

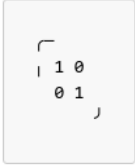
Types de données

Uiua ne possède que quelques types de données: les nombres, les caractères, les listes et les "box".

Les nombres peuvent être des entiers, des flottants et des nombres complexes. Les caractères sont des caractères unicode.

Les listes sont des listes d'éléments de même type. Et les "box" permettent de stocker des éléments de différents types dans une même liste.

Une grande partie des fonctions sont dites "pervasives" afin de pouvoir manipuler une liste de box sans avoir besoin de les débiller.

Code	Output
1	1
1.1	1.1
\mathbb{C} 3 4	4+3i
[1 2 3]	[1 2 3]
[1_0 0_1]	
@a	@a
"abc"	"abc"
□1	□1
□[1 0]	□[0 1]

Représentation des types de données

Les chaînes de caractères sont représentées par des listes de caractères mais du sucre syntaxique est disponible pour les déclarer. Une liste de box peut aussi être déclarée de manière plus lisible.

```
[@H @e @l @l @o] "Hello"
[□1 □2 □3] {1 2 3}
```

Sucre syntaxique pour les chaînes de caractères et les listes de box

Fonctions

Les fonctions sont définies par des symboles ou des noms. Il n'a pas de paramètres nommés: ils sont directement extraits du stack et les valeur de retour sont placées sur le stack. La plupart du temps la signature est inférée par le compilateur mais il est parfois nécessaire de spécifier le nombre de paramètres attendus et retournés (par exemple pour les fonctions récursives).

Fonctions système

Uiua possède un certain nombre de fonctions système qui permettent d'accéder à des fonctionnalités de bas niveau (système de fichiers, sockets, appel de commandes) ainsi que des fonctions pour différents types de médias (images, sons et gifs).

Foreign Function Interface

Il est aussi possible d'appeler des fonctions définies dans des bibliothèques externes. Cela permet d'étendre les fonctionnalités de Uiua en utilisant des bibliothèques écrites en Rust, C ou C++ par exemple.

Motivation du choix

Le paradigme est totalement différent de ce que nous avons l'habitude de voir ce qui force un changement drastique de réflexion. Les possibilités avec les types de médias nous ont aussi beaucoup intéressé pour le projet.

Projet

Notre projet consiste en une application graphique permettant de générer des images de manière paramétrique. L'interface graphique sera réalisée en utilisant la librairie `raylib` et permettra de démontrer les capacités de la `FFI`. La génération d'image permet de montrer les capacités de manipulation de données multi-dimensionnelles du langage, les fonctions de manipulation de format d'image ainsi que les fonctions système pour la sauvegarde de fichiers.

Aperçu du cahier des charges

- Interface graphique simple
 - Modification en temps réel des paramètres
- Génération d'images paramétriques
 - Formes pré-définies (cercles, carrés, ...)
 - Courbes paramétriques (Bézier)
- Sauvegarde des images générées
 - Choix du format (PNG, JPEG, BMP, ICO, QOI)
 - Choix du nom