



Proposition de sujets
Projet long
UE Technologie objet

Alexandre Trotel, Ayoub Bouchama, Clement Cognard,
Ewen Le Bihan, Florent Puy, Gauthier Rancoule, Raphael Giudice

12 Février 2023

Contents

1	Tabless	2
2	Reg7	3
3	Moteur de rendu	4
4	Interface graphique pour <i>ortfo</i>	5

1 Tabless

Un nouveau type de navigateur web sans onglets.

L'idée est d'arrêter d'avoir beaucoup d'onglets ouverts et de devoir les fermer au bout d'une demi-heure d'inutilité, ou de crouler sous les plus de cents onglets ouverts.

À la place, les concepts d'historique, d'onglets et de pages à voir plus tard sont fusionnées en une pile appelée *The Stack*.

Ainsi, les onglets deviennent les sites ouverts actuellement (partie *Actuellement*), qui passent automatiquement dans la partie historique au bout d'un certain temps, et les sites à visiter plus tard passent naturellement dans la partie *Actuellement* à leur ouverture.

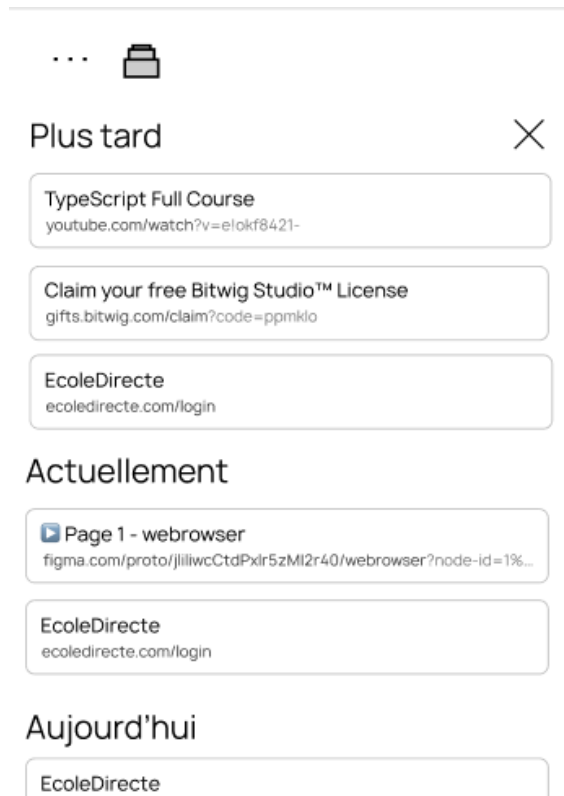


Figure 1: Concept de *The Stack*

Le logiciel devra proposer les fonctionnalités suivantes

- Navigation vers un site depuis la barre d'adresse-recherche avec suggestions
- Mode "immersif" où les quelques éléments d'interface prennent le moins de place possible pour laisser le site prendre le plus de place possible de la fenêtre
- Gestion de préférences de l'utilisateur (moteur de recherche utilisé, thème clair/sombre)
- Gestion intelligente des téléchargements: emplacements de sauvegarde différents selon le type de fichier, le domaine et le titre de l'onglet
- *The Stack* (fonctionnalité expliquée précédemment)
- Support d'extensions Chrome

2 Reg7

Un logiciel d'analyse et de construction d'expressions rationnelles depuis une interface intuitive.

Le but serait de permettre l'écriture d'expressions rationnelles complexes de manière simple et visuelle.

La partie analyse existe déjà sous forme de site web, le site <https://regexr.com>:

