



Ce projet consiste en la création d'un logiciel pour visualiser les astéroïdes proches de la terre grâce à l'api de la Nasa.

Le sujet sera réparti en 3 parties :

- Respect de la POO.
- La récupération des données api de la Nasa.
- La création du logiciel et mise en place du visuel.

Chacune de ces parties ont des consignes et contraintes vis à vis de leur gain de point.

Points de chaque partie :

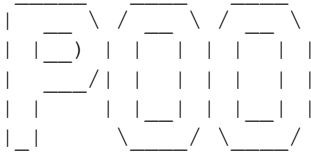
- Partie 1 _POO : 6 points
- Détail :
 - Respect des règles : 5 points
 - Projet qui fonctionne (sans erreur) : 1 points
- Partie 2 _API : 10 points
- Détail :
 - Récupération de l'image du jour de la Nasa : 2 points
 - Récupération de liste des astéroïdes : 2 points
 - Récupération du détail d'un astéroïde : 2 points
 - Maîtrise de l'async : 4 points
- Partie 3 _VISUEL : 4 points
- Détail :
 - Ergonomie du logiciel : 2 points
 - Visuel du logiciel : 2 points
- Malus :
 - Détail :
 - Noms des variables pas claires : -1.5 points
 - Non-respect des demandes de rendu : -2 points
 - Code non commenté : -1.5 points
- Bonus :
 - Détail :
 - Ajout d'une nouvelle donnée de l'api dans votre visuel : 1 points
(Cumulable)
(Doit être précisé dans un README)
(Ne peut être obtenue que si le contenu de base est déjà réalisé)

Le rendu devra être fait sur gitea, dans un nouveau répertoire. Le namespace et nom du répertoire de rendu devront être votre "%pseudo_ytrack%_nasa" (exemple : pchesneau_nasa).

Date du rendu : 5 février 23h59

Si un exercice rendu non commenté, réussi ou non, comporte des ressemblances que nous considérons comme flagrantes avec des ressources en ligne, ou avec des exercices rendu par d'autres étudiants, alors il sera considéré comme de la triche.

Toute triche résultera en a 0 pour l'étudiant. Non négociable.



- Règles :
 - Tout attribut doit être privée.
 - Tout attribut doit avoir un getteur s'il a vocation à être récupéré dans une autre
- Classe :
 - Tout attribut doit avoir un setteur.
 - Toutes les classes non-abstract doivent avoir un constructeur au complet.
 - Toutes méthodes doivent être écrites en CamelCase.
 - Toutes méthodes doivent être privée par défaut, et public si elle a vocation à être.
 - Récupéré dans une autre classe.
 - Toutes les classes doivent avoir un objectif précisé en commentaire.
 - Ces règles sont valables pour tout le projet.

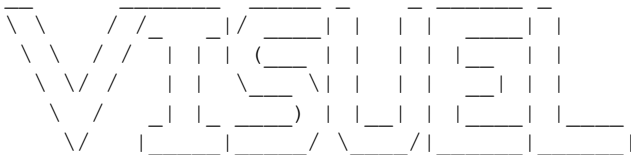


Lien utile pour récupérer les données API : <https://learn.microsoft.com/fr-fr/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

Voici les liens de l'API :

- Pour récupérer la votre clé : <https://api.nasa.gov>
- Exemple de lien :
https://api.nasa.gov/neo/rest/v1/feed?start_date=START_DATE&end_date=END_DATE&api_key=API_KEY

Vous devrez récupérer toute l'info nécessaire à la partie VISUEL.



- Lien d'un tuto complet pour faire des app Windows fonctionnelle : https://www.youtube.com/watch?v=wfwXdh-_k_4
- Lien d'un tuto complet pour faire des app Windows jolie : <https://www.youtube.com/watch?v=PzP8mw7JUzI>

Paul est votre client pour ce projet, il est à l'écoute de vos propositions, et est conscient que tout n'est pas dit ici, donc n'ayez pas peur d'avoir des initiatives, et de demander au client différentes fonctionnalités.

- Fonctionnalité :
 - Avoir le nom de votre application d'afficher
 - Le fond d'écran de votre logiciel doit être l'image de APOD
 - Pouvoir cliquer sur un bouton pour avoir les informations de l'image de APOD
 - Pouvoir voir la liste des astéroïdes proche de la terre
 - Nom
 - Distance min
 - Pouvoir sélectionner un astéroïde pour voir sa liste des approches passé, et future

Vous avez champ libre pour le visuel.

Tout doit être mis en forme de façon ergonomique