Faculté des sciences et techniques de Mohammedia

Filiére ILISI

ShankScript
Un Langage de Programmation

Auteur:
ELKAISSI Souhail
LAGHEZALI Mohammed Réda

Encadrant:
A. BEKKHOUCHA

April 18, 2017

Abstract

ShankScript est un langage de programmation multi-paradigme et multiplateformes , il est doté d'une gestion automatique de la mémoire ,il est ainsi similaire à des langages de programmations comme Python ,C,C+ ...

Le Langage ShankScript est placé sous une licence libre ,il peut fonctionner dans la plateforme windows et aussi dans les systèmes d'exploitations UNIX (Ubuntu ,Fedora ,Kali ...) . Il est conçu pour rependre au besoin des débutant de la programmation et pour augmenter la productivité des programmeurs en offrant des outils de haut niveau et d'un syntaxe simple à utiliser .

Notre but dans l'avenir est qu'il soit apprécié par les pédagogies d'apprentissage et on veut qu'il soit l'un des piliers de la programmation dans le monde de développement .

Contents

Chapter 1

Introduction à ShankScript

1.1 Qu'est ce que ShankScript ?

En informatique, un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations .

Les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par l'appareil informatique, et d'autre part d'indiquer comment sont effectuées les manipulations, selon quels algorithmes. Ils servent de moyens de communication par lesquels le programmeur communique avec l'ordinateur, mais aussi avec d'autres programmeurs ; les programmes étant d'ordinaire écrits, lus, compris et modifiés par une équipe de programmeurs .
Un langage de programmation est mis en œuvre par un traducteur automatique : compilateur ou interpréteur. Un compilateur est un programme informatique qui transforme dans un premier temps un code source écrit dans un langage de programmation donné en un code cible qui pourra être directement exécuté par un ordinateur, à savoir un programme en langage machine ou en code intermédiaire, tandis que l'interpréteur réalise cette traduction « à la volée ».

Les langages de programmation offrent différentes possibilités d'abstraction, et une notation proche de l'algèbre, permettant de décrire de manière concise et facile à saisir les opérations de manipulation de données et l'évolution du déroulement du programme en fonction des situations. La possibilité d'écriture abstraite libère l'esprit du programmeur d'un travail superflu, notamment de prise en compte des spécificités du matériel informatique, et lui permet ainsi de se concentrer sur des problèmes plus avancés.

Chaque langage de programmation supporte un ou plusieurs styles de programmation – paradigmes. Les notions propres au paradigme font partie du langage de programmation, permettant au programmeur d'exprimer dans le langage de programmation une solution qui a été imaginée selon ce paradigme.
Les premiers langages de programmation ont été créés dans les années 1950. De nombreux concepts de l'informatique ont été lancés par un langage, avant d'être améliorés et étendus dans les langages suivants. La plupart du temps la conception d'un langage de programmation a été fortement influencée par l'expérience acquise avec les langages précédents.

Dans nôtre cas , ShankSscript est un peu ceci et cela ,avec le manque du temps qu 'on a ,et pour cette première version le langage et paradigmes ,est il est interprété ;vous pouvez valider commande par commande comme dans le cas de python sinon vous avez la possibilité d'écrire dans un fichier Bash avec l'extension ".sks" et par la suite le compiler .

Chapter 2

Analyse de l'existant

Intro

2.1 Partie 1

Intro

2.1.1 Sous-partie 1

Bla

2.1.2 Sous-partie 2

Bla

Transition

2.2 Partie 2

Bla

Transition

2.3 Bilan récapitulatif

Voici un tableau (cf. fig. 2.1) récapitulatif de notre analyse de l'existant...

