

Correctement sauvegarder un projet de programmation

En informatique, les problèmes sont monnaie courante, surtout pour les débutants qui peuvent parfois naviguer sans savoir exactement ce qu'ils font. La perte de données peut être l'une des expériences les plus frustrantes pour un développeur. C'est pourquoi il est essentiel de bien comprendre comment sauvegarder correctement un projet de programmation, surtout lorsque vous travaillez sur des travaux universitaires importants d'une durée conséquente. Suivre quelques bonnes pratiques peut vous éviter bien des maux de tête. Dans ce document, nous allons explorer les étapes à suivre pour sauvegarder efficacement votre projet sur un stockage externe tel qu'une clé USB ou un disque dur, afin d'éviter tout accident malheureux. Cela garantit que même en cas de panne de matériel ou de défaillance du système, vous avez toujours une copie récente de votre travail.

Organisation du Projet

Avant de penser à la sauvegarde, assurez vous que votre projet est bien organisé. Utilisez une structure de répertoires claire et logique pour stocker vos fichiers source, ressources et documents (souvent cette structure vous sera déjà fournie dans un ZIP). Cela facilitera la sauvegarde et la restauration ultérieures.

Système de Contrôle de Version

L'utilisation d'un système de contrôle de version, tel que Git (accompagné de GitLab ou GitHub), est certainement bénéficielle. Ces outils permettent de garder une trace des modifications apportées à vos fichiers et facilitent la collaboration avec d'autres développeurs (votre binôme par exemple). Assurez vous de faire des commits réguliers et de pousser votre travail sur un serveur distant.

Cependant, Git n'est pas un outil magique, et de fausses manipulations catastrophiques arrivent très vite et ne tiennent qu'à une ou deux commandes. Il est donc important de comprendre que ce n'est pas parce que votre projet est stocké sur un serveur GIT distant qu'il est à l'abri des pépins. Cela ne vous dispense certainement pas de sauvegarder régulièrement votre projet sur un stockage externe.

Sauvegarde

Le système de sauvegarde présenté ci-dessous est un exemple de mise en place efficace contre une perte de travail:

- Munissez vous d'un stockage externe comme d'une clé USB ou d'un disque dur (32 GB devraient être suffisants pour la majorité des projets)
- Idéalement, n'utiliser ce moyen de stockage uniquement dans le cadre des sauvegardes (sinon créer un dossier **Backup**)
- Effectuer la procédure suivante de manière hebdomadaire:
 - Ouvrir votre navigateur de fichiers et aller à l'emplacement où se trouve le projet que vous souhaitez sauvegarder.
 - Copier (clique droit, copier) le dossier racine du projet (celui contenant tous les documents/fichiers nécessaires au bon fonctionnement du programme).
 - Insérer le stockage externe dans votre ordinateur.
 - Créer un nouveau dossier commençant par la date du jour et suivit du nom du projet. Par exemple: **2023_08_17-MP2_pacman**. Cette convention de nommage est très pratique car elle permet de visualiser les sauvegardes dans l'ordre où elles ont été créées et d'identifier immédiatement de quoi il s'agit.
 - Coller (clique droit, coller) maintenant le projet dans ce nouveau dossier
 - Éjecter le stockage externe (bien vérifier que la copie soit terminée avant)
 - Retirer le stockage externe de votre ordinateur et le ranger

