

# **Créer des jeux textuels avec Inform 7**

Par farvardin, mis à jour par Natrium729

(dernière mise à jour : 20/10/2015)

Licence : [Creative Commons 4.0 CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Table des matières

Introduction.....	3
Préparation.....	5
Un petit exemple.....	5
Installation d'Inform 7.....	5
Notre premier code.....	8
Installation de l'extension francophone d'Inform 7.....	9
Créer un monde et les règles qui le régissent.....	13
Les lieux et les objets.....	13
Concepts de base.....	13
Les lieux.....	13
Les choses.....	15
Les propriétés.....	16
Les synonymes.....	18
Les règles.....	18
Les règles <i>instead</i> .....	19
Les règles <i>before</i> .....	20
Les règles <i>after</i> .....	21
Connaître le véritable nom des actions.....	21
Les conditions.....	22
Un jeu complet.....	25
Une vraie histoire.....	25
Quelques conseils d'organisation.....	25
Les en-têtes.....	25
Ordonner sa source.....	26
Créons notre jeu.....	26
Commençons par le commencement.....	27
À l'oasis.....	27
Dans la crypte.....	30
Notre première scène.....	32
La bataille finale.....	34
Version finale de notre jeu.....	37
Montrons notre jeu au reste du monde.....	44
Testons !.....	44
Les privilèges du démiurge.....	44
Test me !.....	44
Déplacer des objets.....	44
Publier son jeu.....	45
Des ressources pour aller plus loin.....	47
Conclusion.....	48

# Introduction

---

Inform 7 est un langage de programmation en langage naturel permettant de créer des mondes imaginaires, de les remplir d'objets et de personnages, et par ce biais, de créer des histoires, des aventures et des jeux, également nommés fictions interactives.

Mais à la différence des jeux modernes impliquant des effets 3D, des animations, des images et du son, les fictions interactives utilisent uniquement du texte pour communiquer avec le joueur et lui décrire son environnement, comme dans un livre ; l'unique moyen d'entrer des commandes pour le joueur est de les taper au clavier.

Cela n'enlève néanmoins pas la possibilité à l'auteur d'ajouter des images, des sons et une interface sophistiquée à sa fiction interactive, s'il le juge nécessaire.

Voici à quoi une fiction interactive peut ressembler. Il s'agit en fait d'un extrait de celle que nous allons créer pas à pas dans ce guide. Les commandes que le joueur a entrées sont écrites après le symbole « > ».

## **Crypte**

Un silence de mort règne entre ces murs oppressants, et le murmure enjoué du vent n'arrive pas jusqu'ici.

Vous pouvez voir une porte de pierre, une momie et un cimetière.

>prendre le cimetière

Ce sabre légendaire, le Shamshir, vous avait toujours fasciné, et maintenant vous pouvez enfin mettre la main dessus.

Au moment où vous prenez l'arme, la porte donnant vers l'extérieur se referme !

L'air commence à manquer...

>ouvrir la porte

La porte est trop lourde pour vous.

L'air commence à manquer..

Inform 7 est l'un des systèmes les plus efficaces et modernes pour créer ce type de jeux.

Il est composé :

- d'un environnement graphique de développement (IDE) pour écrire le code, interpréter le jeu et avoir accès à la documentation ainsi qu'à divers modules (par exemple, la liste des relations entre les objets, la liste des verbes, l'arborescence des choix possibles...);
- d'un compilateur permettant de créer le fichier d'histoire dans un langage machine spécial, qui sera par la suite interprété ;
- de bibliothèques permettant d'établir les règles de base qui régissent le monde.

Ce guide ne prétend pas se substituer au manuel fourni avec le logiciel, très complet même si uniquement en anglais pour le moment. C'est en revanche le premier du genre qui aborde, en français, une initiation à Inform 7 et qui donne les bases nécessaires pour concevoir un univers textuel. Le lecteur pourra ensuite se référer au manuel officiel, et à ses nombreux exemples commentés, pour en savoir plus.

À la fin de votre lecture, vous serez en mesure de créer vous-même une petite aventure textuelle. Pour avoir une idée de résultat final, vous pouvez jouer à *Sable* en ligne :

<http://ifiction.free.fr/parchment/?story=http://ifiction.free.fr/jeux/sable/sable.z5.js>

Si vous souhaitez avoir plus d'informations sur les fictions interactives ou jouer à d'autres jeux créés par les membres de la communauté francophone, rendez-vous sur son portail :

<http://ifiction.free.fr/>

À titre d'exemple des possibilités évoluées de ce langage, notons qu'il est possible de modéliser aisément les règles de la physique, de tester des équations et de les résoudre (nous n'aborderons pas cela ici).

# Préparation

---

## Un petit exemple

La principale originalité d'Inform 7 est d'être un langage naturel déclaratif au lieu d'un langage de bas niveau procédural, impératif, comme le C par exemple (ou comme Inform 6, le prédécesseur d'Inform 7).

Voici un exemple de code écrit en Inform 7 :

```
Les murs sont des choses décoratives dans la Crypte. La description est "Les murs
sont recouverts de carreaux en terre cuite qui représentent des dieux et des animaux
fabuleux. Une partie est restée en pierre brute."
```

```
Instead of inserting le cimenterre into les murs:
```

```
    si la portedepierre est verrouillée:
```

```
        dire "Vous glissez votre cimenterre dans l'interstice entre les deux pierres, et
la porte est de nouveau ouverte.";
```

```
        maintenant la portedepierre est déverrouillée;
```

```
        maintenant la portedepierre est ouverte;
```

```
    sinon:
```

```
        dire "Il ne se passe rien de spécial."
```

Même une personne n'ayant jamais fait de programmation pourrait deviner ce que fait l'extrait ci-dessus, et c'est là l'un des attraits d'Inform 7. Malheureusement, on peut aussi remarquer que certains passages n'ont pas pu être traduits dans notre langue.

En effet, jusqu'à récemment, il n'était possible d'écrire sa source qu'en anglais. Ce n'est que depuis la version 6L02 qu'il est possible d'obtenir une source presque entièrement en français ; les prochaines versions d'Inform 7 devraient permettre de l'écrire entièrement dans la langue de Molière. Gardez à l'esprit que la syntaxe en français n'est pas définitive, et qu'elle est encore susceptible de changer.

Mais trêve de bavardage, et entrons dans le vif du sujet.

Dans cet ouvrage, les extraits de code sont présentés en blanc sur fond noir. Il est recommandé de les taper dans Inform 7 avant de les lire et d'essayer de les comprendre ; en effet, Inform 7 colorera la source et il sera plus facile de vous y retrouver.

## Installation d'Inform 7

En premier lieu, téléchargeons la dernière version d'Inform 7 (6L38 lors de l'écriture de ce tutoriel) sur le site officiel :

<http://inform7.com/>

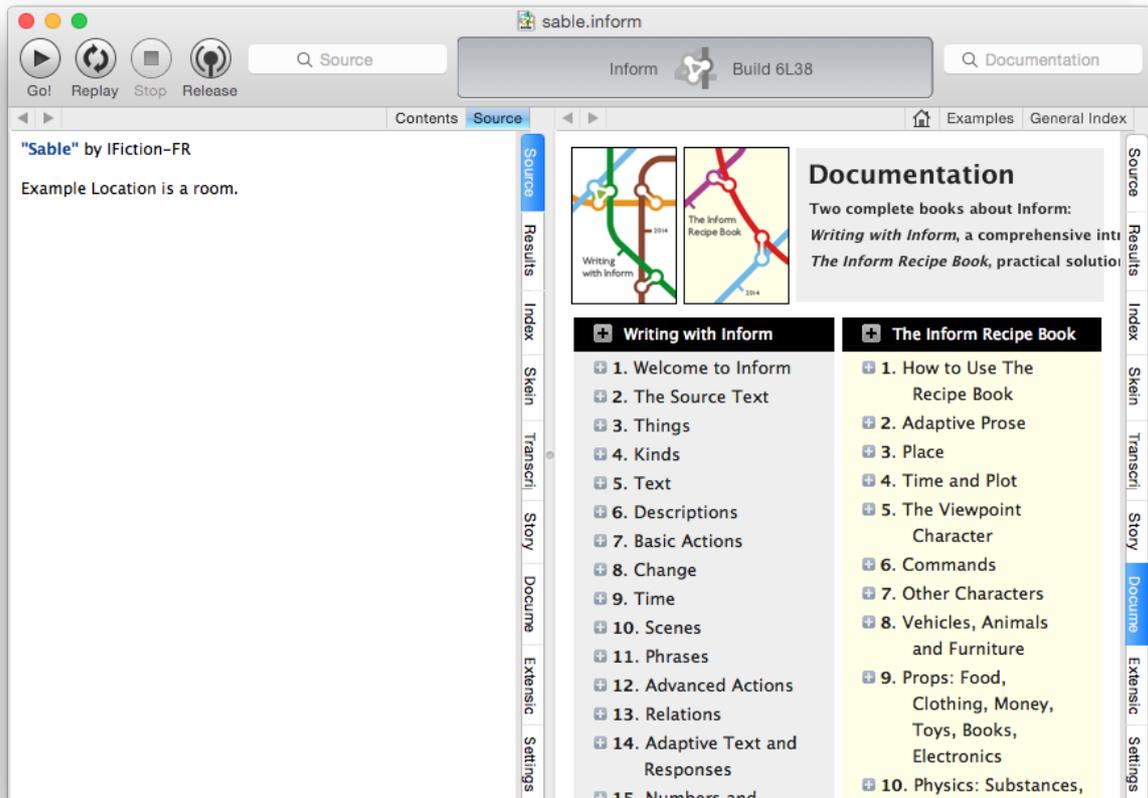


Il est conseillé de créer tous ses projets dans le même dossier :

- Windows : C:\Documents and Settings\vousre-nom\Mes documents\Inform\Projects
- OS X : /Users/votre-nom/Documents/Inform Projects
- Linux : ~/Inform/Projects

Appelons notre nouveau projet « Sable ». Choisissez également votre nom de plume.

La fenêtre principale s'ouvre alors. L'interface est séparée en deux zones, comme dans un livre ouvert.



Ces onglets sont :

- Source (le code source du jeu) ;
- Results (les messages d'erreurs ou d'autres événements s'étant déroulés lors de la compilation) ;
- Index (la liste des lieux, des objets, des verbes, etc.) ;
- Skein (le cheminement des parties que vous avez réalisées, sous forme d'arborescence) ;
- Transcript (la transcription des parties que vous avez réalisées) ;
- Story (où vous jouerez et testerez votre jeu) ;
- Documentation (les deux manuels — an anglais seulement — d'Inform 7) ;
- Extensions (où vous pourrez télécharger de nouvelles extensions et consulter leur documentation) ;
- Settings (les options).

Pour le moment, nous nous concentrerons sur les onglets « Source » et « Story », qui sont les plus importants. Lorsque vous aurez un peu plus d'aisance avec Inform (nous utiliserons à partir de maintenant les termes « Inform » et « Inform 7 » indifféremment), vous pourrez découvrir les autres modules (certains sont très optionnels, comme le skein et le transcript).

