# Guillaume Lay

Mémoire

# En quoi le cycle de vie d’un langage de programmation influe t’il sur les choix technologiques des projets et de leur gestion au sein d’une PME ?

2020-2021

Table des matières

[Guillaume Lay 1](#_Toc70066298)

[En quoi le cycle de vie d’un langage de programmation influe t’il sur les projets ? 1](#_Toc70066299)

[Remerciements 3](#_Toc70066300)

[Résumé 4](#_Toc70066301)

[Abstract 5](#_Toc70066302)

[Introduction 6](#_Toc70066303)

[Partie 1 : Le cycle de vie d’un langage 6](#_Toc70066304)

[Introduction 6](#_Toc70066305)

[Leek Wars 7](#_Toc70066306)

[Résumé 7](#_Toc70066307)

[Le LeekScript 7](#_Toc70066308)

[Sous-partie 1 : La création d’un langage de programmation 9](#_Toc70066309)

[Le contexte 9](#_Toc70066310)

[Pourquoi le point-virgule ? 9](#_Toc70066311)

[Sous-partie 2 : Comment faire évoluer son langage 9](#_Toc70066312)

[Sous-partie 3 : Comment anticiper le déclin 9](#_Toc70066313)

[Partie 2 : Comment choisir son langage 10](#_Toc70066314)

[Sous-partie 1 : Etude 10](#_Toc70066315)

[Sous-partie 2 : Prise de décision 10](#_Toc70066316)

[Sous-partie 3 : Les problèmes d’un retour en arrière (Kendo et le Lierre) 10](#_Toc70066317)

[Partie 3 : L’impact d’un changement sur un projet 10](#_Toc70066318)

[Sous-partie 1 : La formation des équipes 10](#_Toc70066319)

[Sous-partie 2 : La cohabitation des deux versions 10](#_Toc70066320)

[Sous-partie 3 : La transition client 10](#_Toc70066321)

[Conclusion 11](#_Toc70066322)

[Bibliographie 12](#_Toc70066323)

[Livre 1 12](#_Toc70066324)

[Livre 2 12](#_Toc70066325)

[Livre 3 12](#_Toc70066326)

[Déroulé méthodologique 13](#_Toc70066327)

[Présentation du thème : 13](#_Toc70066328)

[Liste des personnes rencontrées : 13](#_Toc70066329)

[Calendrier M2 13](#_Toc70066330)

## Remerciements

Je tiens à remercier ILTR bidule machin.

## Résumé

## Abstract

# Introduction

### Présentation de l’entreprise

### Présentation de GEODP

# Partie 1 : Le cycle de vie d’un langage

Parler d’Angular !!!

Solidity

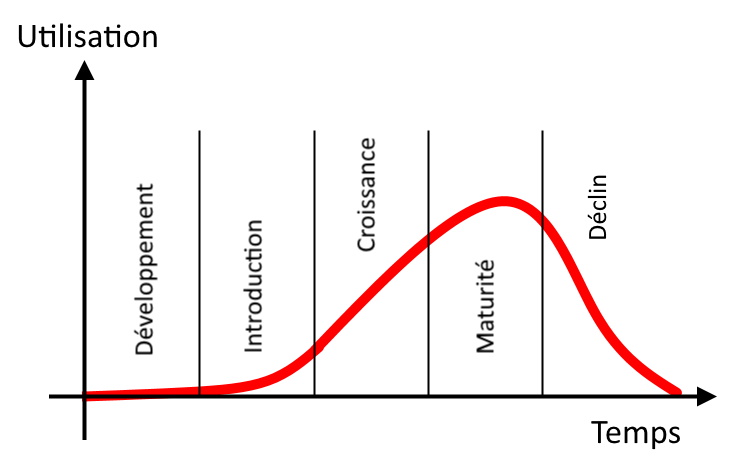
GoLang

GameMaker

## Introduction

Comme tout produit, chaque langage informatique suit un cycle de vie composé de cinq phases distinctes. Chacune de ces périodes possède ses propres questionnements et enjeux pour atteindre les objectifs de ventes ou de taux d’utilisation.