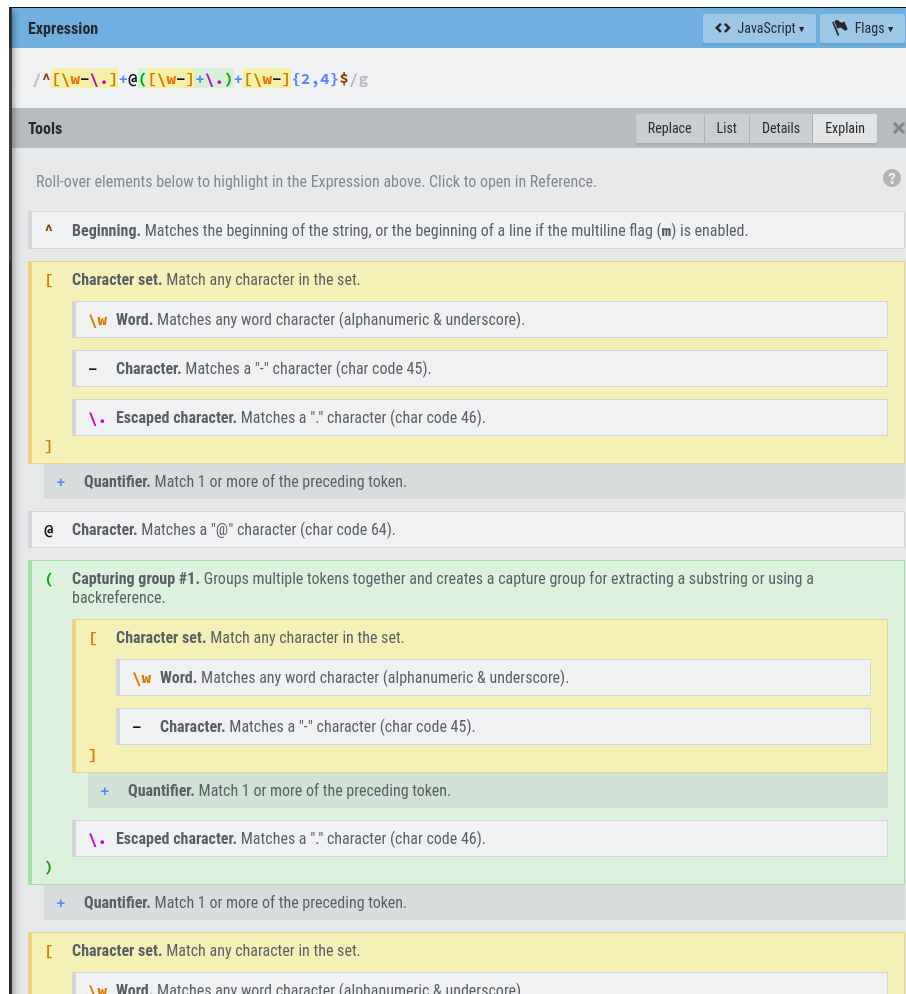


Figure 2: Interface du site regex

Cette logique en blocs imbriqués (à la *Scratch*) pourra être utilisée pour composer des expressions rationnelles, en plus de l'analyse d'expressions existantes.

Le logiciel proposera:

- L'importation d'expressions régulières
- Un pannel de différents blocs (un bloc "set de caractères", un bloc "groupe de capture", etc).
- L'exportation en différentes "saveurs" d'expressions régulières (Perl, Vim, etc.)
- Un champ de texte permettant de tester son expression sur des valeurs concrètes

3 Moteur de rendu

Le but est de créer un moteur de rendu qui prend une carte 2D en entrée, génère la représentation 3D du monde et permet à l'utilisateur d'explorer avec différentes perspectives et angles.

Le moteur de rendu 3D aura les caractéristiques suivantes :

Visualisation 3D Les utilisateurs peuvent naviguer dans le monde 3D.

Éclairage Le monde 3D pourra être éclairé à l'aide de sources de lumières.

Ombres Des ombres seront projetées par les objets et les murs.

Réflexions Les surfaces réfléchissantes seront incluses .

Textures Des textures pourront être appliquées

Paramètres personnalisables L'utilisateur aura la possibilité de régler divers paramètres tels que l'éclairage, les ombres, les reflets et les textures.

Les utilisateurs peuvent utiliser le moteur de rendu de deux manières différentes

En utilisant une image Les utilisateurs peuvent utiliser une image qui représente la carte 2D dans le logiciel. Cette image sera traitée par le logiciel, qui extraira les informations sur les murs, les objets et les autres éléments de la carte depuis l'image en lisant les pixels.

Saisie directe depuis le logiciel Les utilisateurs peuvent également saisir la carte directement dans le logiciel en créant la carte en 2D à l'aide d'outils de dessin pour les murs, les objets et d'autres éléments.

Le rendu sera principalement fait à partir de raytracing qui permettra de calculer l'affichage des murs et des objets sur le point de vu de l'utilisateur.