## Notre premier code

Les projets nouvellement créés contiennent le code source minimal requis. Adaptez-le afin d'obtenir le code suivant. Ce n'est pas grave si vous ne le comprenez pas encore, tout sera expliqué par la suite.

```
"Sable" by IFiction-FR
```

```
Oasis is a room. "Une belle oasis ensoleillée...".  
The dromadaire is an animal in Oasis.
```

```
When play begins, say "Bonjour, voyageur."
```

Cliquez sur le bouton « Go! » en haut à gauche de la fenêtre pour compiler le jeu. Vous verrez le résultat s'afficher sur la page de droite :

```
Bonjour, voyageur.
```

```
Sable
```

```
An Interactive Fiction by IFiction-FR
```

```
Release 1 / Serial number 150903 / Inform 7 build 6L38 (I6/v6.33 lib 6/12N) SD
```

## Oasis

Une belle oasis ensoleillée...

You can see a dromadaire here.

Argh, nous avons du français et de l'anglais mélangé et ce n'est pas ce que nous souhaitons !

En effet, Inform 7 étant à la base tout en anglais, il n'a pas encore moyen de savoir que nous voulons que le jeu soit en français pour toutes nos réponses, ce qui est normal. Aussi, à l'exception des noms et autres chaînes de caractères que nous avons écrits nous-mêmes en français, tout sera traité en anglais.

Mais même si ce n'est pas encore parfait, nous pouvons commencer dès maintenant à interagir avec cette histoire en français, en tapant des instructions (instructions pour le moment en anglais) après le signe « > » (appelé le prompt) ; le résultat s'affichera sur la ligne suivante.

Par exemple :

```
>examine the dromadaire  
You see nothing special about the dromadaire.
```

## Installation de l'extension francophone d'Inform 7

Inform 7 utilise un système d'extensions qui permet d'ajouter des fonctionnalités. Ainsi, pour franciser Inform 7, il suffit de télécharger l'extension « French Language.i7x », qui se trouve à cette adresse :

<https://bitbucket.org/informfr/i7-french-language/src>

Téléchargeons également l'extension « Experimental French Features.i7x » (nous verrons par la suite pourquoi).

Ensuite, allons dans les fichiers de l'application Inform et ouvrons le fichier « about.txt » situé ici :

- Windows : C:\Program Files\Inform 7\Internal\Languages\French\about.txt (vous devez avoir les droits administrateur pour pouvoir modifier le fichier.)
- OS X : clic droit sur l'icône d'Inform puis *Afficher le contenu du paquet*, puis /Contents/Ressources/Internal/Languages/French/about.txt
- Linux : /usr/share/gnome-inform7/Extensions/Reserved/Languages/French/about.txt ou /usr/share/gnome-inform7/Languages/French/about.txt

Modifions la ligne 5 pour remplacer « Eric Forgeot » par « Nathanael Marion » (cette étape ne devrait plus être nécessaire dans les prochaines versions).

Une fois que cela est fait, nous sommes prêts à installer les extensions. Retournons dans Inform et allons dans le menu :

File > Install extension...

et choisissons le fichier « French Language.i7x » que nous venons de télécharger. Répétons l'opération pour « Experimental French Features.i7x ».

L'extension est fonctionnelle, mais des bugs sont fréquemment corrigés. Assurez-vous régulièrement que vous possédez bien la dernière version à l'adresse ci-dessus !

Maintenant, ajoutons « (in French) » à la fin de la première ligne, ce qui permettra d'utiliser l'extension « French Language » :

```
"Sable" by IFiction-FR (in French)
```

Nous nous retrouvons donc avec ce code :

```
"Sable" by IFiction-FR (in French)
```

```
Oasis is a room. "Une belle oasis ensoleillée..."
```

```
The dromadaire is an animal in Oasis.
```

```
When play begins, say "Bonjour, voyageur."
```

Mais il y a mieux, nous pouvons également écrire une partie de notre source en français ! Ajoutons la ligne « Include Experimental French Features by Nathanael Marion » (nécessaire pour que certaines choses puissent être écrites en français) après la ligne de titre, et modifions notre code pour qu'il ressemble à ceci :

```
"Sable" by IFiction-FR (in French)
```

```
Include Experimental French Features by Nathanael Marion.
```

```
[Attention, il y a une espace entre le mot « Oasis » et son déterminant !]
```

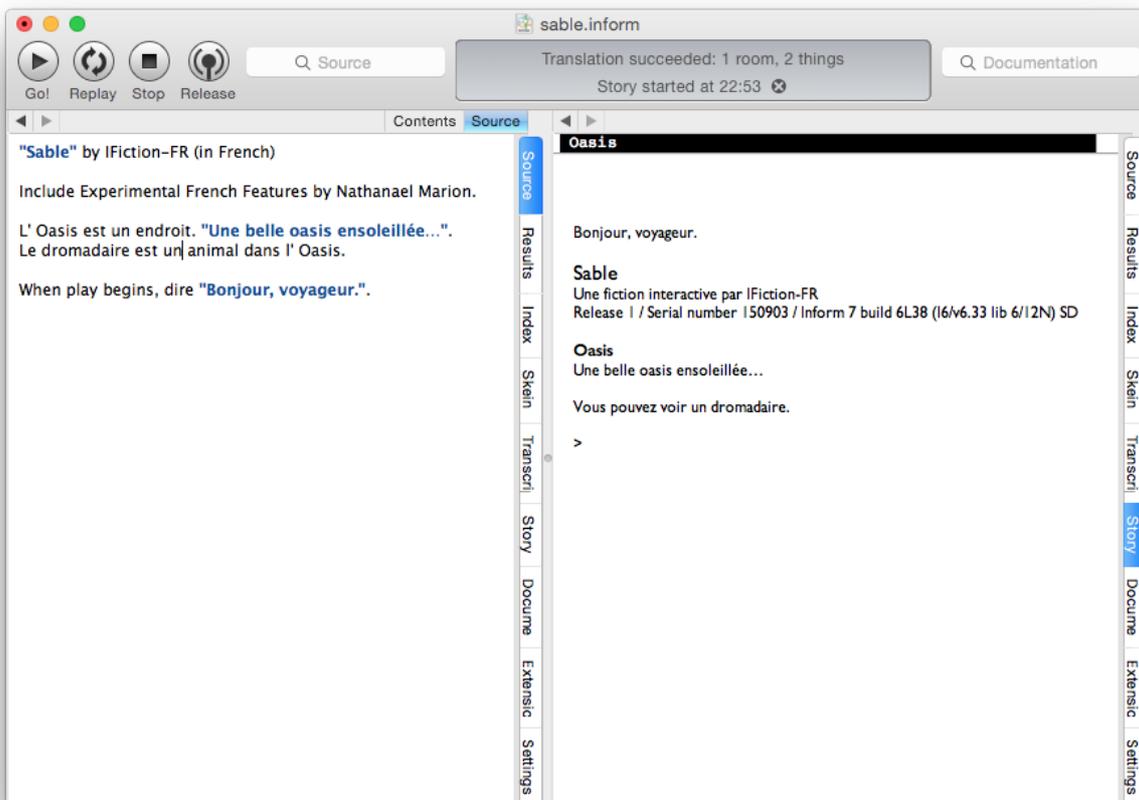
```
L' Oasis (f) est un endroit. "Une belle oasis ensoleillée..."
```

```
Le dromadaire est un animal dans l' Oasis.
```

```
When play begins, dire "Bonjour, voyageur."
```

(Dans la source, un texte entre crochets est un commentaire, c'est-à-dire une note que l'auteur a ajoutée, mais qui ne sera pas compilée. Par défaut, les commentaires sont affichés en vert dans Inform.)

Si nous compilons de nouveau le jeu avec « Go! », nous obtenons maintenant :



Bonjour, voyageur.

### **Sable**

Une fiction interactive par IFiction-FR

Release 1 / Serial number 150903 / Inform 7 build 6L38 (I6/v6.33 lib 6/12N) SD

### **Oasis**

Une belle oasis ensoleillée...

Vous pouvez voir un dromadaire.

Nous avons maintenant toutes les répliques du jeu en français ! De plus, nous pouvons interagir avec l'histoire en tapant nos commandes en français.

Par exemple :

>examiner le dromadaire

Rien de particulier concernant le dromadaire.

>embrasser le dromadaire

Le dromadaire n'aimera peut-être pas ça.

>le frapper

La violence n'est pas une solution ici.

Ici, comme nous n'avons rien précisé de plus, nous obtenons les réponses types prévues dans Inform 7 (et plus précisément dans les traductions de l'extension francophone).

Voici quelques précisions techniques sur ce que nous venons de faire. Il n'est pas nécessaire de les connaître pour poursuivre le tutoriel.

Ajouter « (in French) » à la première ligne fait en sorte que l'extension « French Language » est utilisée. Cette extension permet de traduire les réponses du jeu et les commandes du joueur en français. Elle permet également de taper une partie de sa source en français. C'est grâce à elle qu'il devient possible d'écrire « L' Oasis est un endroit » et « Le dromadaire est un animal dans l' Oasis ».

La ligne « Include Experimental French Features by Nathanael Marion » nous fait utiliser la seconde extension téléchargée plus tôt. Elle permet d'écrire une encore plus grande partie de son code en français (« dire "[...]" » en est un exemple). Cette extension est expérimentale dans le sens que les fonctionnalités qu'elle ajoute ne sont pas encore officiellement supportées par Inform. Cependant, il ne devrait pas y avoir un grand risque de l'utiliser. À terme, quand Inform supportera ces fonctionnalités, elles seront ajoutées dans l'extension « French Language ».

Pour plus de précisions sur ce qui est traduit et par quoi, consultez la documentation de ces extensions (onglet « Extensions »).

# Créer un monde et les règles qui le régissent

---

Dans ce chapitre, nous allons voir la syntaxe de base d'Inform, ce qui nous donnera tous les outils nécessaires pour créer un jeu, certes simple, mais complet. Expérimentez un peu avec les différents exemples donnés afin de vous assurer de bien comprendre le fonctionnement d'Inform, mais ne vous inquiétez pas si quelque chose vous échappe : le chapitre suivant vous accompagnera dans la création de votre première fiction interactive.

## Les lieux et les objets

### Concepts de base

Revenons un peu sur le code utilisé pour placer les objets dans le jeu, et expliquons-le un peu mieux, même si le langage naturel permet de comprendre ce qui a été fait.

Tout d'abord, il faut savoir que l'état initial du monde du jeu est créé à partir de phrases déclaratives écrites au présent. De plus, chaque objet, au sens large du terme, doit posséder un type particulier. Par exemple, l'oasis est de type « endroit », le dromadaire de type « animal ». Les types représentent l'essence même de l'objet, et déterminent ce qu'il est possible de faire avec lui.

Pour créer un objet, il faut donc écrire :

```
Nom de l'objet est un type de l'objet.
```

Commençons par les lieux où se déroulera l'histoire du jeu.

