

Projet d'Informatique

Séance "Analyse"

Comprendre et préparer le développement du projet



UMONS

Sébastien Bonte

Département d'Informatique

Faculté des Sciences

Année académique 2022–2023

Par quoi commencer ?

Par quoi commencer ?

- Premier projet informatique conséquent ($\sim 120h$, plusieurs milliers de lignes de code).

Par quoi commencer ?

- Premier projet informatique conséquent ($\sim 120h$, plusieurs milliers de lignes de code).
- Notions inconnues pour le moments (introduites dans le cours d'Algorithmique et Programmation II).

Par quoi commencer ?

- Premier projet informatique conséquent ($\sim 120h$, plusieurs milliers de lignes de code).
- Notions inconnues pour le moments (introduites dans le cours d'Algorithmique et Programmation II).

Mais ce n'est pas grave

Vous pouvez déjà vous poser certaines questions afin de préparer et organiser le développement du projet.

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?
- Avez-vous identifié les utilisateurs du projet ? Quelles sont leurs attentes spécifiques ? Quels seront leurs principaux critères à satisfaire ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?
- Avez-vous identifié les utilisateurs du projet ? Quelles sont leurs attentes spécifiques ? Quels seront leurs principaux critères à satisfaire ?
- Pouvez-vous cataloguer les attentes du projet dans les catégories suivantes : base, partie logique, partie graphique, peaufinage ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?
- Avez-vous identifié les utilisateurs du projet ? Quelles sont leurs attentes spécifiques ? Quels seront leurs principaux critères à satisfaire ?
- Pouvez-vous cataloguer les attentes du projet dans les catégories suivantes : base, partie logique, partie graphique, peaufinage ?
- Êtes-vous sûrs de connaître en détails les règles ou les spécificités de chaque partie du projet ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?
- Avez-vous identifié les utilisateurs du projet ? Quelles sont leurs attentes spécifiques ? Quels seront leurs principaux critères à satisfaire ?
- Pouvez-vous cataloguer les attentes du projet dans les catégories suivantes : base, partie logique, partie graphique, peaufinage ?
- Êtes-vous sûrs de connaître en détails les règles ou les spécificités de chaque partie du projet ?
- Y-a-t'il des attentes ou des fonctionnalités qui sortent de l'ordinaire ?

Comprendre l'énoncé du projet

La première étape du projet consiste à bien le comprendre :

- Avez-vous déjà pris connaissance de l'énoncé, des objectifs ?
- Êtes-vous capables de synthétiser ce qui est attendu de vous ?
Pouvez-vous distinguer les objectifs principaux des objectifs secondaires ?
- Avez-vous identifié les utilisateurs du projet ? Quelles sont leurs attentes spécifiques ? Quels seront leurs principaux critères à satisfaire ?
- Pouvez-vous cataloguer les attentes du projet dans les catégories suivantes : base, partie logique, partie graphique, peaufinage ?
- Êtes-vous sûrs de connaître en détails les règles ou les spécificités de chaque partie du projet ?
- Y-a-t'il des attentes ou des fonctionnalités qui sortent de l'ordinaire ?
- En plus du développement, devez-vous fournir des documents supplémentaires tels que de la documentation, un rapport, une présentation, des modèles, ... ?

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

- Est-ce que le projet nécessite un langage de programmation *nouveau*, que je dois encore apprendre et mettre en place sur ma machine ?

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

- Est-ce que le projet nécessite un langage de programmation *nouveau*, que je dois encore apprendre et mettre en place sur ma machine ?
- Quelles sont les dépendances logicielles du projet ? De quoi vais-je avoir besoin pour écrire le code, le compiler, l'exécuter ?

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

- Est-ce que le projet nécessite un langage de programmation *nouveau*, que je dois encore apprendre et mettre en place sur ma machine ?
- Quelles sont les dépendances logicielles du projet ? De quoi vais-je avoir besoin pour écrire le code, le compiler, l'exécuter ?
- Est-ce que je dispose déjà de tout cela ? Si ce n'est pas le cas, suis-je capable de mettre tout cela en place dans un délai raisonnable ?

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

- Est-ce que le projet nécessite un langage de programmation *nouveau*, que je dois encore apprendre et mettre en place sur ma machine ?
- Quelles sont les dépendances logicielles du projet ? De quoi vais-je avoir besoin pour écrire le code, le compiler, l'exécuter ?
- Est-ce que je dispose déjà de tout cela ? Si ce n'est pas le cas, suis-je capable de mettre tout cela en place dans un délai raisonnable ?
- Quelles sont les matières nécessaires au projet pour lesquelles je rencontre des soucis, ou que je ne maîtrise pas ? Puis-je facilement trouver de l'aide pour ces matières ?

