniste doit trouver son chemin.

# LES PROJETS

## 15 MINIRT

Cette initiative de groupe sert d'introduction immersive au passionnant monde du Raytracing. Les participants explorent ensemble les fondements de cette technique de rendu, ouvrant ainsi la voie à des projets plus avancés dans le domaine de la modélisation et du graphisme 3D.

## 16 CPP MODULE 00

Ce premier module de C++ est fait pour comprendre les spécificités du C++ par rapport au C. Ceci est le premier contact avec la programmation orientée objet !

## 17 CPP MODULE 01

Ce module approfondi en C++ se concentre sur l'allocation de mémoire, les références, les pointeurs vers les membres et l'utilisation du switch. Les participants bénéficieront d'une immersion pratique dans ces concepts clés, renforçant ainsi leur expertise en programmation orientée objet.

## 18 CPP MODULE 02

Ce module approfondi en C++ explore le polymorphisme ad-hoc, les surcharges de fonctions et les classes canoniques orthodoxes. Les participants acquerront une compréhension approfondie de ces concepts fondamentaux, renforçant ainsi leurs compétences en programmation orientée objet.

# LES PROJETS

## 19 CPP MODULE 03

Cette mission individuelle vise à approfondir la compréhension de l'héritage en C++. Les participants exploreront les concepts et les applications de cette composante essentielle de la programmation orientée objet, consolidant ainsi leur expertise dans le langage.

## 20 CPP MODULE 04

Ce module approfondi en C++ explore le polymorphisme de sous-type, les classes abstraites et les interfaces. Les participants développeront ainsi une compréhension solide de ces concepts essentiels, renforçant leur expertise en programmation orientée objet.

## EXAM RANK 04

Cet examen valide les compétences acquises lors des projets précédents.

## 21 CPP MODULE 05

Ce module approfondi en C++ se concentre sur les fonctions Try/Catch et la gestion des exceptions. Les participants développeront une compréhension robuste de ces mécanismes fondamentaux, renforçant ainsi leur expertise en programmation sécurisée.



# LES PROJETS

## 22 CPP MODULE 06

Ce module spécialisé en C++ explore les divers types de conversions (cast) disponibles. Les participants acquerront une compréhension approfondie de ces techniques, renforçant ainsi leur maîtrise des opérations de conversion sécurisées et efficaces.

## 23 CPP MODULE 07

Ce module spécialisé en C++ se concentre sur la compréhension des templates. Les participants exploreront en profondeur cette fonctionnalité puissante du langage, renforçant ainsi leur capacité à créer des programmes flexibles et génériques.

## 24 CPP MODULE 08

Ce module spécialisé en C++ met l'accent sur la compréhension des conteneurs templates, des itérateurs et des algorithmes. Les participants exploreront en détail ces concepts fondamentaux, renforçant ainsi leur capacité à manipuler efficacement les structures de données et à utiliser les fonctionnalités avancées du langage.

## 25 CPP MODULE 09

Ce module se concentre sur la familiarisation avec les conteneurs en C++. Les participants exploreront une variété de conteneurs disponibles dans la bibliothèque standard du langage, renforçant ainsi leur compréhension des structures de données essentielles pour le développement logiciel.

# LES PROJETS

## 26 INCEPTION

Ce projet a pour but d'approfondir les connaissances acquises en virtualisation en utilisant Docker. Il s'agit de virtualiser plusieurs images Docker en les créant dans une nouvelle machine virtuelle personnelle.

## 27 WEBSERV

Ce projet demande aux étudiants de créer leur propre serveur HTTP. Ils doivent suivre la RFC d'HTTP et être capable de tester avec un vrai navigateur web. HTTP est un des protocoles les plus utilisés sur internet. Connaître son fonctionnement est fondamental même pour les étudiants qui ne feraient pas de web à la fin de leur cursus.

## 28 FT\_IRC

L'objectif de ce projet de groupe est de concevoir un serveur IRC personnalisé en suivant les normes RFC d'Internet. Les membres du groupe collaboreront pour explorer en profondeur les protocoles de communication réseau, tout en acquérant une expérience pratique précieuse dans le développement de logiciels.

## EXAM RANK 05

Cet examen valide les compétences acquises lors des projets précédents.

# LES PROJETS

## 29 FT\_TRANSCENDENCE

Ce projet engageant implique la création d'un site web pour participer à une compétition du célèbre jeu Pong ! C'est une étape cruciale pour conclure le tronc commun, intégrant la gestion de données, une interface utilisateur interactive, ainsi que l'utilisation de Docker et d'autres technologies essentielles. Les membres du groupe travailleront ensemble pour réaliser une expérience de jeu en ligne immersive et fonctionnelle.