### Les lieux

Suivons la règle énoncée ci-dessus pour créer un endroit :

```
L' Oasis (f) est un endroit.
```

Le déterminant est très important : c'est lui qui indique à Inform le genre et le nombre de l'objet ou de l'endroit en question. Ici, comme le déterminant est ambigu (« l' »), il faut ajouter « (f) » après le nom de l'oasis pour qu'Inform sache que le mot est féminin. Il ne faut pas non plus oublier que, pour qu'Inform sache où s'arrête le déterminant et où commence le nom, il faut ajouter une espace après l'apostrophe dans des situations comme « l' Oasis ».

Ajoutons maintenant une description à notre nouvel endroit. Il y a deux méthodes y parvenir.

Nous pouvons écrire la description de l'endroit entre guillemets, juste après l'avoir créé (si vous ne l'aviez pas encore remarqué, tous les textes qui seront affichés en jeu doivent être écrits entre guillemets dans la source) :

```
L' Oasis est un endroit. "Une belle oasis ensoleillée..."
```

Ou bien nous pouvons spécifier explicitement quelle doit être la description :

La description de l' Oasis est "Une belle oasis ensoleillée..."

Les phrases pouvant facilement se permuter et se déplacer dans la majorité des cas, cela peut être une bonne idée de préciser à chaque fois la description d'un objet au lieu d'utiliser la version courte, mais c'est une question de préférence.

Nous sommes maintenant en mesure de créer plusieurs autres endroits :

La Crypte est un endroit. Le Désert est un endroit.

Mais il y a un problème. Le joueur commencera dans le premier endroit défini dans la source (ici, l'oasis), mais comment pourra-t-il se rendre autre part ?

Il faut relier les endroits entre eux. Nous pouvons faire cela en les situant les uns par rapport aux autres. Par exemple :

La Crypte est au sud de l' Oasis.  
Le Désert est au nord de l' Oasis, à l'ouest de l' Oasis et à l'est de l' Oasis.

Ces phrases créent les endroits s'ils n'ont pas été créés auparavant.

Les différentes connexions possibles sont :

- nord, sud, est, ouest ;
- nord-ouest, sud-ouest, nord-est, sud-est ;
- au-dessus, en dessous, à l'intérieur, à l'extérieur.

Testons cela :

### **Oasis**

Une belle oasis ensoleillée...

Vous pouvez voir un dromadaire.

>aller au sud

### **Crypte**

>nord

### **Oasis**

Une belle oasis ensoleillée...

Vous pouvez voir un dromadaire.

Une petite note sur la casse : les majuscules n'ont pas d'importance pour les références à un lieu ou un objet dans le code, mais le nom du lieu ou de l'objet s'affichera en jeu de la même façon que sa première occurrence dans la source. Il est conseillé d'écrire les lieux avec une majuscule et les objets tout en minuscules.

## Les choses

La chose est le type le plus basique qu'un objet peut avoir. Les choses sont utilisées pour représenter un peu tout et n'importe quoi, tant que l'objet ne possède pas de caractéristique particulière.

Créer un objet est aussi facile que créer un endroit, si ce n'est qu'il faut spécifier son emplacement :

Le palmier est une chose dans l' Oasis.

Pour créer un objet de type plus spécifique, il suffit de remplacer « chose » par « homme », « femme », « animal », « conteneur », ou tout autre type existant (voir l'index pour en obtenir la liste) :

Le dromadaire est un animal dans l' Oasis.

Cela permettra de donner des réponses types plus adaptées à la nature de ces objets (qu'ils soient vivants ou non) ainsi que de leur attribuer des propriétés spécifiques. Ainsi, pour un être vivant, la réponse par défaut à l'action « toucher » sera « Vous savez tenir vos mains tranquilles », alors que pour un objet inanimé, cela sera « Vous ne sentez rien de particulier ». De même, le joueur sera en mesure de mettre des objets dans un conteneur, mais pas dans une simple chose.

À présent, nous avons une petite oasis où se trouvent un palmier et un dromadaire.

D'ailleurs, ce palmier est très bien, mais il y a un petit problème de conception, que vous avez peut-être remarqué si vous avez testé le jeu : essayez de prendre le palmier avec la commande « prendre le palmier » ; la réponse « Voilà qui est fait » sera affichée. En consultant votre inventaire avec « inventaire » (abrégé par « i »), vous le trouverez effectivement dans vos affaires :

>i  
Vous avez :  
un palmier

Comment résoudre ce problème ? Utilisons des propriétés.

## Les propriétés

Les objets possèdent des propriétés, qui déterminent leur façon de se comporter. Sans le savoir, vous en avez déjà utilisé une : la description. En effet, la description est un texte qu'un objet possède, et qui est affiché quand le joueur le regarde.

Les propriétés les plus basiques sont des adjectifs, qui représentent l'état actuel dans lequel se trouve les objets qu'ils qualifient. De nombreux adjectifs sont déjà utilisables par défaut, dont un qui va nous permettre de résoudre le problème du palmier.

Car si les objets définis comme être vivants ne pourront pas être transportables par défaut (c'est une question de savoir-vivre...), ce n'est pas le cas de notre arbre.

Ajoutons-lui une caractéristique. Pour qu'il soit impossible de le prendre, il suffit de dire qu'il est « fixé sur place » :

**Le palmier est une chose dans l' Oasis. Il est fixé sur place.**

Il est possible de simplifier :

**Le palmier est une chose fixée sur place dans l' Oasis.**

Essayez de le prendre à présent. Le jeu vous répondra que c'est impossible.

Maintenant que le problème du palmier est résolu, ajoutons du réalisme et créons du sable. Nous savons comment faire pour qu'il soit impossible de le prendre :

**Le sable est fixé sur place dans l' Oasis.**

Mais ce n'est pas bon, car maintenant le jeu nous annonce :

Vous pouvez voir un dromadaire, un palmier et un sable.

Le déterminant indéfini pose problème : « un sable » n'est pas correct, puisqu'il faut un article partitif.

Pour résoudre cela, il faut modifier la propriété « indefinite article » du sable (notons au passage que cette propriété n'a pas pu être traduite) :

**L' indefinite article du sable est "du".**

Cela affichera un résultat correct, mais il est possible de faire mieux.

En effet, le sable n'est pas un objet que nous souhaitons mettre en avant lors des descriptions (c'est évident qu'il y a du sable dans un désert et une oasis) ; il peut être intéressant de le masquer dans les descriptions, tout en permettant au joueur d'interagir avec lui (l'examiner, le toucher, etc.)

Pour cela, nous utilisons la propriété « décoratif » :

**Le sable est une chose décorative dans l' Oasis.**

Un objet défini comme décoratif sera automatiquement fixé sur place, il n'est donc pas nécessaire de l'indiquer.

Mieux, si nous souhaitons que le sable soit présent dans plusieurs endroits à la fois, il faut changer son type « chose » en « toile de fond ». Une toile de fond est toujours décorative, et donc également fixée sur place.

Profitions-en pour ajouter un soleil.

Le sable est une toile de fond dans l' Oasis, dans le Désert et dans la Crypte.  
Le soleil est une toile de fond dans l' Oasis et dans le Désert.

Lorsque plusieurs objets (ici, des lieux) sont concernés par une action ou une définition, il est possible d'utiliser une virgule ainsi que « et » pour les séparer, cela évite d'écrire plusieurs phrases distinctes comme ceci :

« Le sable est une toile de fond dans l' Oasis. Le sable est une toile de fond dans le Désert. »

N'oubliez pas qu'un objet ne doit être une toile de fond que s'il se trouve dans plusieurs endroit à la fois. S'il s'agit d'un simple décor dans un endroit unique, la propriété «décoratif » suffit :

Le chemin est une chose décorative dans l' Oasis.

Il existe de nombreuses autres propriétés, mais nous les verrons au fur et à mesure de leur utilisation.

Enfin, en plus des propriétés existantes qui sont utilisées par les règles prédéfinies dans Inform, nous pouvons en créer nous-mêmes, et les utiliser dans des règles que nous écrirons. Nous verrons ceci plus tard, mais voyons brièvement comment créer une nouvelle propriété :

Une personne peut être mort. Une personne est rarement mort.  
Une personne peut être heureux. Une personne est généralement heureux.

Ces lignes feront en sorte que chaque personnage créé sera toujours vivant et heureux par défaut (ce qui pour l'instant ne sert à rien, car nous n'avons pas écrit de règles concernant ces propriétés).

Aussi, les mots « mort » et « heureux » ont été utilisés, mais l'auteur peut choisir les adjectifs qu'il veut, même « zedfzg » et « ftagg » s'il le souhaite vraiment, du moment qu'il s'y retrouve ensuite...

Les adjectifs dans l'exemple ci-dessus ont été créés au masculin, bien que le mot « personne » soit féminin. En fait, la version actuelle d'Inform ne reconnaît pas les différentes formes des adjectifs, c'est pourquoi il faut choisir la forme que l'on veut

utiliser dans sa source (par exemple, le masculin singulier). Si l'on avait écrit « morte », il aurait alors été nécessaire de l'écrire au féminin partout.

Ceci n'est pas valable avec les adjectifs définis par Inform, comme « décoratif » ou « ouvert » : il est donc possible d'écrire « décoratives » ou « ouverts ».

## Les synonymes

Nous avons enfin un petit cadre sympathique, mais il nous reste une nouvelle chose à régler pour rendre le jeu encore un peu plus attrayant : imaginons que nous créons un cimenterre (une sorte d'épée courbe) et que le joueur souhaite y faire référence en l'appelant « épée » ou « lame » (parce que c'est plus court, parce que qu'il ne se souvient plus de mot exact...).

Pour permettre l'utilisation de synonymes **en jeu uniquement**, il suffit d'ajouter :

Le cimenterre est une chose dans la Crypte. La description est "Un beau sabre à la lame courbe et damassée qui scintille de mille feux."

Comprendre "sabre", "epee", "lame", "courbe" ou "damas" comme le cimenterre.

Nous n'indiquerons jamais les accents dans les synonymes possibles, car l'extension française enlève les diacritiques des commandes du joueur avant de les interpréter. Un synonyme défini avec accents ne sera jamais utilisable.

À présent, le joueur pourra taper n'importe quelle combinaison des synonymes créés pour faire référence au cimenterre, comme « épée », « lame courbe » ou « sabre ».

Si votre objet possède des accents dans son nom, il faut ajouter un synonyme sans accents, sinon le joueur ne pourra jamais interagir avec l'objet !

Par exemple, si nous avons écrit « L' épée (f) est une chose », il aurait fallu ajouter « Comprendre "epee" comme l' épée », même si cela paraît redondant.

## Les règles

Nous pouvons maintenant créer tout un monde avec des lieux, des connexions, des objets, des personnages... Mais ce monde reste terriblement statique, car nous n'avons pas encore vu comment provoquer des réponses s'adaptant en fonction de l'action envisagée.

Il faut donc créer des règles pour rendre notre monde plus vivant. Nous en avons déjà vu une : la règle « When play begins », qui permet d'exécuter du code au début du jeu :

When play begins, dire "Une introduction..."

Ici, cela ne fait qu'afficher un petit texte d'introduction (grâce à l'instruction « dire »), mais nous pouvons imaginer placer là certaines initialisations supplémentaires.

Voyons à présent les règles les plus pratiques, celles qui peuvent modifier le déroulement d'une action.