Préparer le travail

Identifiez les outils, contraintes, préparatifs, connaissances, etc., requis par le projet.

- Est-ce que le projet nécessite un langage de programmation *nouveau*, que je dois encore apprendre et mettre en place sur ma machine ?
- Quelles sont les dépendances logicielles du projet ? De quoi vais-je avoir besoin pour écrire le code, le compiler, l'exécuter ?
- Est-ce que je dispose déjà de tout cela ? Si ce n'est pas le cas, suis-je capable de mettre tout cela en place dans un délai raisonnable ?
- Quelles sont les matières nécessaires au projet pour lesquelles je rencontre des soucis, ou que je ne maîtrise pas ? Puis-je facilement trouver de l'aide pour ces matières ?
- De quoi ai-je besoin pour rédiger les documents annexes ? Est-ce que je dispose déjà des ressources nécessaires (logiciel, images, documentation) ?

Décomposer les problèmes

« Tant que le projet n'est pas terminé, on ne peut pas le tester. »

Décomposer les problèmes

~~« Tant que le projet n'est pas terminé, on ne peut pas le tester. »~~

N'attendez pas d'avoir terminé tous les composants pour les tester. Fixez vous de petits objectifs à réaliser petit à petit. Chaque objectif doit répondre à des attentes particulières.

Décomposer les problèmes

Implémentez des « mock-up », des interfaces de code sans les fonctionnalités, afin de pouvoir tester directement certains composants.

Décomposer les problèmes

Implémentez des « mock-up », des interfaces de code sans les fonctionnalités, afin de pouvoir tester directement certains composants.

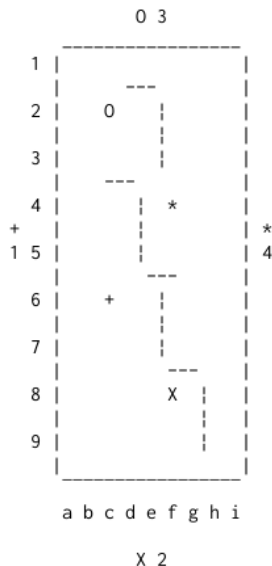
Exemple : une fonction de sauvegarde de partie dans l'interface graphique qui effectue un appel de méthode qui, en fait, ne fait rien (vous implémentez la sauvegarde plus tard, ou en parallèle avec votre binôme).

Décomposer les problèmes

Implémentez des « mock-up », des interfaces de code sans les fonctionnalités, afin de pouvoir tester directement certains composants.

Exemple : une fonction de sauvegarde de partie dans l'interface graphique qui effectue un appel de méthode qui, en fait, ne fait rien (vous implémentez la sauvegarde plus tard, ou en parallèle avec votre binôme).

Exemple : une interface graphique simplifiée ou console.



Player X: e8

Décomposition grâce à l'analyse top-down

Vue au cours de « Programmation et Algorithmique I ».

Décomposition grâce à l'analyse top-down

Vue au cours de « Programmation et Algorithmique I ».

On décompose le problème général en plusieurs sous-problèmes, que l'on décompose encore et encore, jusqu'à obtenir un ensemble de petits problèmes qu'on peut facilement aborder et résoudre dans un délai raisonnable.

Décomposition grâce à l'analyse top-down

Vue au cours de « Programmation et Algorithmique I ».

On décompose le problème général en plusieurs sous-problèmes, que l'on décompose encore et encore, jusqu'à obtenir un ensemble de petits problèmes qu'on peut facilement aborder et résoudre dans un délai raisonnable.

- Elle force la réflexion générale sur le problème, en obligeant les développeurs à envisager le problème dans son ensemble, et à considérer certains détails qui n'auraient pas été considérés si on se limitait à une vue générale du problème ;
- Elle impose d'une certaine manière de structurer ses idées et d'y voir plus clair, en mettant en évidence les liens ou les dépendances entre certaines parties du projet ;
- Elle permet de se concentrer par la suite sur la résolution de problèmes spécifiques qui, bien que liés au problème général, peuvent être résolus de manières isolées.

Analyse top-down

Les questions suivantes peuvent vous aider dans le cadre d'une telle approche :

Analyse top-down

Les questions suivantes peuvent vous aider dans le cadre d'une telle approche :

- À l'instar d'une recette de cuisine, êtes-vous capable de donner les grandes étapes de résolution du problème, sans penser en termes de *code*?

Analyse top-down

Les questions suivantes peuvent vous aider dans le cadre d'une telle approche :

- À l'instar d'une recette de cuisine, êtes-vous capable de donner les grandes étapes de résolution du problème, sans penser en termes de *code*?
- Quels sont les composants principaux de votre application, comment vont-ils interagir entre eux ? Pouvez-vous schématiser ces interactions et ces composants ?