AdonneCosmos

1. Titre du projet

Cosmos Language Server

2. Description

Créer un addon pour vscode qui supporte la syntaxe du langage cosmos.

3. Table des matières

4. Requirements

- Node
- Visual Studio Code
- Le language cosmos

5. Installation

- Si vous n'avez pas Visual Studio Code, installez-le à cette adresse
- `git clone https://github.com/quemet/AdonneCosmos.git`

6. Utilisation

- `cd AdonneCosmos`
- Lire le README.md

7. Journal de travail

Pour voir le journal de travail, veuillez vous rendre dans le dossier racine de ce projet et ouvrez le fichier nommé **JNLTRAV-QueMetroz.xlsm**, puis allez dans l'onglet **Journal de Travail**

8. Technologie Utilisé

MARKDOWN

{json}

VISUAL STUDIO CODE

GIT

GITHUB

NODE.JS

NPM

9. Taches qui reste à faire

- Adapter l'extension pour les erreurs de syntaxes et les warnings
- Test de l'extension avec Tests unitaire
- Publier l'extension

10. Commentaires sur les 3 fichiers JSON

10.1. package.json

```
{
    "name": "cosmos-extension", // Define the name of the extension
    "displayName": "Cosmos Extension", // Define the name of the extension when it's displayed by VS Code
    "description": "Supported Language Cosmos", // Define the description of the extension
    "version": "0.0.1", // Define the version of the extension
    // Define the engine for the extension
    "engines": {
        "vscode": "^1.93.0"
    },
    // Define the category of the extension
    "categories": [
        "Programming Languages"
    ],
    // Define the fonctionality of the extension
    "contributes": {
        // Define the configuration of the language
        "languages": [{
            "id": "cosmos", // Define the id of the language
            "aliases": ["Cosmos", "cosmos"], // Define the aliases of the
language
            "extensions": [".cosmos"], // Define the extension of the language
            "configuration": "./language-configuration.json" // Define the file
of the configuration of the language
        }],
        // Define the configuration of the extension (Syntax Highlighting)
        "configurationDefaults": {
            // Define the color of the token
            "editor.tokenColorCustomizations": {
                // Defne the rule of the extension
                "textMateRules": [
                    {
                        "scope": "keyword.control.cosmos", // Define keyword
like "Afficher" / "Effacer l'écran"
                        // Défine the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "comment.line.double-slash.cosmos", // Define
keyword for the single block commentary
                        // Défine the color and the style of the keyword
                        "settings": {
                            "foreground": "#6A9955" // Define a dark green
color like this https://singlecolorimage.com/get/6A9955/500x500
                        }
                    },
                    {
                        "scope": "comment.block.cosmos", // Define the keyword
for the multi block commentary
                        // Défine the color and the style of the keyword
                        "settings": {
                            "foreground": "#6A9955" // Define a dark green
color like this https://singlecolorimage.com/get/6A9955/500x500
                        }
                    },
                    {
                        "scope": "keyword.operator.arithmetic.cosmos", //
Define the symbol of arithmetic operation like "+" / "-"
                        // Défine the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.operator.arithmetic.fullname.cosmos",
// Define the full word of arithmetic like "plus" / "moins"
                        // Défine the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "string.cosmos", // Define the start and the
end of a string like '"'
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#CE9178" // Define a skin color like
this https://singlecolorimage.com/get/CE9178/500x500
                        }
                    },
                    {
                        "scope": "variable.cosmos", // Define the start of a
variable like "#abc"
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#9CDCFE" // Define a light blue
color like this https://singlecolorimage.com/get/9CDCFE/500x500
                        }
                    },
                    {
                        "scope": "keyword.operator.logical.cosmos", // Define
the fullword for condition operation
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.operator.arithmetic", // Define the
symbol for condition operation
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.operator", // Define alternative for
if else if and else condition
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
```

```
                        }
                    },
                    {
                        "scope": "keyword.operator.questionmark", // Define the
end of a condition before the return of the line
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.loop", // Define the keyword for the
loop of the language
                        // Define the color and the style of the keyord
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.control.other", // Define function
additional like "Découper"
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.color.cosmos", // Define all the
color of cosmos
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "variable.input.cosmos", // Define all the
keyword related of the input of the user
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "variable.name.cosmos", // Define the keyword
to store value into a variable
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
```

```
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "variable.random.cosmos", // Define the
keyword to store random value into a variable
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#569CD6" // Define a light blue
color like this https://singlecolorimage.com/get/569CD6/500x500
                        }
                    },
                    {
                        "scope": "keyword.slash.comos", // Define the special \
function like \t and \n
                        // Define the color and the style of the keyword
                        "settings": {
                            "foreground": "#FFFF00" // Define a yellow color
like this https://singlecolorimage.com/get/FFFF00/500x500
                        }
                    }
                ]
            }
        },
        // Define the grammar of the language
        "grammars": [{
            "language": "cosmos", // Define the name of the language
            "scopeName": "main.cosmos", // Define the name of the scope of the
main
            "path": "./syntaxes/cosmos.tmLanguage.json" // Define a file to all
the grammar rule
        }]
    }
}
```