Malheureusement, les noms des règles n'ont pas pu être traduits en français. Il faudra les écrire en anglais !

## Les règles *instead*

La première d'entre elles, et la plus utilisée, est la règle *instead* (« au lieu de » en anglais) :

```
Instead of taking NomDeLaChose: [« taking » est l'action pour « prendre »]  
    dire "D'abord il se passe ça."  
    dire "Puis il se passe ça."
```

Une règle *instead* se déclenche quand le joueur essaye l'action spécifiée (ici, prendre l'objet appelé NomDeLaChose). Cette dernière est alors interrompue et les instructions écrites dans la règle sont exécutées. Il s'agit donc d'une règle spécifique qui va empêcher la règle générale, qui ne provoque qu'un message bateau, de se dérouler.

La ponctuation pour les règles (*instead* ou autre) doit être comme suit :

- Le préambule (ici, le « Instead of... ») doit être suivi d'un deux-points.
- Chaque instruction doit être suivie d'un point-virgule, sauf la dernière.
- La dernière instruction doit être suivie d'un point, pour indiquer que la règle est terminée.
- Il est recommandé d'écrire une instruction par ligne pour plus de clarté et d'indenter (avec une tabulation) les instructions, mais ce n'est pas obligatoire.

Si la règle ne contient qu'une instruction, alors il est possible d'utiliser une virgule à la place du deux-points :

```
Instead of taking NomDeLaChose, dire "Ma réplique super drôle."
```

Ceci n'est valable que lorsqu'il n'y a qu'une instruction, et n'est pas obligatoire.

Attention de bien vérifier la ponctuation et la syntaxe dans Inform 7, car langage naturel ne veut pas dire que le logiciel comprend tout ce qui est écrit ! En général, les messages d'erreur générés, bien qu'en anglais, permettent de rapidement voir où se situe le problème, mais il peut arriver que certaines erreurs de ponctuation passent à la compilation du fait d'une structure considérée comme valide, mais que le jeu ne se comporte pas comme attendu.

Ce genre de règle pourra être utilisée pour empêcher l'affichage du message « C'est trop difficile à transporter » lorsque le joueur essaye de prendre du sable : bien que le sable est effectivement difficile à transporter, nous voulons souligner l'absence d'intérêt particulier à prendre du sable dans le désert. Écrivons cela :

Instead of taking le sable, dire "Il n'y a pas de technocrate ici pour essayer de lui vendre votre sable, et vous n'avez de toute façon pas de récipient pour l'emporter."

Le message par défaut sera ainsi remplacé par votre réplique super drôle !

Attention, les actions n'ont pas été traduites en français non plus, il faut donc utiliser leur nom anglais !

## Les règles *before*

Les règles *before* (« avant » en anglais) servent à intervenir avant qu'une action ait lieu, mais n'empêcheront pas son déroulement : par défaut, l'action continuera comme si de rien n'était.

Before taking NomDeLaChose, dire "Description de ce qu'il se passe."

Cette règle se déclenchera avant que le joueur prenne le NomDeLaChose. Si le joueur tente de prendre cet objet, le message « Description de ce qu'il se passe » sera affiché puis l'action continuera, le message par défaut sera affiché et le joueur aura l'objet dans son inventaire. Voici un exemple :

Before taking le cimenterre, dire "Ce sabre légendaire vous avait toujours fasciné, et maintenant vous pouvez enfin mettre la main dessus."

Pour éviter que ce message n'apparaisse à chaque fois que le joueur repose le sabre puis le reprend, ce qui ridiculiserait l'effet, nous pouvons préciser que la règle ne peut être appliquée que la première fois que le cimenterre est pris, avec la précision « pour la première fois » :

Before taking le cimenterre pour la première fois, dire "Ce sabre légendaire vous avait toujours fasciné, et maintenant vous pouvez enfin mettre la main dessus."

## Les règles *after*

Si l'on veut que quelque chose se passe après qu'une certaine action soit réussie, il faut utiliser une règle *after* (« après » en anglais) :

```
After taking le cimenterre, dire "Bon ben finalement c'est pas si impressionnant que cela de porter ce sabre, mais qu'est-ce qu'il est lourd !".
```

Si l'on ne n'écrit pas l'instruction « continuer l'action » dans une règle *after*, alors le message par défaut ne sera pas affiché (vous n'avez qu'à tester pour vérifier !).

## Connaître le véritable nom des actions

Il est important de faire la différence entre une *action* et une *commande*.

Quand le joueur tape la commande « prendre le cimenterre », c'est l'action *taking* qui a lieu. Le joueur aurait pu taper « ramasser le cimenterre », et la même action *taking* aurait eu lieu.

Dans les règles que nous venons de voir, il **faut** utiliser le nom de l'action (*taking*), et non la commande (« ramasser » ou « prendre »). Il existe deux méthodes simples pour trouver le nom exact d'une action

La première consiste à taper la commande de débogage « actions » en jeu, puis de tester différentes actions. Par exemple, supposons que l'on cherche à chevaucher le dromadaire : quels noms d'action utiliser pour qu'Inform comprenne cela ?

Essayons la méthode :

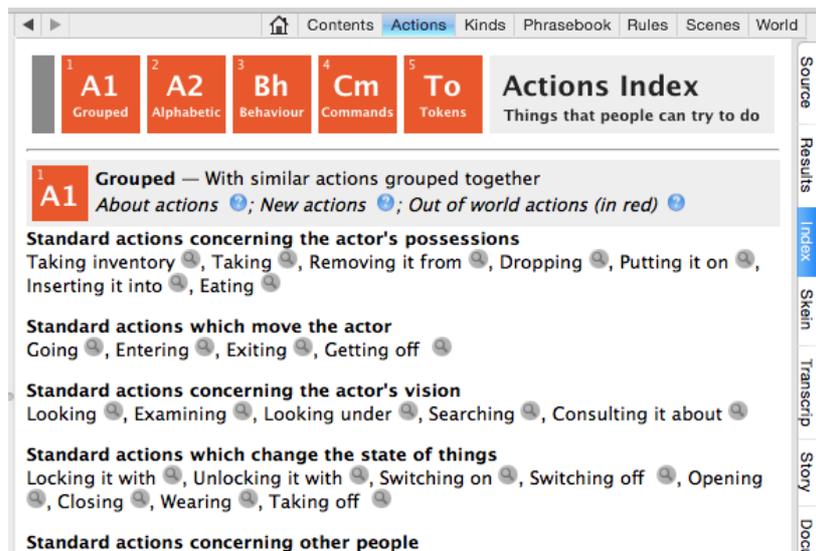
```
>actions
Actions listing on.

>grimper sur le dromadaire
[climbing le dromadaire]
Vous ne pouvez pas arriver à grand-chose de cette manière.
[climbing le dromadaire - failed the block climbing rule]

>monter sur le dromadaire
[entering le dromadaire]
Vous ne pouvez pas y aller.
[entering le dromadaire - failed the French can't enter what's not enterable rule]
```

On sait maintenant que ces commandes entraînent les actions *climbing* et *entering* (le verbe « entrer » peut sembler étrange ici, mais c'est un verbe générique utilisé pour entrer dans un fauteuil ou monter sur une plate-forme).

La deuxième méthode consiste à aller dans l'index (onglet « Index »), puis à cliquer sur « Actions » en haut à droite. Une liste des actions possibles en jeu apparaîtra.



Concrétisons donc cela :

Instead of climbing or entering le dromadaire, dire "L'animal ne se laisse pas faire."

Remarquons au passage qu'il est possible de créer une règle s'appliquant à plusieurs actions à la fois en écrivant « or » (« ou » en anglais) entre chacune des actions.

## Les conditions

Il ne nous manque plus qu'un concept essentiel pour pouvoir voler de nos propres ailes : les tests et les conditions. Cela va se faire avec des « si ».

Mais avant tout, il va falloir copier le code qui suit au tout début de notre source, juste après « Include Experimental French Features by Nathanael Marion ». La traduction des conditions ne fonctionnera pas sinon. Cela est dû au fait que certaines parties d'Inform ne peuvent pas être traduites de manière « standard ». Les prochaines versions d'Inform devraient corriger le problème.

```

Include (-
language French

<control-structure-phrase> ::=
  /a/ si ... est début |
  /b/ si ... est |
  /c/ si ... |
  /c/ à moins que |
  /d/ répéter ... |
  /e/ tant que ... |
  /f/ sinon |

```

```
/g/ sinon si ... |  
/g/ sinon à moins que |  
/h/ -- sinon |  
/i/ -- ...
```

-) in the Preform grammar.

Vous n'avez pas à comprendre ce code (il s'agit de quelque chose de beaucoup trop technique), copiez-le simplement dans votre source. Il faut reconnaître qu'il est plutôt embêtant d'avoir à se trimbaler ce code dans chacun de nos projets, alors si cela vous gêne, vous pouvez utiliser les conditions en anglais (« if », « else ») à la place de leur traduction.

Ceci étant dit, voyons un exemple de la syntaxe utilisée :

Instead of attacking le dromadaire:

dire "Attaquer le dromadaire serait-il une bonne idée ?";

si le joueur a le cimenterre:

    dire "Il est vrai que vous avez de quoi tuer la bête.";

sinon:

    dire "Vous pourriez frapper la bête avec vos poings.";

    dire "Mais vous n'êtes pas si cruel.".

Voici la syntaxe à suivre :

- Commencer la condition par « si *condition*: » (ne pas oublier le deux-points). Il est possible de tester plusieurs conditions en même temps en les liant avec « et » ou « ou », comme « si le joueur porte le cimenterre et le joueur est dans l' Oasis: ».
- Écrire une instruction par ligne sous le « si ». Ces instructions **doivent** être indentées un cran plus profond que le « si » (ne pas oublier les points-virgules). Elles ne seront exécutées que si la condition est vraie.
- Si l'on veut que quelque chose se passe si la condition est fausse, écrire « sinon: » (ne pas oublier le deux-points). Le « sinon » doit être indenté au même niveau que le « si ». Ajouter un « sinon » est facultatif.
- Écrire une instruction par ligne sous le « sinon ». Ces instructions doivent être indentées un cran plus profond que le « sinon » (ne pas oublier les points-virgules). Elles ne seront exécutées que si la condition est fausse.
- Pour exécuter des instructions dans tous les cas, sans se soucier de la condition, écrire la suite de ses instructions en revenant à l'indentation du « si ».

Si ce n'est pas encore clair, relisez bien l'exemple ci-dessus, testez-le et essayez de le modifier. Il est très important que vous compreniez cette structure, qui est utilisée *très* souvent.

Un peu comme pour les règles *instead*, *before* et *after*, s'il n'y a qu'une instruction à exécuter, il est possible (mais non obligatoire) de remplacer le deux-points par une virgule : « si le joueur a le cimenterre, dire "Il est vrai que vous avez de quoi tuer la bête." ».

Aussi, il arrive souvent de vouloir qu'une partie d'un texte change en fonction d'une certaine condition. Il est alors possible d'utiliser la syntaxe suivante, intégrée directement dans le texte :

La description du cimenterre est "Un beau sabre à la lame courbe et damassée qui scintille de mille feux[si le joueur porte le cimenterre]. Vous êtes vraiment heureux de l'avoir trouvé[sinon] est planté dans le sol[fin si].".