Solution	Critère 1	Critère 2	Critère 3	Critère 4
Solution 1(cf. ref. [?])	Oui	Oui	Oui	Oui
Solution 2(cf. ref. [?])	Oui	Oui	Oui	Non
Solution 3(cf. ref. [?])	Oui (sauf telle chose)	Non	Non	Oui
Solution 4(cf. ref. [?])	Oui	Non	Oui	Non
Solution 5(cf. ref. [?])	Oui (uniquement ceux-ci)	Non	Oui	Non

Figure 2.1: Tableau récapitulatif des solutions

Chapter 3

Analyse des besoins

Intro

3.1 Besoins fonctionnels

Après une analyse des besoins fonctionnels du projet, nous avons défini deux sous catégories. D'un côté, les besoins [...], de l'autre, les besoins [...].

3.1.1 Sous-partie 1

Bla

3.1.2 Sous-partie 2

Bla

3.2 Besoins non-fonctionnels

Comme précédemment, nous avons choisi de distinguer deux catégories pour les besoins non-fonctionnels. D'une part, nous avons les besoins non-fonctionnels pour les [...], et d'autre part ceux pour [...]. Nous avons aussi pris en compte les contraintes de développement, que nous détaillerons à la fin de cette partie.

3.2.1 Sous-partie 1

Bla

Aperçu du rendu souhaité :

Figure 3.1: Rendu attendu

3.2.2 Sous-partie 2

Bla

3.3 Développement

Intro

3.3.1 Tâches

Bla

Priorité	Nom	Raison
1	Tache 1	Doit être vérifié en premier car sinon [...]
2	Tache 2	On doit pouvoir [...]
3	Tache 3	Comme les principales fonctionnalités permettant de tester sont opérationnelles, nous pouvons passer à cette tâche.
4	Tache 4	Parce que [...]
5	Tache 5	La tache 5 fait partie des principales [...].
6	Tache 6	Dernière fonctionnalité essentielle à mettre en place.
7	Tache 7	Non-essentiel, mais apporterait un plus au projet.
8	Tache 8	Non-essentiel, mais apporterait un plus au projet.

Figure 3.2: Tableau récapitulatif des tâches

3.3.2 Tests

Bla

Fonctionnalité	Test
Fonction 1	Quand [...], vérifier [...].
	Et quand [...], vérifier [...].
Fonction 2	Vérifier [...].
Fonction 3	Vérifier [...].
Fonction 4	Avoir [...].
Fonction 5	Accéder à [...].
	Vérifier que [...].
Fonction 6	Accéder à [...].
	Et vérifier [...].
Fonction 7	Installer [...].
	Vérifier [...].
Fonction 8	Compter [...].

Figure 3.3: Tableau récapitulatif des tests

Chapter 4

Autre partie

Dans cette partie nous cherchons à décrire dans un premier temps [...], puis, c[...].

4.1 Partie 1

Intro

4.1.1 Sous-partie 1

Figure 4.1: autre partie image 1[1]

[1]Schéma d'après : *Auteur 1 & Propriétaire image*, LICENCE (cf. ref. [**?**])

4.1.2 Sous-partie 2

Figure 4.2: autre partie globale de notre quelque chose

Nous retrouvons ici, blabla[2] [...].

4.1.2.1 Sous-sous-partie 1

Le bla (cf. ref. [?]) est [...]:

- item1;
- item2;
- item3;
- item4;
- item5.

[2] Application bla - Interface blabla

4.1.2.2 Sous-sous-partie 2

4.1.2.2.1 Paragraphe 1 (agissant comme titre niveau 5)

Figure 4.3: Structure d'une autre chose[3]

Ce schéma représente bla.