Remarques

- **Sauvegarde avant des changements importants:** Avant d'apporter des modifications majeures à votre code, effectuez une sauvegarde. Cela peut vous éviter des problèmes si quelque chose ne se passe pas comme prévu, et vous pouvez toujours revenir à une version antérieure en cas de besoin.
- **Ne jamais utiliser directement une sauvegarde:** Ne travaillez jamais directement sur l'une de vos sauvegardes. Si vous avez besoin de récupérer l'une de vos sauvegardes, copier le contenu du dossier depuis le stockage externe vers votre ordinateur et déconnecter le stockage externe directement après. Cela vous évitera de "corrompre" vos sauvegardes afin qu'elles puissent être utilisées à nouveau.
- **Ne sous-estimer pas l'importance des sauvegardes:** Le processus de sauvegarde peut parfois sembler long et inutile. Détrompez-vous! La mise en place d'un processus de sauvegarde est tout à fait crucial dans tout projet informatique, et d'autant plus dans l'industrie et en entreprise plus tard. En effet, la plupart du temps ces sauvegardes ne sont pas utilisées (même si vous vous surprendrez sans doute à en avoir plus besoin que ce que vous pensiez), mais le jour où vous en aurez besoin, leur valeur sera inestimable.
- **Testez votre sauvegarde:** De temps en temps, assurez-vous de tester vos sauvegardes. Faites comme si vous n'aviez plus accès à votre projet pendant un instant, et essayez de correctement restaurer une copie depuis une sauvegarde sur votre stockage externe. Vérifiez alors que tout fonctionne correctement. Cela vous donnera l'assurance que vos sauvegardes sont prêtes à être utilisées en cas de besoin et lorsque la situation sera moins calme (par exemple proche d'une échéance).

English version

How to Properly Backup a Programming Project

In computer science, issues are commonplace, especially for beginners who may sometimes navigate without exactly knowing what they are doing. Data loss can be one of the most frustrating experiences for a developer. That's why it's essential to understand how to properly backup a programming project, especially when working on significant university assignments that span a considerable duration. Following some best practices can save you a lot of headaches. In this document, we will explore the steps to effectively backup your project to external storage such as a USB drive or a hard disk, to avoid any unfortunate accidents. This ensures that even in the event of hardware failure or system malfunction, you always have a recent copy of your work.

Project Organization

Before thinking about backup, make sure your project is well-organized. Use a clear and logical directory structure to store your source files, resources, and documents (often this structure will be provided in a ZIP file). This will make backup and future restoration easier.

Version Control System

Using a version control system, such as Git (accompanied by GitLab or GitHub), is certainly beneficial. These tools keep track of changes made to your files and facilitate collaboration with other developers (like your partner). Make sure to make regular commits and push your work to a remote server.

However, Git is not a magic tool, and catastrophic mistakes can happen quickly with just one or two commands. It is essential to understand that just because your project is stored on a remote GIT server doesn't mean it's immune to issues. This certainly does not exempt you from regularly backing up your project to external storage.

Backup

The backup system outlined below is an example of an effective setup against work loss:

- Get an external storage such as a USB drive or a hard disk (32 GB should be sufficient for most projects).
- Ideally, use this storage only for backups (otherwise, create a 'Backup' folder).
- Perform the following procedure weekly:
 - Open your file browser and go to the location of the project you want to back up.

- Copy (right-click, copy) the project’s root folder (the one containing all the documents/files necessary for the program to function correctly).
- Insert the external storage into your computer.
- Create a new folder starting with today’s date followed by the project’s name. For example: 2023_08_17-MP2_pacman. This naming convention is handy as it allows you to view backups in the order they were created and immediately identify what it is.
- Paste (right-click, paste) the project into this new folder.
- Eject the external storage (make sure the copy is complete before).
- Remove the external storage from your computer and store it.

Notes

- **Backup before significant changes:** Before making major changes to your code, perform a backup. This can save you trouble if something doesn’t go as planned, and you can always revert to an earlier version if needed.
- **Never work directly on a backup:** Never work directly on one of your backups. If you need to retrieve one of your backups, copy the content of the folder from external storage to your computer and disconnect the external storage directly afterward. This will prevent “corrupting” your backups so they can be used again.
- **Do not underestimate the importance of backups:** The backup process may sometimes seem long and unnecessary. Don’t be fooled! Establishing a backup process is crucial in any IT project, especially in industry and later in the workplace. Most of the time, these backups are not used (although you may find yourself needing them more than you thought), but when you need them, their value will be invaluable.
- **Test your backup:** From time to time, make sure to test your backups. Pretend as if you have lost access to your project for a moment and try to properly restore a copy from a backup on your external storage. Verify that everything works correctly. This will give you confidence that your backups are ready to be used when needed, especially in a less calm situation (such as approaching a deadline).