Tout ce qui est entre crochets dans du texte **n'est pas un commentaire** et ne s'affichera pas littéralement. Avec cette construction, le texte sera différent, selon une condition : tout ce qui se trouve entre « [si ...] » et « [sinon] » ne s'affichera que si la condition est vraie, sinon c'est ce qui est entre « [sinon] » et « [fin si] » qui sera affiché. Encore une fois, le « [sinon] » est optionnel.

Ici, si le joueur a le cimenterre, la description de celui-ci sera « Un beau sabre à la lame courbe et damassée qui scintille de mille feux. Vous êtes vraiment heureux de l'avoir trouvé. » S'il ne le porte pas, il verra « Un beau sabre à la lame courbe et damassée qui scintille de mille feux est planté dans le sol. ».

# Un jeu complet

---

## Une vraie histoire

Nous sommes à présent capables de créer un petit monde réagissant aux actions du joueur, mais nous n'avons pas encore de fil conducteur pour le motiver. Il est toujours bien d'élaborer au moins un scénario qui contiendra les actions les plus importantes que le joueur devra effectuer. Voici, dans les grandes lignes, ce que nous allons faire :

*Le héros, c'est-à-dire le joueur, est le prince de Perse, le fameux « prince of Persia ». Il arrive dans une oasis, entourée de désert, et s'apprête à entrer dans une crypte ancienne contenant un cimetière mythique nommé le « Shamshir ». Lorsqu'il pénètre dans la crypte, la porte se referme et le seul moyen de sortir est de fouiller les murs jusqu'à trouver une encoche servant d'interrupteur pour rouvrir la porte. Lorsque celle-ci s'ouvre, un div, démon persan, apparaît. Le héros doit s'en débarrasser avec son arme nouvellement acquise. Une fois que ceci est fait, il peut repartir, mais pas à dos du dromadaire qui se trouve dans l'oasis : il doit trouver son cheval, ce qui convient mieux à sa lignée. Le jeu peut alors se terminer avec une victoire du joueur.*

## Quelques conseils d'organisation

### Les en-têtes

Il est possible d'utiliser des en-têtes dans sa source, qui permettent de la séparer et de l'organiser.

Part 1 – L'Oasis

L' Oasis (f) est un endroit. "Une belle oasis ensoleillée..."

Chapter 1.1 – Le dromadaire

Le dromadaire est un animal dans l' Oasis.

[...]

Un en-tête est une ligne qui commence par le mot « Volume », « Book », « Part », « Chapter » ou « Section » (en ordre d'importance). La numérotation et le titre ne sont pas importants, mais un en-tête « Volume » englobera tous les en-têtes de niveau inférieur, et ainsi de suite. Rien ne vous force à toutes les utiliser, ou même à les utiliser du tout, mais l'organisation de la source est importante.

## Ordonner sa source

Une façon standard d'organiser son code est d'écrire les choses dans cet ordre :

- Une section qui inclut toutes les extensions nécessaires et qui fait les initialisations requises par le jeu ;
- Une section pour les règles générales que nous voulons ajouter ;
- Et enfin, une section par endroit, avec des sous-sections pour chaque objet si nécessaire.

Bien évidemment, ce modèle est à adapter en fonction du jeu et des besoins, et rien ne nous force à le suivre.

Voici une suggestion d'organisation pour notre premier jeu :

```
Part 1 - Configurations et initialisations générales
Chapter 1 - les extensions
Chapter 2 - La propriété Mort
Chapter 3 - Les objets initialement hors jeu
Part 2 - Les objets et les lieux
Chapter 1 - L'oasis
Chapter 2 - La crypte
Part 3 - Les scènes
Chapter 1 - Enfermé
Chapter 2 - Bataille
Part 4 - Tests
```

Ne vous inquiétez pas si vous ne savez pas encore ce que certaines sections vont contenir, nous verrons cela au fur et à mesure de l'écriture de notre code.

## Créons notre jeu

Nous pouvons enfin commencer à créer notre fiction interactive.

Chacun de ses éléments seront décrits en détail dans les sections suivantes, et les nouvelles notions nécessaires à l'élaboration du jeu seront expliquées. De plus, chaque section contiendra un exemple de code possible pour arriver à nos fins ; il est recommandé d'essayer d'implémenter seul ce qui est demandé à chaque section avant de regarder le code correspondant !

Enfin, sauvegardez et tester souvent votre jeu : il est plus facile de trouver une erreur si vous le testez après chaque fonctionnalité implémentée. Si vous le testez à la toute fin, il y a d'énormes chances que vous vous retrouviez avec une montagne d'erreurs d'un coup à résoudre.

## Commençons par le commencement

Tout d'abord, incluons les extensions nécessaires :

```
"Sable" by IFiction-FR (in French)
```

```
Part 1 - Configurations et initialisations générales
```

```
Chapter 1 - Les extensions
```

```
Include Experimental French Features by Nathanael Marion.
```

```
[Pour faire fonctionner les conditions en français.]
```

```
Include (-  
language French
```

```
<control-structure-phrase> ::=
```

```
  /a/ si ... est début |
```

```
  /b/ si ... est |
```

```
  /c/ si ... |
```

```
  /c/ à moins que |
```

```
  /d/ répéter ... |
```

```
  /e/ tant que ... |
```

```
  /f/ sinon |
```

```
  /g/ sinon si ... |
```

```
  /g/ sinon à moins que |
```

```
  /h/ -- sinon |
```

```
  /i/ -- ...
```

```
-) in the Preform grammar.
```

## À l'oasis

Concentrons-nous maintenant sur le chapitre portant sur l'oasis. Nous afficherons une petite introduction qui explique la situation initiale au joueur et nous créons l'oasis.

```
When play begins, dire "Bienvenue, aventurier. Vous êtes le prince de Perse, et au cours d'une partie de chasse dans la forêt, votre cheval vous a désarçonné. Vous avez alors erré à sa recherche, jusqu'à atteindre une oasis."
```

```
L' Oasis (f) est un endroit. "C'était une belle oasis ensoleillée mais, maintenant, le soleil va déclinant... Il y a le désert tout autour d'ici, à l'exception d'une forêt à l'est ; au sud descend un chemin en direction d'une crypte. La forêt n'est pas sûre en fin de journée."
```

Il est judicieux d'indiquer les directions dans lesquelles le joueur pourra aller dans les descriptions des endroits ; sans cela, il serait perdu et ne saurait pas trop quoi faire. Ne l'oubliez pas !

Implémentons maintenant le dromadaire :

- Il faut qu'il soit un animal.
- Un message approprié doit être affiché et l'action doit être stoppée si le joueur tente de grimper sur le dromadaire.
- Le dromadaire tue le joueur si celui-ci l'attaque.

Nous aussi voulons que le message lorsque le joueur essaie de grimper sur le dromadaire change si le cheval n'est pas encore arrivé. Par exemple :

```
Instead of climbing or entering le dromadaire, dire "Vous, un prince, n'oseriez certainement pas retourner à votre palais à dos de dromadaire ![si le cheval n'est pas dans l' Oasis]Mais où est donc votre cheval ?[fin si]".
```

Puisque la condition mentionne le cheval, il faut que le cheval soit créé quelque part dans le code pour que le jeu fonctionne.

Pour terminer la partie, il faut utiliser l'une de ces instructions :

```
[Si le joueur a perdu, ou qu'il s'agit d'une « mauvaise » fin]  
fin de l' histoire en disant "Vous êtes mort";
```

```
[Si le joueur a gagné, ou qu'il s'agit de la « vraie » fin]  
fin définitive de l'histoire en disant "Vous avez gagné";
```

Le message est à adapter en fonction de ce qu'il s'est passé dans le jeu.

Voici le résultat final pour le dromadaire :

```
Le dromadaire est un animal dans l' Oasis.
```

```
Instead of climbing or entering le dromadaire, dire "Vous, un prince, n'oseriez certainement pas retourner à votre palais à dos de dromadaire ![si le cheval n'est pas dans l' Oasis]Mais où est donc votre cheval ?[fin si]".
```

```
Instead of attacking le dromadaire:
```

```
dire "Le dromadaire n'a pas apprécié, et il vous le rend bien, avec un coup de sabot bien placé.";  
fin de l'histoire en disant "Vous êtes mort".
```

Créons aussi le cheval du joueur pour que la condition dans le texte fonctionne. Comme il n'est pas présent au début de la partie, nous ne spécifierons pas d'emplacement lors de sa création (nous verrons comment le placer en jeu en cours de partie plus tard).

Le cheval est un animal.  
La description est "Votre cheval est un splendide Akhal-Teke."  
Comprendre "akhal" ou "akhal teke" comme le cheval.

Ajoutons le palmier (rien de bien sorcier ici) :

Le palmier est une chose fixée sur place dans l' Oasis.  
La description est "Un palmier qui oscille au gré du vent."  
Comprendre "arbre" comme le palmier.

Ajoutons comme décor dans l'oasis un chemin vers la crypte. Nous pourrions écrire « le chemin vers la crypte est une chose dans l' Oasis », mais cela pourrait causer un conflit avec l'endroit « Crypte » que nous allons bientôt créer, donc nous écrirons « le chemin est une chose décorative dans l' Oasis ». Il reste néanmoins possible de modifier le nom qui sera affiché en jeu avec la propriété « printed name », qui est analogue à la propriété « description ».

Voici le code obtenu :

Le chemin est une chose décorative dans l' Oasis.  
Le printed name est "chemin vers la crypte".  
[Le printed name est une propriété similaire à la description, mais pour le nom qui sera affiché en jeu.]

Au tour du sable, maintenant :

- Il doit être une toile de fond dans l'oasis, la crypte et le désert (ajoutez également ces deux derniers endroits, nous les modifierons plus tard).
- Le prendre et le fouiller doivent donner des messages appropriés et bloquer l'action (rappelez-vous le moyen de trouver à quelle action correspond la commande « fouiller »).

Le code final pour le sable :

Le sable est une toile de fond dans l' Oasis, dans le Désert et dans la Crypte.  
L' indefinite article du sable est "du".  
  
Instead of taking le sable, dire "Il n'y a pas de technocrate ici pour essayer de lui vendre votre sable, et vous n'avez de toute façon pas de récipient pour l'emporter."  
  
[Pour « fouiller le sable ».]  
Instead of searching le sable, dire "Autant chercher une aiguille dans une botte de

foin."

Le Désert est au nord de l' Oasis et au nord-ouest de l' Oasis et à l'ouest de l' Oasis.

[Un petit bonus : permet d'expliquer pourquoi le joueur ne peut aller dans une direction qui ne mène à aucun endroit depuis le désert.]

Instead of going nulle part from le désert, dire "Sans cheval, vous ne pouvez risquer de vous aventurer dans le désert seul."

Ajoutons également une forêt à l'est de l'oasis, mais en empêchant le joueur de s'y rendre : l'endroit n'est pas sûr à la tombée de la nuit. On peut pour cela ajouter « to la forêt » après l'action *going* dans la règle.