Ces cinq étapes sont les suivantes :

* Développement : Toute la partie conception et création du produit, l’établissement de ses règles et propriétés.
* Introduction : Le moment où le produit est révélé aux utilisateurs.
* Croissance : La période pendant laquelle le produit gagne en notoriété et se voit évoluer rapidement.
* Maturité : Lorsque le produit est installé et profite d’un taux d’utilisation maximal.
* Déclin : Le produit cesse d’être utilisé et tombe en désuétude.

Dans cette partie, nous détaillerons ces différentes étapes du cycle de vie d’un langage informatique en se concentrant sur les points les plus importants afin d’observer l’impact du temps sur une technologie informatique. Nous nous baserons sur des cas concrets et nous comparerons leur historique afin d’en tirer des enseignements.

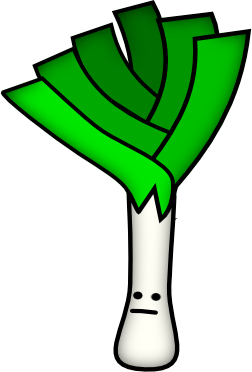


## Leek Wars

### Résumé

Leek Wars est un jeu vidéo français créé en 2013 par des étudiants en informatique. Son objectif est d’initier les joueurs à la conception d’intelligences artificielles et à la programmation en général. Le principe est que chaque joueur dispose d’un personnage à l’effigie d’un poireau et doit affronter les personnages adverses. Pour cela, les actions que le poireau devra effectuer lors de l’affrontement devront être définie au préalable par le biais d’un script écrit par le joueur.

Sachant que les combats se déroulent au format tour par tour, à la manière d’un jeu d’échecs, le comportement du poireau doit prévoir toutes les situations afin d’espérer vaincre son ennemi. Toute une panoplie d’armes et de sortilèges sont à la disposition des combattants. Tout cet attirail leur laisse le choix de la stratégie à aborder. Ils ont donc une totale liberté sur le comportement de leur poireau mais ils devront user de tactique et de fins calculs pour qu’ils aient le plus de chances de se sortir de toutes les situations.



Même si le projet est réservé à un public très ciblé, à savoir les curieux au sujet de l’informatique et les développeurs en quête d’expérimentation, le jeu a su gagner en notoriété au fil des ans et la communauté ne cesse de s’agrandir. La barre des cinquante mille inscrits a été dépassée et l’application est en constante évolution grâce à des mises à jour régulières apportant équilibrage et nouveau contenu.

C’est donc un jeu de programmation ayant un contexte et un objectif propre à lui-même. Le code source du projet étant en accès libre, nous allons pouvoir l’étudier en détail afin de mieux saisir les bases de ce qui fait un langage informatique.

### Le LeekScript

Afin de permettre aux joueurs de définir le comportement de leur poireau, un langage de programmation a été créé spécifiquement pour le jeu. Syntaxiquement proche du JavaScript, il a été pensé pour être totalement lié au système du jeu et pour être abordable pour les néophytes. Il est accompagné par un IDE intégré au jeu ainsi que son compilateur dédié.

Le LeekScript V1 est la version actuelle de ce langage et est celle qui est accessible aux joueurs utilisant l’application. Cependant, une nouvelle version, la V2, plus légère, plus performante et offrant de nouvelles possibilités est en cours de développement à l’heure de la rédaction de ce document. Elle doit notamment ajouter la programmation orientée objet (POO).

Suite à l’annonce relativement précoce de ce nouveau langage et de ses fonctionnalités, l’attente des joueurs a poussé le créateur du jeu à implémenter dans la V1 la POO. Cette évolution majeure de la technologie, causant une rétrocompatibilité difficile, a lancé la version 1.1 du LeekScript. La V2 est donc toujours en développement mais le projet entier étant open-source, il nous a été possible d’en prendre connaissance.

Etant donné que ce projet présente dans son histoire à la fois la naissance, l’évolution et la quasi-disparition d’un langage, il nous servira de cas d’étude pour le sujet principal de cette première partie. Un entretien avec son créateur Pierre Laupêtre est joint en annexe.

## Sous-partie 1 : La création d’un langage de programmation

### Qu’est-ce qu’un langage de programmation ?

Avant de commencer toute analyse, il est nécessaire de définir ce qu’est un langage de programmation. Par définition, il s’agit d’un moyen de communication entre l’homme et la machine. Il permet à un être humain d’interagir avec un outil électronique afin d’en tirer le comportement voulu.