## EXAM RANK 06

Cet examen valide les compétences acquises lors des projets précédents et permet, avec le projet Transcendance, de valider le tronc commun.

# LES SKILLS

## Programmation impérative

### Connaître et comprendre

Les bases de la programmation en C : la syntaxe C, les variables, les boucles, les branches conditionnelles, les fonctions, la récursivité, les instructions, les calculs et les expressions, les opérateurs de comparaison, les types standard et avancés, le traitement des chaînes de caractères, les structures, les inclusions et les bibliothèques, l'allocation et la libération de mémoire, les listes chaînées, les arbres, la bibliothèque standard C.

### Pouvoir réaliser

Créer plusieurs logiciels en C : un clone de printf de la LibC ; un clone léger du shell bash (impliquant la programmation système) ; un jeu à la première personne en temps réel utilisant le lancer de rayons (impliquant la graphique).

## Programmation orientée objet

### Connaître et comprendre

Les principes de la programmation orientée objet en C++ : les classes, les espaces de noms, les constructeurs et destructeurs, la gestion de la mémoire en C++, l'héritage, l'abstraction, la surcharge, les modèles, les types et outils standard de la bibliothèque C++.

### Pouvoir réaliser

Créer des logiciels en C++, jusqu'à un serveur web pleinement conforme à la RFC (impliquant la programmation système). Créer une plateforme web complète (un marché, un réseau social) en utilisant un framework web aléatoire et orienté objet comme Rails, Django, Symfony, Nodes, etc. (impliquant le Web).

## Programmation système : Unix

### Connaître et comprendre

Les interactions système Unix classiques : appels système, accès et gestion du système de fichiers, création, exécution et gestion des processus ; communications inter-processus : tubes et signaux ; gestion des périphériques et input/output control, capacités du terminal ; communication réseau : sockets TCP et UDP, résolution DNS, endianisme.

### Pouvoir réaliser

Créer des logiciels utilisant les ressources Unix : un client et serveur IRC pleinement conformes à la RFC, compatibles avec les normes Internet (impliquant la programmation orientée objet) ; un clone léger du shell bash (impliquant la programmation impérative) ; un sous-ensemble de la bibliothèque C standard codé en ASM Intel64.

# LES SKILLS

## Graphiques

### Connaître et comprendre

La gestion des images, la structure RGB d'une image, la manipulation des zones, le dessin dans une image, l'interaction avec le système de gestion de fenêtres et la réception des événements et des entrées utilisateur du clavier et de la souris, la programmation avec des rappels et une boucle d'événements.

### Pouvoir réaliser

Créer des logiciels graphiques qui ouvrent et gèrent une fenêtre graphique dédiée avec des interactions utilisateur : un programme de lancé de rayons qui calcule des images basées sur un fichier de description ; un jeu en première personne jouable.

## Réseau et administration système

### Connaître et comprendre

Les bases du réseau informatique : adresses IP, sous-réseaux, routage par défaut, structure du réseau local, connectivité hôte à hôte aux services réseau ; les bases de l'administration système : installation du système d'exploitation avec Linux, configuration de la sécurité, de l'accès, des utilisateurs, du stockage, installation de services réseau tels que le courrier, le DNS, le serveur web, etc.

### Pouvoir réaliser

Installer une distribution Linux, installer divers paquets, configurer l'accès graphique et distant, configurer des services réseau standard ; Utiliser plusieurs machines virtuelles, créer un réseau local de communication entre elles et avec Internet ; Créer et exécuter des conteneurs avec Docker. Créer une infrastructure Kubernetes qui exécute des conteneurs de services.

## Algorithmes et IA

### Connaître et comprendre

Les algorithmes standards sur des structures standards : recherche, tri, insertion, suppression, équilibrage, sur : tableaux, listes chaînées, arbres. Machine à états et gestion asynchrone.

### Pouvoir réaliser

Utiliser différents algorithmes pour manipuler diverses structures de données dans de nombreux logiciels en C et C++ : liste d'objets géométriques dans un programme de lancé de rayons, gestion de multiples connexions réseau simultanément dans un serveur IRC.



# LES SKILLS

## Rigueur

### Connaître et comprendre

La nécessité de respecter des contraintes administratives, techniques et de temps. La nécessité d'un processus de test large et approfondi pour éliminer les échecs.