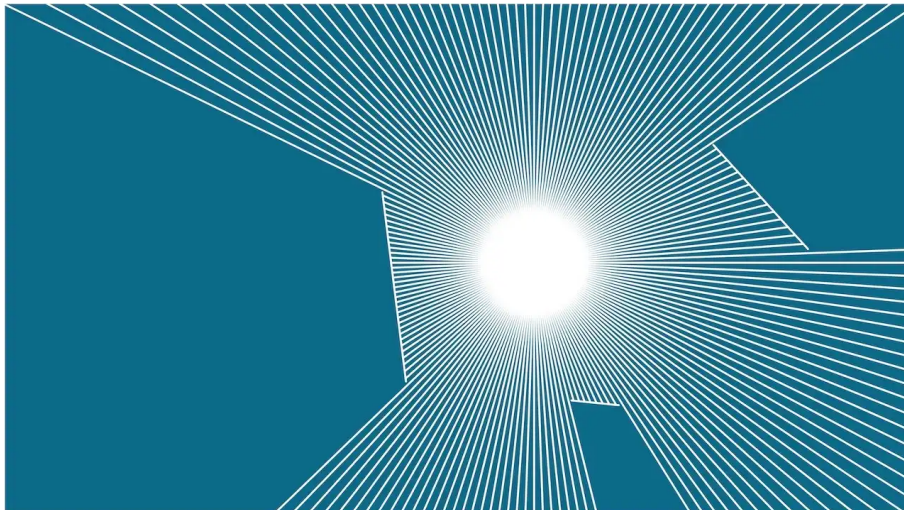


Figure 3: Raycaster

Le résultat final de ce projet sera un moteur de rendu 3D qui peut générer des mondes 3D à partir d'une carte 2D en utilisant du raycasting. Le moteur de rendu offrira une expérience réaliste et immersive aux utilisateurs.

4 Interface graphique pour *ortfo*

ortfo (<https://github.com/ortfo>) est un système de génération de sites portfolio à partir de fichiers de descriptions stockés à même les dossiers des projets figurant dans ce portfolio, ce qui simplifie grandement l'organisation pour l'utilisateur.

Cependant, écrire ce fichier de description demande une certaine aisance technique: c'est un fichier de texte brut écrit en markdown, avec une syntaxe particulièrement complexe pour l'utilisateur lambda pour décrire l'agencement des différents blocs de contenu (par exemple, avoir un paragraphe et une image côte-à-côte):

```
---
tag: [cli, program, app, gui, web]
made with: [go, svelte, json, yaml, markdown]
layout: [[intro], [why], [description], [description_example, database, database_example]]
started: 2020-06-26
---

# ortfo

:: fr

@intro
Un système de génération de portfolios faits maison.

@why
Ayant un grand nombres de projets (plus de 200), il me fallait une solution permettant de facilement
les ajouter à mon portfolio. Je voulais un système qui me permette d'avoir l'entrée dans mon portfolio
d'un projet stockée avec le projet lui même, dans le même dossier, par soucis d'organisation. Je voulais
également un site totalement généré, car il en résulterait un site très rapide et plus facile à maintenir.

@description
## ortfo/db
J'ai eu l'idée de stocker l'entrée portfolio d'un projet dans un sous-dossier .portfoliodb à l'intérieur
du dossier du projet, d'y décrire chaque projet avec du [markdown](/using/markdown).
Ensuite, il y a 3 types de blocs: des paragraphes, des liens isolés (utiles pour mettre en valeur un lien
vers le site réalisé ou le code source, par exemple) et des médias (vidéos, images, PDFs, etc.).
On peut également, avec une extension syntaxique sur mesure, traduire les articles en plusieurs langues

@description_example


@database
Après, ces entrées sont collectées et mises en commun dans un fichier [JSON](/using/json) constituant la
base de donnée du portfolio: c'est la partie 'db' de ortfo

@database_example

```

L'application permettrait à l'utilisateur de gérer la création automatique de ces fichiers à partir d'une interface graphique basée sur des blocs, de gérer:

- L'édition des métadonnées du projet (date de création, tags, technologies utilisées, etc.)
- L'écriture du contenu de l'entrée dans des blocs de contenu
- La traduction du contenu dans plusieurs langues
- L'ajout de médias (images, vidéos, etc.)
- L'agencement en colonnes des blocs en glissant et déposant (à la *Elementor*, *Wix*, etc.)
- L'exportation et publication du site en un clic