Comme le dialecte initial de l’ordinateur ne se compose que de suites d’uns et de zéros, la question est uniquement affaire de traduction. Un langage de programmation est donc un dialecte qui fait office d’intermédiaire entre ces deux mondes. L’humain peut ainsi donner des tâches à faire à la machine.

Afin d’exécuter son œuvre, la machine dispose d’un élément fondamental : la mémoire. Tout composant électronique destiné à exécuter des calculs dispose d’un espace où il peut entreposer ses informations. C’est la ressource principale utilisée dans un calcul informatique.

### Pourquoi créer un langage ?

En règle générale, la création d’un nouveau langage de programmation survient pour combler un besoin technologique. Par exemple, le lancement d’une toute nouvelle machine électronique s’accompagne souvent de son langage dédié.

### Comment créer un langage de programmation ?

Etant donné que les langages de programmation sont créés par des humains et pour des humains, il n’est pas étonnant de constater la grande similitude entre la construction syntaxique d’une instruction à un ordinateur et une phrase destinée à un humain.

Créer un compilateur de langage,

Typage

### Pourquoi le point-virgule ?

C’est maintenant une convention dans la plupart des langages de programmation, le point-virgule termine une instruction. Le choix de ce caractère n’a rien d’un hasard. En effet, pour éviter toute ambigüité sémantique, il était nécessaire d’employer un caractère n’ayant aucun sens mathématique. Ensuite, il fallait qu’il remplisse la fonction de terminateur d’état. Là où la virgule, dans le langage courant, lie deux expressions sans forcément les séparer (Exemple : « Deux individus, l’un d’eux est roux »), le point-virgule a davantage le rôle de séparateur. Aussi, le point était inutilisable puisque servant déjà à désigner les décimaux.

Le choix du point-virgule comme terminateur d’instruction est un débat qui eut notamment lieu aux débuts de l’informatique. Cette décision était notamment défendue par les chercheurs Gannon et Horning. Aujourd’hui, cette règle s’est installée comme convention syntaxique. Même si elle n’est pas appliquée dans tous les langages, elle fait partie des habitudes des développeurs.

Compilateur

Compilé ou interpreté

Comment créer le compilateur

Esolangs, cette mine d’or

## Sous-partie 2 : Comment faire évoluer son langage

Etaler les mises à jour

Gérer la Retro compatibilité

## Sous-partie 3 : Comment anticiper le déclin

Angular

# Partie 2 : Comment choisir son langage

## Sous-partie 1 : Etude

Competences de l’équipe

Etude de marché

Etude du langage

Plateforme cible

Contexte du projet et cible

## Sous-partie 2 : Prise de décision

## Sous-partie 3 : Les problèmes d’un retour en arrière (Kendo et le Lierre)

Les raisons d’un retour arrière

Décrocher Kendo

Conséquences

# Partie 3 : Gestion de projet

Cas d’étude GEODP V1 et V2

La transition entre la V1 et la V2

Gérer la transition (Client/ Technique)

Faire le parallèle entre LeekWars et GEODP

## Sous-partie 1 : La formation des équipes

Bien communiquer sur les avantages de la nouvelle version et au BON MOMENT

Migrer le personnel de la V1 à la V2

Former techniquement

## Sous-partie 2 : La cohabitation des deux versions

Différence d’approche entre la gestion des versions

## Sous-partie 3 : La transition client

# Conclusion

# Bibliographie

## Livre 1

## Livre 2

## Livre 3

Déroulé méthodologique :

### Présentation du thème :

Durant tout mon parcours informatique, dès que je parlais de langages de programmation à des non-initiés, j’ai toujours eu droit à cette interrogation : « Mais pourquoi y en a-t-il autant ? ». Même si j’ai une très vague idée de cette raison, j’ai trouvé intéressant de répondre une bonne fois pour toutes à cette question. C’est un sujet qui me passionne et en l’abordant, il me permettra de faire un bilan complet de mes compétences acquises jusqu’à présent.

Ma thématique est assez vague ce qui me permettra d’aborder un large panel de sujets différents.

Liste des personnes rencontrées : 0

### Calendrier M2