

Université de Montréal

Développement de jeux vidéo en Scheme

par
David St-Hilaire

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Décembre, 2009

© David St-Hilaire, 2009.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Développement de jeux vidéo en Scheme

présenté par:

David St-Hilaire

a été évalué par un jury composé des personnes suivantes:

Mostapha Aboulhamid
président-rapporteur

Marc Feeley
directeur de recherche

Yann-Gaël Guéhéneuc
membre du jury

Mémoire accepté le

RÉSUMÉ

Mots clés: Language de programmation fonctionnels, Scheme, jeux vidéo, programmation orientée objet.

ABSTRACT

Keywords: Functional programming languages, Scheme, video games, object oriented programming.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES FIGURES	viii
REMERCIEMENTS	ix
CHAPITRE 1 : INTRODUCTION	1
1.1 Contexte	1
1.1.1 jeu video -; \$\$	1
1.1.2 prog de jeu complexe : bcp années/hommes de travail	1
1.2 Motivation	1
1.2.1 Comment faciliter le dév ?	1
1.2.2 Prog haut niveau vs bas niveau	1
1.2.3 Scheme	1
1.3 Problématique	1
1.4 Méthodologie	2
1.4.1 Dev 1er jeu simple pour determiner les besoins pour Scheme	2
1.4.2 Augmenter Scheme pour repondre a ces besoins	2

1.4.3	Ecrire un nouveau jeu utilisant les techniques developpees	
	pour le 1er jeu	2
CHAPITRE 2 : DÉVELOPPEMENT DE JEUX VIDÉO		3
2.1	Contexte (intro ?)	3
2.2	Historique	4
2.2.1	4
2.3	Contraintes de programmation	4
2.3.1	fluidité	4
2.3.2	Modularité	4
CHAPITRE 3 : LE LANGAGE SCHEME		5
3.1	Programmation fonctionnelle	5
3.2	Continuations	5
3.3	Gestion mémoire automatique	5
CHAPITRE 4 : PROGRAMMATION ORIENTÉE OBJET		6
4.1	Description du langage	6
4.2	Implantation	6
CHAPITRE 5 : SYSTÈME DE COROUTINES		7
5.1	Description du langage	7
5.2	Implantation	7

CHAPITRE 6 : ÉVALUATION ET EXPÉRIENCES	8
6.1 Développement de « Space Invaders »	8
6.1.1 Objectifs	8
6.1.2 Développement itératif	8
6.2 Développement de « Lode Runner »	8
6.2.1 Objectifs	8
CHAPITRE 7 : TRAVAUX RELIÉS	9
7.1 Comparaison de langages	9
7.1.1 Lua	9
7.1.2 C++	9
7.2 Jeux en Lisp	9
7.2.1 QuantZ	9
7.2.2 Naughty Dogz	9
CHAPITRE 8 : CONCLUSION	10
BIBLIOGRAPHIE	11

LISTE DES FIGURES

REMERCIEMENTS

blablabla

CHAPITRE 1

INTRODUCTION

1.1 Contexte

1.1.1 jeu video -¿ \$\$

1.1.2 prog de jeu complexe : bcp années/hommes de travail

1.2 Motivation

1.2.1 Comment faciliter le dév ?

1.2.2 Prog haut niveau vs bas niveau

1.2.3 Scheme

1.3 Problématique

Ce mémoire de maîtrise vise à répondre à la problématique suivante :

Est-ce possible ou envisageable de concevoir et développer des jeux vidéo en Scheme ? Quels en sont les avantages et les inconvénients ?

1.4 Méthodologie

1.4.1 Dev 1er jeu simple pour determiner les besoins pour Scheme

1.4.2 Augmenter Scheme pour repondre a ces besoins

1.4.3 Ecrire un nouveau jeu utilisant les techniques developpees pour le 1er jeu

Afin de pouvoir répondre à ces questions, 2 jeux vidéo ont été développés en utilisant le langage de programmation Scheme. Le premier jeux a servi de plateforme d'exploration permettant d'élaborer une méthodologie qui semble efficace pour le développement de jeux. Afin d'obtenir une telle méthodologie, plusieurs itérations de développement ont été effectuées, chacune permettant d'explorer de nouveaux aspects sur la manière de résoudre les problématiques associées à la création de jeux, comme par exemple comment arriver à synchroniser des entités dans le jeux ou comment arriver à décrire efficacement un système de détection et de résolution de collisions.

Suite à l'écriture de ce premier jeu, un autre jeu plus complexe que le premier a été écrit afin de consolider les méthodologies précédemment utilisées.

CHAPITRE 2

DÉVELOPPEMENT DE JEUX VIDÉO

2.1 Contexte (intro ?)

Les jeux vidéo font parti d'un domaine de l'informatique en pleine effervescence grâce à une demande constante de nouveaux produits. Ces produits possèdent plusieurs caractéristiques de qualité auxquelles les consommateurs s'attendent à obtenir en effectuant l'acquisition d'un nouveau titre. Ces attentes du consommateur peuvent se traduire par les besoins suivant :

mettre ici une liste des attentes pour un jeu moderne

2.2 Historique

2.2.1

2.3 Contraintes de programmation

2.3.1 fluidité

2.3.1.1 Taux de rafraîchissement

2.3.1.2 Réponse quasi temps réelle

2.3.2 Modularité

2.3.2.1 Complexité des jeux ... gros logiciels

2.3.2.2 Développement itératif ... plusieurs modifications du système

CHAPITRE 3

LE LANGAGE SCHEME

[1]

3.1 Programmation fonctionnelle

3.2 Continuations

3.3 Gestion mémoire automatique

CHAPITRE 4

PROGRAMMATION ORIENTÉE OBJET

4.1 Description du langage

4.2 Implantation

CHAPITRE 5

SYSTÈME DE COROUTINES

5.1 Description du langage

5.2 Implantation

CHAPITRE 6

ÉVALUATION ET EXPÉRIENCES

6.1 Développement de « Space Invaders »

6.1.1 Objectifs

6.1.2 Développement itératif

6.1.2.1 Version initiale

6.1.2.2 Version orientée objet

6.1.2.3 Version avec système de co-routine

6.2 Développement de « Lode Runner »

6.2.1 Objectifs

CHAPITRE 7

TRAVAUX RELIÉS

7.1 Comparaison de langages

7.1.1 Lua

7.1.2 C++

7.2 Jeux en Lisp

7.2.1 QuantZ

7.2.2 Naughty Dogz

CHAPITRE 8

CONCLUSION

L'expérience d'écriture de ces jeux aura permis de faire le point sur les avantages et les inconvénients de l'utilisation d'un langage tel que Scheme pour le développement de jeu vidéo.

- + puissance d'expression / d'abstraction
- + langage dynamique (développement en-direct, malléabilités)
- + création de langages spécifiques au domaine
- Garbage Collection et sur-allocation
- Profilage plus difficile avec des LSD (pour Gambit-C et statprof)
- Balance entre abstraction et efficacité

BIBLIOGRAPHIE

- [1] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and interpretation of computer programs*. MIT Press, Cambridge, Mass., 1996.