La Forêt est un endroit. La Forêt est à l'est de l' Oasis.

Instead of going to la forêt, dire "La nuit va bientôt tomber, et la jungle est la plus dangereuse à ce moment ; autant rester dans le désert, essayer de retrouver votre cheval ou explorer les environs."

## Dans la crypte

Maintenant que l'oasis est terminée, penchons-nous sur la crypte.

Il faut d'abord créer une porte de pierre (en utilisant le type *porte*), qui se situera entre la crypte et l'oasis. Il faut qu'elle soit ouverte, mais qu'il soit impossible pour le joueur de l'ouvrir ou de la fermer. Voici à quoi peut ressembler son implémentation :

[On ne peut pas écrire « porte de pierre », car il y aurait une confusion avec le verbe « porter ».

Nous allons donc l'appeler « portedepierre » dans notre code.]

La portedepierre est une porte ouverte.

La portedepierre est au nord de la Crypte et au sud de l' Oasis.

Le printed name de la portedepierre est "porte de pierre".

Comprendre "porte", "seuil" ou "crypte" comme la portedepierre.

[Permet d'autoriser le joueur à quitter la crypte lorsqu'il tape « sortir » plutôt que « aller au nord ».]

Instead of exiting quand l'emplacement est la crypte, essayer de going nord.

[L'instruction « essayer de » force le joueur à faire une action, ici aller au nord.]

Instead of opening or closing la portedepierre, dire "La porte est trop lourde pour vous[si la portedepierre est ouverte], de plus elle est déjà ouverte[fin si]."

Maintenant que ceci est fait, attaquons-nous à la suite. Nous aurons besoin de créer une nouvelle propriété pour les personnages morts. Nous voulons que les personnes

puissent être mortes, et qu'interagir avec une personne morte bloque l'action (n'oubliez pas d'écrire votre code dans la bonne section) :

Une personne peut être morte. Une personne est rarement morte.

[« Rarement » signifie que les personnes ne seront pas mortes par défaut.]

Instead of doing something to une mort personne, dire "Cette créature est morte, laissons-la en paix."

[Il arrive qu'Inform ne comprenne pas quand l'adjectif est après le nom, c'est pourquoi il est écrit « to une mort personne ».]

Revenons à la crypte. Créons-la, ajoutons-lui une description, une momie et le fameux cimetière. Il faut qu'un message personnalisé apparaisse lorsque le joueur prend le cimetière pour la première fois. La momie sera considérée comme une personne et non une chose, afin de pouvoir lui attribuer la propriété que nous venons de créer (ce qui affichera le message adéquat si le joueur interagit avec elle).

Cela donne :

La Crypte est un endroit. "Un silence de mort règne entre ces murs oppressants, et le murmure enjoué du vent n'arrive pas jusqu'ici."

La momie est une personne morte dans la Crypte.

[On affiche un petit texte avant d'examiner la momie. On n'utilise pas la description, car elle ne s'affichera jamais, puisque toute action sur personne morte est bloquée par la règle que nous avons définie plus tôt.]

Before examining la momie, dire "En temps normal, cela ne ferait pas aussi peur, mais il faut bien avouer qu'en ces lieux, l'évocation de la mort prend une autre dimension."

Le cimetière est une chose dans la Crypte.

La description est "Un beau sabre à la lame courbe et damassée qui scintille de mille feux."

Comprendre "sabre", "epee", "lame", "shamshir", "courbe" ou "damas" comme le cimetière.

[Nous remplaçons le message par défaut lorsque le joueur prend le cimetière, sans pour autant l'empêcher de le prendre.]

After taking le cimetière pour la première fois:

dire "Ce sabre légendaire, le Shamshir, vous avait toujours fasciné, et maintenant vous pouvez enfin mettre la main dessus."

## Notre première scène

Le scénario que nous avons conçu plus tôt nous dit que la porte doit se refermer après le joueur a pris le cimenterre. Décidons que le joueur meurt d'asphyxie s'il n'arrive pas à ouvrir la porte à temps.

Si nous savons comment organiser spatialement le jeu, nous n'avons pas encore abordé comment l'organiser temporellement. Les scènes sont ce qu'il nous faut.

L'utilisation de scènes n'est pas nécessaire pour créer un jeu fonctionnel avec Inform, car il est possible d'utiliser des tests sur des propriétés d'objets ou des variables à la place. Néanmoins, les scènes apportent un confort d'utilisation qu'il serait dommage de négliger.

Les scènes dans Inform sont analogues aux scènes dans une pièce de théâtre : elles commencent à un événement précis, se terminent à un événement précis, et des choses se passent lors de leur déroulement. Elles aident ainsi à découper et organiser le temps. Par exemple, lorsque tel personnage ouvre son courrier ou pousse un cri, un autre comédien arrive sur scène, et ainsi de suite.

Créons une scène nommée « Enfermé » et donnons les conditions qui la font débiter et se terminer.

Enfermé est une scène.

Enfermé begins when le joueur porte le cimenterre.

[La scène commence quand le joueur porte le cimenterre.]

Enfermé ends sadly when the time since Enfermé began is 10 minutes.

[Elle se termine mal quand dix minutes se sont passées depuis son commencement (1 min = 1 tour).]

Enfermé ends happily when la portedepierre est ouverte.

[Elle se termine bien quand la porte de pierre est ouverte.]

Nous pouvons voir que nous avons spécifié deux fins pour notre scène : l'une d'elles fera mourir le joueur, et l'autre le permettra de continuer (les noms des différentes fins importent peu, nous pouvons choisir ceux que nous voulons). Ceci, bien que pratique, est optionnel: il est possible de créer une scène ne possédant qu'une seule fin (il n'est dans ce cas plus nécessaire de lui donner un nom).

De plus, dans le code ci-dessus, nous avons décidé que la scène commence quand le joueur porte le cimenterre. Cela fera-t-il en sorte que la scène recommence encore et encore s'il pose le cimenterre et le reprend ? Non, car une scène ne peut arriver qu'une seule fois (par défaut du moins). Il n'y a donc pas de problème de ce côté.

Aussi, une scène ne peut pas se terminer avant d'avoir commencé, donc le fait qu'elle prenne fin quand la porte est ouverte n'est pas dérangent par rapport au fait que la porte est déjà ouverte au début du jeu, alors que la scène n'a pas encore commencé.

Enfin, une scène ne peut pas finir de plusieurs façons à la fois, donc, si les 10 minutes finissent de s'écouler alors que la scène s'est déjà terminée joyeusement, rien ne se passera.

Définissons à présent ce qu'il se passe lorsque la scène commence et se termine. Il faut que la porte se referme et se verrouille, et qu'un message l'indique au joueur. Si elle se termine mal, nous faisons mourir le joueur d'asphyxie. Si elle se termine bien, nous amenons le div dans l'oasis, afin de préparer la bataille finale.

Créons d'abord le div. Il doit se trouver hors du jeu, comme le cheval.

Le div est un animal.

La description est "Un démon sanguinaire à la face simiesque et aplatie qui ne vous fera pas de quartier."

Comprendre "monstre", "demon" ou "sanguinaire" comme le div.

Puis spécifions les événements liés à la scène :

When Enfermé begins: [quand la scène commence.]

dire "Au moment où vous prenez l'arme, la porte donnant vers l'extérieur se referme !";

maintenant la portedepierre est fermée;

maintenant la portedepierre est verrouillée.

When Enfermé ends sadly: [quand la scène se termine mal.]

dire "Vous ne pouvez plus respirer. Cette crypte sera votre tombeau.";

fin de l'histoire en disant "Vous êtes mort".

When Enfermé ends happily: [quand la scène se termine bien.]

maintenant le div est dans l' Oasis.

L'instruction « maintenant ... » permet de changer un état dans le monde (rendre la porte fermée, ou déplacer le div, par exemple).

Affichons aussi un message à chaque tour pendant la scène pour indiquer au joueur que son temps est compté :

[Une règle « Every turn » s'exécute à chaque tour.

Celle-ci en particulier s'exécute à chaque tour, mais seulement si Enfermé est en cours.]

Every turn during Enfermé: [« during » signifie « pendant »]

dire "L'air commence à manquer...".

La scène est fonctionnelle, mais il faut donner un moyen au joueur d'ouvrir la porte. Ajoutons des murs dans la crypte, avec des synonymes appropriés, et récompensons le joueur qui les fouille en lui indiquant qu'il y a un interstice entre deux pierres. Il faut que, quand le joueur insère le cimenterre dans l'interstice, la porte s'ouvre (ce qui terminera la scène).

Lors de la conception de ce genre d'énigme, il est toujours bon de penser à donner au joueur toutes les informations dont il aura besoin (subtilement ou pas), même si elles peuvent sembler évidentes. Par exemple, si nous n'avions pas mentionné les murs dans la descriptions de la crypte, le joueur n'aurait peut-être pas pensé à les toucher, et il n'aurait donc jamais trouvé la solution de l'énigme.

Vous devriez obtenir quelque chose comme ceci :

Les murs sont des choses décoratives dans la Crypte.

La description est "Les murs sont recouverts de carreaux en terre cuite qui représentent des dieux et des animaux fabuleux. Il y a également une partie qui est restée en pierre brute."

Comprendre "dieux", "animaux", "fabuleux", "mur", "interstice", "pierre" ou "pierres" comme les murs.

Instead of searching or touching les murs, dire "À un certain endroit, il y a un interstice plus marqué entre deux pierres."

Instead of inserting le cimenterre into les murs:

si la portedepierre est verrouillée: [alors la porte s'ouvre]

dire "Vous glissez votre cimenterre dans l'interstice entre les deux pierres, et la porte est de nouveau ouverte.";

maintenant la portedepierre est déverrouillée;

maintenant la portedepierre est ouverte;

sinon: [rien ne se passe]

dire "Il ne se passe rien de spécial."

[Il est aussi possible d'ajouter des synonymes à une action pour permettre plus de choix au joueur.]

Comprendre "planter [cimenterre] dans [quelque chose]" ou "glisser [cimenterre] dans [quelque chose]" comme inserting it into.

## La bataille finale

Nous sommes prêts pour la dernière partie du jeu : la combat contre le div !

Nous aurons besoin d'un moyen de savoir où en est rendu le combat. Pour cela, nous allons créer une nouvelle propriété pour notre div. Il s'agira d'un nombre que nous appellerons « attaque » :

Le div has a number called attaque. L' attaque du div est 1.

Il faut aussi faire gagner le joueur quand il monte sur son cheval :

Instead of climbing or entering le cheval:

dire "Vous remontez sur votre cheval, et retournez enfin à votre palais."  
fin définitive de l'histoire.