10.2. language-configuration.json

```
{
    // Define a pattern to find the word
    "wordPattern": "[a-zA-Z_][a-zA-Z0-9_]",
    // Define the syntax to find the comments
    "comments": {
        "lineComment": "//", // Define the syntax for single block commentary
        "blockComment": ["/*", "*/"] // Define the syntax for multiple block
commentary
    },
    // Define the syntax of the brackets
    "brackets": [
        ["{", "}"],
        ["[", "]"],
        ["(", ")"]
    ],
    // Define the syntax for closing automaticly the following bracket
    "autoClosingPairs": [
        ["{", "}"],
        ["[", "]"],
        ["(", ")"],
        ["\"", "\""],
        ["'", "'"]
    ],
    // Define the syntax when we have to put a string in an array for example
    "surroundingPairs": [
        ["{", "}"],
        ["[", "]"],
        ["(", ")"],
        ["\"", "\""],
        ["'", "'"]
    ],
    // Define the colorization of the bracket in mode auto
    "colorizedBracketPairs": [
        [
            "(",
            ")"
        ],
        [
            "[",
            "]"
        ],
        [
            "{",
            "}"
        ]
    ],
}
```

10.3. cosmos.tmLanguage.json

```
{
    "scopeName": "*.cosmos", // Define all the file who is touched by this
extension
    "name": "Cosmos", // Define the name of the theme
    // Define all the patern to colorize the good keyword
    "patterns": [
        {
            "name": "keyword.operator.arithmetic", // Define the name who
coordinates in the file package.json
            "match": " <= | < | > | >= | = | == | est égal à | n'est pas égale
à | est supérieure à | est inférieure à | est supérieure ou égale à | est plus
petit ou égale à différente de | est plus grand que | est plus petit que | est
plus grande ou égale à | est plus petite ou égale à " // Define the keyword to
```

```
be selected to be colorize
        },
        {
            "name": "keyword.control.cosmos", // Define the name who coordinates in the file package.json
            "match": "\\b(Afficher|Effacer l'écran|Placer le curseur à la ligne |Placer le curseur à la colonne |Choisir la couleur|pour le texte|pour le fond)\\b" // Define the keyword to be selected to be colorize
        },
        {
            "name": "comment.line.double-slash.cosmos", // Define the name who coordinates in the file package.json
            "match": "//.*" // Define the keyword to be selected to be colorize
        },
        {
            "name": "comment.block.cosmos", // Define the name who coordinates in the file package.json
            "begin": "/\\*", // Define the begin of the keyword to be selected to be colorize
            "end": "\\*/" // Define the end of the keyword to be selected to be colorize
        },
        {
            "name": "variable.cosmos", // Define the name who coordinates in the file package.json
            "match": "#[\\w]*" // Define the keyword to be selected to be colorize
        },
        {
            "name": "keyword.operator.arithmetic.cosmos", // Define the name who coordinates in the file package.json
            "match": "[+\\-*/%^]" // Define the keyword to be selected to be colorize
        },
        {
            "name": "keyword.operator.arithmetic.fullword.cosmos", // Define the name who coordinates in the file package.json
            "match": " plus | moins | fois | divisé par | élevé à la puissance | puissance |racine carrée de |le reste de la division entière de | par | modulo " // Define the keyword to be selected to be colorize
        },
        {
            "name": "string.cosmos", // Define the name who coordinates in the file package.json
            "begin": "\"", // Define the begin of the keyword to be selected to be colorize
            "end": "\"", // Define the end of the keyword to be selected to be colorize
            // Define pattern into the string to be colorized like variable or special backslash command
            "patterns": [
                {
                    "name": "variable.cosmos", // Define the name who
coordinates in the file package.json
                    "match": "#[\\w]*" // Define the keyword to be selected to
be colorize
                },
                {
                    "name": "keyword.slash.comos", // Define the name who
coordinates in the file package.json
                    "match": "\\\\[nt]" // Define the keyword to be selected to
be colorize
                }
            ]
        },
        {
            "name": "keyword.operator.logical.cosmos", // Define the name who
coordinates in the file package.json
            "match": " ou au contraire | et |ou |et au contraire|l'inverse
de|not|!" // Define the keyword to be selected to be colorize
        },
        {
            "name": "keyword.operator", // Define the name who coordinates in
the file package.json
            "match": "Si | alors|sinon si |sinon |et sinon" // Define the
keyword to be selected to be colorize
        },
        {
            "name": "keyword.operator.questionmark", // Define the name who
coordinates in the file package.json
            "match": "\\?" // Define the keyword to be selected to be colorize
        },
        {
            "name": "keyword.loop", // Define the name who coordinates in the
file package.json
            "match": "Répéter autant de fois qu'il y a de |Répéter le nombre de
fois correspond à |Répéter tant que |Répéter \\d+x" // Define the keyword to be
selected to be colorize
        },
        {
            "name": "variable.input.cosmos", // Define the name who coordinates
in the file package.json
            "match": "Récupérer la saisie et la stocker dans la zone mémoire
nommée |Récupérer la saisie et la stocker dans |Attendre la prochaine touche et
la stocker dans " // Define the keyword to be selected to be colorize
        },
        {
            "name": "variable.name.cosmos", // Define the name who coordinates
in the file package.json
            "match": "Copier le texte |Créer une zone mémoire |Insérer la
valeur | dans la zone mémoire nommée |Allouer une zone de mémoire nommée| dans
la zone mémoire |dans |Allouer " // Define the keyword to be selected to be
colorize
        },
        {
            "name": "keyword.control.other", // Define the name who coordinates
```

```
in the file package.json
            "match": "Découper | sur |Attendre | ms" // Define the keyword to be selected to be colorize
        },
        {
            "name": "keyword.color.cosmos", // Define the name who coordinates in the file package.json
            "match": " rouge | vert | bleu | blanc | noir | gris |foncé" // Define the keyword to be selected to be colorize
        },
        {
            "name": "variable.random.cosmos", // Define the name who coordinates in the file package.json
            "match": "Placer un nombre aléatoire compris entre " // Define the keyword to be selected to be colorize
        }
    ]
}
```

11. Auteur(s)

Auteur : Quentin Métroz

12. Licence

Ce projet est sous licence MIT.