4.1.2.2.2 Paragraphe 2

Bla

Sous-paragraphe 1

Bla

Figure 4.4: Diagramme de truc

[3]Schéma et explication d'après le wiki bla (cf. ref. [**?**])

Sous-paragraphe 2

Bla

Bla

Sous-paragraphe 3

Bla

4.1.2.3 Sous-sous-partie 3

Bla

4.2 Partie 2

Bla
Bla

4.2.1 Sous-partie 1

Bla

4.2.2 Sous-partie 2

Bla

Paragraphe 1 (n'apparaitra pas dans l'index) Bla

Paragraphe 2 Bla

Paragraphe 3 Bla

4.2.3 Sous-partie 3

Bla

[3] D'après le schéma disponible sur la documentfation officielle disponible sur le site blalbla

Chapter 5

Résultats

5.1 Partie 1

Intro

5.1.1 Sous-partie 1

Paragraphe 1 (n'apparaitra pas dans l'index) Bla

Paragraphe 2 Bla

Paragraphe 3 Bla

5.1.2 Sous-partie 2

Bla

5.1.3 Sous-partie 3

Bla

5.2 Partie 2

Intro

Sous-partie 1 ('apparaitra pas dans l'index)

Bla

Paragraphe 1 ('apparaitra pas dans l'index) Bla

Paragraphe 2 Bla

Paragraphe 3 Bla

Sous-partie 2

Bla

(a) Première image (b) Deuxième image (c) Troisième image

(d) Quatrième image (e) Cinquième image

Figure 5.1: Différents screenshots quelque chose, en gallerie

Chapter 6

Bilan

Intro / Rappel Contexte

Nous avons donc pu en tirer la problématique suivante :

Problématique du sujet

Bla
Bla

Bla

Bla
Bla

Bla
Bla

Bla
Bla

Bla

Annexes

Annexe 1

Intro

1 Partie 1

Bla

1.1 Sous-partie 1

Bla

1.2 Sous-partie 2

Bla

1.3 Sous-partie 3

Bla

2 Partie 2

Bla

2.1 Sous-partie 1

Bla

2.2 Sous-partie 2

Bla

2.3 Sous-partie 3

Bla

Annexe 2

Intro

Prérequis

Bla

- item1;
- item2;
- item3;
- item4.

Bla

1 Partie 1

Bla

1.1 Sous-parie 1

Bla

1.2 Sous-parie 2

Bla

2 Partie 2

Attention !
Texte d'avertissement

Bla

3 Partie 3

Bla

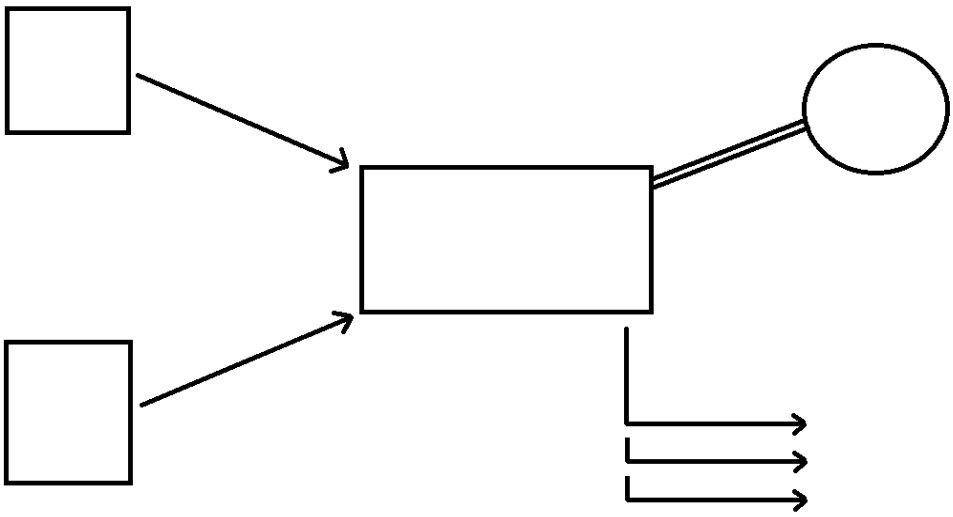

Figure 6.1: Presentation schema

Paragraphe 1

Bla

Paragraphe 2

Bla

Paragraphe 3

Bla