Et enfin, la scène finale :

- Elle commence quand le joueur et le div se trouvent dans l'oasis. Nous affichons alors un message qui indique que le div se rue sur le joueur.
- Elle se termine quand le div est mort. Nous déplaçons alors le cheval dans l'oasis, pour que le joueur puisse terminer le jeu.
- À chaque tour durant la bataille et si le joueur est dans l' Oasis, l'attaque du div augmente de 1 si elle est plus petite que 4 et un petit texte indiquant qu'il attaque le joueur est affiché. Si l'attaque du div est plus grande que 4, il dévore le joueur et la partie est finie.
- Pour battre le div, le joueur doit l'attaquer avec le cimenterre. S'il ne possède pas l'arme, on lui indique qu'il n'arrivera pas à grand-chose. S'il la possède, l'attaque du div est diminuée de 2. Le div meurt si son attaque est inférieure à 0 (n'oubliez pas la propriété *mort* !)

Il nous manque cependant encore quelques notions.

Il nous faut comparer des nombres :

si l' attaque du div est inférieure à 0:  
si l' attaque du div est supérieure à 4 :

Et pour changer un nombre :

augmenter l' attaque du div de 1;  
diminuer l'attaque du div de 2;

Enfin, pour ne pas afficher le même texte plusieurs fois à la suite et ne pas ennuyer le joueur, voici une petite astuce :

dire "[parmi]La lame s'élève dans les airs, légère comme une flamme, et retombe sur le div, le blessant gravement[ou]Vous fendez comme si vous étiez le vent, perçant votre ennemi et réduisant ses attaques à néant[ou]Vous vous battez bien, vous sentez votre avantage[stopnant].";

Cette structure est analogue à « [si condition]...[fin si] ». La première fois, « La lame s'élève dans les airs, légère comme une flamme, et retombe sur le div, le blessant gravement » sera affiché, puis ce sera « Vous fendez comme si vous étiez le vent, perçant votre ennemi et réduisant ses attaques à néant ». Après cela, le message « Vous vous

battez bien, vous sentez votre avantage » sera toujours montré. Les différents textes séparés par « [ou] » entre « [parmi] » et « [stopnant] » seront donc affichés de manière séquentielle.

Et voilà, nous sommes en mesure de terminer l'écriture de notre jeu :

Bataille est une scène.

Bataille begins when le joueur est dans l' Oasis et le div est dans l' Oasis.

Bataille ends when le div est mort.

When bataille begins:

dire "Le div force vers vous !".

[Nous avons ici des conditions imbriqués. Assurez-vous de bien comprendre ce qui se passe et quand.]

Every turn during Bataille:

si le joueur est dans l' Oasis:

si l' attaque du div est inférieure à 4:

dire "[parmi]Vous feriez bien de trouver un moyen de vaincre ce monstre ![ou]Il fait tournoyer son poing au-dessus de sa tête.[ou]Il pousse un grognement hideux.[stopnant]";

augmenter l' attaque du div de 1;

sinon:

dire "Il finit par vous attraper et vous dévorer !";

fin de l'histoire en disant "Vous êtes mort".

Instead of attacking le div:

si le joueur porte le cimenterre:

dire "[parmi]La lame s'élève dans les airs, légère comme une flamme, et retombe sur le div, le blessant gravement[ou]Vous fendez comme si vous étiez le vent, perçant votre ennemi et réduisant ses attaques à néant[ou]Vous vous battez bien, vous sentez votre avantage[stopnant].";

diminuer l' attaque du div de 2;

si l' attaque du div est inférieure à 0, maintenant le div est mort;

sinon:

maintenant l' attaque du div est 4; [accélère la fin du joueur]

dire "Vous n'irez pas loin avec vos petits poings...".

When bataille ends:

dire "Le monstre s'écroule lourdement, sans vie.";

dire "Votre cheval arrive enfin, mais après la bataille !";

maintenant le cheval est dans l' Oasis.

## Version finale de notre jeu

Voici pour résumer le code complet que nous venons d'écrire, avec commentaires additionnels pour vous permettre de bien comprendre :

"Sable" by IFiction-FR (in French)

Part 1 - Configurations et initialisations générales

Chapter 1 - Les extensions

Include Experimental French Features by Nathanael Marion.

[Pour faire fonctionner les conditions en français.]

Include (-  
language French

<control-structure-phrase> ::=

/a/ si ... est début |

/b/ si ... est |

/c/ si ... |

/c/ à moins que |

/d/ répéter ... |

/e/ tant que ... |

/f/ sinon |

/g/ sinon si ... |

/g/ sinon à moins que |

/h/ -- sinon |

/i/ -- ...

-) in the Preform grammar.

Chapter 2 - La propriété Mort

[À utiliser avec les monstres de notre histoire]

Une personne peut être morte. Une personne est rarement morte.

[« Rarement » signifie que les personnes ne seront pas mortes par défaut.]

[Cette règle empêche le joueur d'interagir avec une personne morte, en remplaçant le message par défaut par un laconique « Cette créature est morte, laissons-la en paix. »]  
Instead of doing something to une mort personne, dire "Cette créature est morte, laissons-la en paix."

## Chapter 3 - Les objets initialement hors jeu

[Si on ne précise pas où est le cheval, il ne sera à aucun endroit au début du jeu. Nous le ferons venir plus tard, à la fin.]

Le cheval est un animal.

La description est "Votre cheval est un splendide Akhal-Teke."

Comprendre "akhal" ou "akhal teke" comme le cheval.

Instead of climbing or entering le cheval:

dire "Vous remontez sur votre cheval, et retournez enfin à votre palais.";

fin définitive de l'histoire.

Le div est un animal.

La description est "Un démon sanguinaire à la face simiesque et aplatie qui ne vous fera pas de quartier."

Comprendre "monstre", "demon" ou "sanguinaire" comme le div.

[Le div possède un nombre que nous allons appeler « attaque », et qui servira à déterminer s'il est sur le point de mourir ou de gagner.]

Le div has a number called attaque. L'attaque du div est 1.

## Part 2 - Les objets et les lieux

### Chapter 1 - L'oasis

[Affichons notre petite introduction au tout début de la partie.]

When play begins, dire "Bienvenue, aventurier. Vous êtes le prince de Perse, et au cours d'une partie de chasse dans la forêt, votre cheval vous a désarçonné. Vous avez alors erré à sa recherche, jusqu'à atteindre une oasis."

L' Oasis (f) est un endroit. "C'était une belle oasis ensoleillée mais, maintenant, le soleil va déclinant... Il y a le désert tout autour d'ici, à l'exception d'une forêt à l'est ; au sud descend un chemin en direction d'une crypte. La forêt n'est pas sûre en fin de journée."

Le dromadaire est un animal dans l' Oasis.

Instead of climbing or entering le dromadaire, dire "Vous, un prince, n'oseriez certainement pas retourner à votre palais à dos de dromadaire ![si le cheval n'est pas dans l' Oasis]Mais où est donc votre cheval ?[fin si]".

[ce type de code est un moyen de faciliter un test de condition impliquant uniquement

du texte,  
nous aurions pu l'écrire autrement, mais cela aurait été plus long.]

[Le dromadaire tue le joueur si ce dernier tente de l'attaquer.]

Instead of attacking le dromadaire:

dire "Le dromadaire n'a pas apprécié, et il vous le rend bien, avec un coup de sabot bien placé.";

fin de l'histoire en disant "Vous êtes mort".

Le palmier est une chose fixée sur place dans l' Oasis.

La description est "Un palmier qui oscille au gré du vent."

Comprendre "arbre" comme le palmier.

Le chemin est une chose décorative dans l' Oasis.

Le printed name est "chemin vers la crypte".

Le sable est une toile de fond dans l' Oasis, dans le Désert et dans la Crypte.

[Pour changer l'article indéfini du sable.]

L' indefinite article du sable est "du".

Instead of taking le sable, dire "Il n'y a pas de technocrate ici pour essayer de lui vendre votre sable, et vous n'avez de toute façon pas de récipient pour l'emporter."

[L'action « searching » résulte de la commande « fouiller ».]

Instead of searching le sable, dire "Autant chercher une aiguille dans une botte de foin."

Le Désert est au nord de l' Oasis et au nord-ouest de l' Oasis et à l'ouest de l' Oasis.

[Permet d'expliquer pourquoi le joueur ne peut aller dans une direction qui ne mène à aucun endroit depuis le désert.]

Instead of going nulle part from le désert, dire "Sans cheval, vous ne pouvez risquer de vous aventurer dans le désert seul."

La Forêt est un endroit. La Forêt est à l'est de l' Oasis.

[Nous empêchons le joueur d'entrer dans la forêt en lui affichant un message.]

Instead of going to la forêt, dire "La nuit va bientôt tomber, et la jungle est la plus dangereuse à ce moment ; autant rester dans le désert, essayer de retrouver votre cheval ou explorer les environs."

## Chapter 2 - La crypte

[On ne peut pas écrire « porte de pierre », car il y aurait une confusion avec le verbe « porter ».

Nous allons donc l'appeler « portedepierre » dans notre code.]

La portedepierre est une porte ouverte déverrouillée.

La portedepierre est au nord de la Crypte et au sud de l' Oasis.

Comprendre "porte", "seuil" ou "crypte" comme la portedepierre.

[La propriété « printed name » correspond au nom qui sera affiché au joueur.

Utile pour que ça n'affiche pas « Vous pouvez voir une portedepierre ».]

Le printed name de la portedepierre est "porte de pierre".

[Permet d'autoriser le joueur à quitter la crypte lorsqu'il tape « sortir » plutôt que « aller au nord ».]

Instead of exiting quand l'emplacement est la crypte, essayer de going nord.

Instead of opening or closing la portedepierre, dire "La porte est trop lourde pour vous[si la portedepierre est ouverte], de plus elle est déjà ouverte[fin si]."

La Crypte est un endroit. "Un silence de mort règne entre ces murs oppressants, et le murmure enjoué du vent n'arrive pas jusqu'ici."

La momie est une femme mort dans la Crypte.

[Nous affichons un petit texte avant d'examiner la momie. Nous n'utilisons pas la description car elle ne s'affichera jamais, puisque toute action sur une personne morte est bloquée par la règle que nous avons définie plus tôt.]

Before examining la momie, dire "En temps normal, cela ne ferait pas aussi peur, mais il faut bien avouer qu'en ces lieux, l'évocation de la mort prend une autre dimension."

Le cimenterre est une chose dans la Crypte.

La description est "Un beau sabre à la lame courbe et damassée qui scintille de mille feux."

Comprendre "sabre", "epee", "lame", "shamshir", "courbe" ou "damas" comme le cimenterre.

[Nous remplaçons le message par défaut lorsque le joueur prend le cimenterre, sans pour autant l'empêcher de le prendre.]

After taking le cimenterre pour la première fois:

dire "Ce sabre légendaire, le Shamshir, vous avait toujours fasciné, et maintenant

vous pouvez enfin mettre la main dessus."

Les murs sont des choses décoratives dans la Crypte.

La description est "Les murs sont recouverts de carreaux en terre cuite qui représentent des dieux et des animaux fabuleux. Une partie est restée en pierre brute."

Comprendre "dieux", "animaux", "fabuleux", "mur", "interstice", "pierre" ou "pierres" comme les murs.

Instead of searching or touching les murs, dire "À un certain endroit, il y a un interstice plus marqué entre deux pierres."