### Pouvoir réaliser

Créer des logiciels en respectant un style de codage local et en respectant les obligations techniques : fonctions de bibliothèque C obligatoires ou interdites ; Pouvoir créer un logiciel évalué par un autre logiciel, nécessitant des réponses précises et un formatage strict en sortie ; Pouvoir créer un logiciel en effectuant de nombreux tests pour éviter tous les crashes possibles qui entraîneraient un échec global du projet.

## Programmation parallèle

### Connaître et comprendre

Programmes multi-threadés, threads POSIX, concurrence de threads, mutex.

### Pouvoir réaliser

Créer des logiciels threadés : lutte pour une ressource commune.

## Web

### Connaître et comprendre

L'architecture client-serveur impliquée sur le Web, rôle et actions du serveur web, rôle et actions du navigateur web ; Le protocole HTTP ; Les technologies web impliquées : HTML, CSS, Javascript, images et vidéos ; Langage et framework backend pour les sites web dynamiques : parmi php, ruby, python, go, javascript, Rails, Symfony, Django, Node, ... ; Modèle MVC ; Services web utilisateurs : sessions web, authentification, cookies, recherche, panier, configuration du backoffice, ... ; Bases de l'expérience utilisateur, de l'interface utilisateur et de la conception.

### Pouvoir réaliser

Créer un site web complet en utilisant un framework web classique aléatoire : marché, VOD/streaming, réseau social, wiki, ...

# LES SKILLS

## Données et BDD

### Connaître et comprendre

Concepts de stockage et de structure des données ; Installation et configuration d'une base de données open source (postgresql, mysql, ..) ; Manipulation de la base de données : tables, relations, langage de requête SQL ; Base de données en tant que backend de serveur web pour le stockage d'informations.

### Pouvoir réaliser

Utilisation de bases de données relationnelles en conjonction avec un framework web pour créer un site web complet : marché, VOD/streaming, réseau social, wiki...

## Adaptation

### Connaître et comprendre

Déployer une stratégie de recherche et de test face à un problème ou un contexte inconnu. Savoir que les technologies, marques, langages de programmation sont tous volatiles et peuvent être obsolètes après plusieurs années. Comprendre que faire face à des situations et des problèmes inconnus fait partie du travail après le cursus, nécessitant des compétences d'adaptation.

### Pouvoir réaliser

Tous les projets du cursus, chacun d'eux nécessite une partie d'adaptation car il n'y a pas de cours, pas de professeur, pas de MOOC, pas de recette, ni aucune information fournie pour résoudre le projet.

## Groupe et interpersonnel

### Connaître et comprendre

Collaboration, relations et situations de gestion de groupe, y compris différents types d'interactions entre les personnes (amicales, tendues...)

### Pouvoir réaliser

Créer un logiciel avec une équipe d'étudiants, nécessitant une coopération entre les membres. Se connecter avec de nouveaux pairs pour échanger des idées et débattre de tout projet afin de trouver de nouvelles hypothèses et de nouveaux moyens de les tester. Pouvoir prendre la responsabilité d'évaluer quelqu'un lors des évaluations par les pairs.

# LES SKILLS

## Organisation

### Connaître et comprendre

La régularité et l'implication conduisent à la productivité et à l'efficacité. Avantages d'une routine de travail et de respect d'un horaire. Efficacité supérieure du travail en journée par rapport au travail de nuit. Organiser des pauses hebdomadaires, des vacances et une vie saine.

### Pouvoir réaliser

Pouvoir achever le cursus en harmonie avec sa vie personnelle, sans être pris par le système de détection des "touristes".

## Sécurité

### Connaître et comprendre

Les meilleures pratiques de codage qui réduisent les violations de sécurité : vérification des erreurs, entrées aléatoires d'utilisateurs et de réseau, cas particuliers, éviter les fonctions longues et complexes à but multiple, tests fonctionnels.

### Pouvoir réaliser

Créer des logiciels tout au long du cursus avec un niveau de sécurité croissant et maîtrisé.

## Expérience en entreprise

### Connaître et comprendre

Collaboration, relations et situations de gestion de groupe, y compris différents types À travers diverses conférences, meet-up, hackathon, élargir son propre horizon sur les emplois, les domaines de l'économie qui demandent des talents numériques.

### Pouvoir réaliser

Développer une vision beaucoup plus claire du marché du travail et pouvoir commencer une recherche de stage productive et efficace.



contact@42perpignan.fr