Instead of inserting le cimenterre into les murs:

si la portedepierre est verrouillée: [alors la porte s'ouvre]

dire "Vous glissez votre cimenterre dans l'interstice entre les deux pierres, et la porte est de nouveau ouverte.";

maintenant la portedepierre est déverrouillée;

maintenant la portedepierre est ouverte;

sinon: [si elle n'est pas verrouillée, rien ne se passe]

dire "Il ne se passe rien de spécial."

[Nous ajoutons des synonymes à une action pour permettre plus de choix au joueur.]

Comprendre "planter [cimenterre] dans [quelque chose]" ou "glisser [cimenterre] dans [quelque chose]" comme inserting it into.

### Part 3 - Les scènes

#### Chapter 1 - Enfermé

[Nous créons la scène.]

Enfermé est une scène.

[La scène commence quand le joueur porte le cimenterre.]

Enfermé begins when le joueur porte le cimenterre.

[Elle se termine mal quand dix minutes se sont passées depuis son commencement (1 min = 1 tour).]

Enfermé ends sadly when the time since Enfermé began is 10 minutes.

[Elle se termine bien quand la porte de pierre est ouverte.]

Enfermé ends happily when la portedepierre est ouverte.

When Enfermé begins: [quand la scène commence]

dire "Au moment où vous prenez l'arme, la porte donnant vers l'extérieur se referme !";

maintenant la portedepierre est fermée;

maintenant la portedepierre est verrouillée.

[Une règle « Every turn » s'exécute à chaque tour.

Celle-ci en particulier s'exécute à chaque tour, mais seulement si Enfermé est en cours.]

Every turn during Enfermé:

dire "L'air commence à manquer...".

When Enfermé ends sadly: [quand la scène se termine mal]

dire "Vous ne pouvez plus respirer. Cette crypte sera votre tombeau.";

fin de l'histoire en disant "Vous êtes mort".

When Enfermé ends happily: [quand la scène se termine bien]

maintenant le div est dans l' Oasis.

## Chapter 2 - Bataille

Bataille est une scène.

Bataille begins when le joueur est dans l' Oasis et le div est dans l' Oasis.

Bataille ends when le div est mort.

When bataille begins:

dire "Le div force vers vous !".

[Faites bien attention aux conditions imbriquées !]

Every turn during Bataille:

si le joueur est dans l' Oasis: [On vérifie cela pour que le joueur puisse se réfugier dans la crypte à tout moment.]

si l' attaque du div est inférieure à 4:

[la structure « parmi » permet d'afficher des messages de manière séquentielle.]

dire "[parmi]Vous feriez bien de trouver un moyen de vaincre ce monstre ![ou]Il fait tournoyer son poing au-dessus de sa tête.[ou]Il pousse un grognement hideux.[stopnant]";

augmenter l' attaque du div de 1;

sinon:

dire "Il finit par vous attraper et vous dévorer !";  
fin de l'histoire en disant "Vous êtes mort".

Instead of attacking le div:

si le joueur porte le cimenterre:

dire "[parmi]La lame s'élève dans les airs, légère comme une flamme, et retombe sur le div, le blessant gravement[ou]Vous fendez comme si vous étiez le vent, perçant votre ennemi et réduisant ses attaques à néant[ou]Vous vous battez bien, vous sentez votre avantage[stoppant].";

diminuer l' attaque du div de 2;

si l' attaque du div est inférieure à 0, maintenant le div est mort;

sinon:

maintenant l' attaque du div est 4; [accélère la fin du joueur.]

dire "Vous n'irez pas loin avec vos petits poings...".

When bataille ends:

dire "Le monstre s'écroule lourdement, sans vie.";

dire "Votre cheval arrive enfin, mais après la bataille !";

maintenant le cheval est dans l' Oasis.

Part 4 - Tests

[Nous verrons quoi mettre dans cette section bientôt !]

Eh oui, nous avons écrit tout ça. Et pourtant, il est encore possible de faire beaucoup de choses : par exemple, il manque un plan d'eau dans cette oasis ! Vous pouvez vous en charger vous-même ? Vous pourrez par exemple traiter l'eau comme le sable, et rajouter en plus des exceptions si le joueur essaye de la boire ou d'y entrer.

Une fois que vous serez satisfait de votre fiction interactive, il sera temps de la faire tester.

# Montrons notre jeu au reste du monde

---

## Testons !

Il serait bête de diffuser un jeu qui ne fonctionne pas comme prévu. Testez-le bien, puis demandez à d'autres personnes d'essayer votre petite aventure !

### Les privilèges du démiurge

En tant qu'auteur, nous avons la possibilité de faire des choses qu'un joueur normal ne pourrait pas. En voici un aperçu.

### Test me !

Il est possible de tester le cheminement du joueur au travers de certaines commandes, pour vérifier rapidement que le jeu est finissable, ou pour atteindre instantanément certaines zones ou certains objets.

La procédure de test automatique implique de rajouter dans le code cette partie :

```
Test me with "aller au sud/prendre le cimenterre/fouiller les murs/mettre la lame dans l'interstice/sortir/attaquer le div/encore/encore/monter sur le cheval".
```

Ainsi, durant le jeu, en tapant « test me », chaque commande sera automatiquement tapée les unes à la suite des autres.

Nous pouvons également créer des batteries de tests pour tester différentes parties du jeu :

```
Test me with "test one/test two/test three".  
Test one with "aller au sud/prendre le cimenterre"  
Test two with "ouvrir la porte/frapper la porte"  
Test three with "toucher la momie/fouiller les murs"
```

Ces commandes de débogage ne seront bien entendu pas disponibles lorsque le jeu final sera sorti (voir plus bas).

*Test me* est très utile pour vérifier qu'un jeu peut se dérouler correctement d'un bout à l'autre, jusqu'à la victoire.

### Déplacer des objets

En jeu, l'auteur peut utiliser la commande « goner *objet* » se déplacer instantanément dans un endroit ou dans le même endroit qu'un objet. À l'inverse, « purloin *objet* » permet de mettre dans son inventaire n'importe quel objet que l'on veut tester durant la conception du jeu.

Dans un petit jeu comme le nôtre, cela n'est pas forcément très utile, mais cela peut l'être si, par exemple, on souhaite tester un objet sans avoir à faire à chaque fois toutes les étapes permettant de l'obtenir.

## Publier son jeu

Lorsque votre jeu est prêt, il vous suffit de cliquer sur le bouton « Release » en haut à gauche de la fenêtre, et un fichier de jeu sera créé dans le dossier « sable.materials », situé au même endroit que votre projet.

Si vous voulez rendre votre jeu plus attrayant, vous pouvez ajouter dans ce même dossier une image au format JPG ou PNG nommée « Cover.jpg » (ou « Cover.png »). Elle servira de couverture à votre fiction interactive ; le manuel recommande une taille de 960 x 960 pixels et vous devez également fournir une version plus petite de 120 x 120 pixels (nommée « Small Cover.jpg/png »). Il ne vous reste plus qu'à ajouter dans votre code la ligne :

```
Release along with cover art ("Description de la couverture").
```

Le texte entre parenthèses est utilisé par les malvoyants, afin qu'ils puissent savoir de quoi a l'air la couverture.

Ne piquez pas d'images sur Google ! Créez votre couverture vous-même, demandez à une connaissance de la faire, ou prenez des images dont la licence vous le permet !

Si tout se passe bien, vous vous retrouverez avec un fichier « sable.gblorb ». Ce fichier contient votre jeu, sa couverture et d'autres données importantes comme son numéro d'identification unique. Il existe deux moyens de l'ouvrir :

**Sur internet** : écrivez la ligne suivante dans votre source :

```
Release along with an interpreter.  
[OU si vous avez aussi une couverture]  
Release along with an interpreter and cover art ("Description de la couverture").
```

Lorsque vous cliquerez sur le bouton « Release », un mini site internet contenant tout ce qu'il faut pour jouer à votre fiction interactive en ligne sera créé dans « sable.materials ». Il vous suffira de placer ce site sur un serveur (par exemple, sur votre blog).

**Sur votre ordinateur** : vous aurez besoin d'un interpréteur. Il s'agit d'un logiciel qui va vous permettre de lire votre jeu, comme un lecteur vidéo vous permet de lire un fichier MP4.

Un bon interpréteur est le logiciel *Gargoyle*, qui fonctionne sous Windows, OS X et Linux :

<https://github.com/garglk/garglk/releases>

Mais si vos joueurs préfèrent ne rien avoir à installer, il reste la version en ligne.

Le format Glulx (extension `.ulx` ou `.gblorb`) est en fait interprété par une machine virtuelle (à l'instar d'un programme en java), et des interpréteurs pour cette machine existent pour presque toutes les plates-formes existantes (Windows, OS X, Linux, FreeBSD, Nintendo DS, Pocket PC, iOS, ou même Atari et Amiga). Le gros avantage, c'est qu'en plus de pouvoir être interprété dans un navigateur, c'est le même fichier qui pourra être redistribué sur tous les systèmes d'exploitation à la fois, nul besoin de recompiler une version différente pour chaque OS.

## Des ressources pour aller plus loin

---

La ressource la plus complète à ce jour est le manuel inclus avec Inform. Pour y accéder, il suffit de cliquer sur l'onglet « Documentation ». Malheureusement, il n'est disponible qu'en anglais, et ne couvrira donc pas la traduction de la syntaxe et de toutes les instructions en français (à terme, le manuel devrait être traduit). Pour avoir la liste des correspondances français-anglais des instructions, un fichier PDF est disponible à la même adresse que l'extension française :

<https://bitbucket.org/informfr/i7-french-language/src>

Pour obtenir de l'aide ou jouer aux fictions interactives que d'autres personnes ont réalisées, vous pouvez vous rendre sur le site de la communauté francophone de la fiction interactive :

<http://ifiction.free.fr/>

Et notamment sur son forum, où vous pourrez demander de l'aide ou présenter votre œuvre :

<http://ifiction.free.fr/taverne/index.php>

En anglais, il existe beaucoup plus de sites internet consacrés aux fictions interactives.

Le forum de la communauté anglophone :

<http://www.intfiction.org/forum/>

L'IFDB, une base de données contenant toutes les fictions interactives ainsi que des critiques :

<http://ifdb.tads.org/>

L'IFWiki :

[http://ifwiki.org/index.php/Main\\_Page](http://ifwiki.org/index.php/Main_Page)

Si vous souhaitez étendre les possibilités d'Inform, de nombreuses extensions sont disponibles. Cliquez sur l'onglet « Extensions » puis sur « Public Library » en haut à droite. Encore plus d'extensions sont disponibles à l'adresse suivante (ne téléchargez pas les extensions sur le site officiel d'Inform 7, elles ne sont pas à jour) :

<https://github.com/i7/extensions>

## Conclusion

---

Et voilà, nous avons terminé ce survol des bases d'Inform 7, et cet ouvrage aurait sans doute nécessité beaucoup plus de place pour développer toutes les fonctionnalités étonnantes de ce langage. Néanmoins, j'espère que cela pourra donner à certains l'envie d'en savoir plus, et d'entreprendre la rédaction de fictions interactives passionnantes !

Si jamais vous avez des remarques sur cet ouvrage, n'hésitez pas les communiquer sur le forum de la communauté francophone. Merci !