

# Spécification technique

## **Game of Life**

Pour le client MOE

Réalisé par DIDIER Nino

# Identification du besoin du client et objectifs

Le client demande une application centré sur le jeu de la vie de Conway, mais avec quelques fonctionnalités parallèles.

Les demandes du clients sont :

Application qui permettrait de se connecter à un profil utilisateur afin d'avoir accès à des données personnelles et de récupérer les règles spécifiques au jeu de la vie à implémenter.

L'application doit présenter à jeu de la vie qui se joue à 0 joueur.

Un chat doit être intégré à l'application permettant aux utilisateurs de communiquer entre eux et de proposer des sets de règles du jeu de la vie.

# Technologies et compétences requises pour réaliser la demande du client :

Compte tenu des compétences de l'équipe de développement et des besoins de l'application, cette dernière sera développée sur .NET Core 6. Les besoins demandés peuvent être réalisés sous une application Windows Forms, Framework qui a fait ses preuves au cours de sa longue vie.

Pour la base de données, le choix de SQLite semble adapté afin d'avoir une base de données rapide et légère au vu de la base de données simple et petite.

La communication du chat utilisera des WebSockets comme protocole de communication, permettant l'intégrité des données lors des échanges de messages.

# Planification des tâches

Tâches à réaliser en fonction des besoins

Tâche	Répond au besoin	Technologie utilisée	Temps heure/homme estimé
Création d'une BDD	Profil utilisateur avec règles	SQLite/ORM C#	0.5 conception + 0.5 développement
IHM de l'application	Application	C# WinForms	0.5 conception + 1 développement
Algorithme du jeu de la vie	Jeu de la vie dans l'application	C#	0.25 conception + 0.25 développement
Tests unitaire pour le jeu de la vie et la partie utilisateur	Maintient futur de l'application par la mise en place de bonnes pratiques	C# Nunit	0.25 conception + 0.75 développement
Affichage du jeu de la vie	Jeu de la vie dans l'application	C#	0.25 conception + 0.75 développement
Mise en place de la partie serveur du chat	Chat intégré	C#, .NET WebSockets	0.5 conception + 1.5 développement
Mise en place de la partie client du chat	Chat intégré	C#, .NET WebSockets	0.5 conception + 1 développement
Documentation technique (commentaires code, readme.md)	Maintient futur de l'application par la mise en place de bonnes pratiques	Markdown, summaries C#	0.75
Pipelines CI/CD	Assurance qualité et gain de temps futur en automatisant les builds/tests	Jenkins (solution open source - gratuite)	1
Total			2.75 Conception + 7.5 développement

## Modèles de monétisation de l'application

### *Option 1 : Licensing*

Vente de clés de licences : Une clé publique est écrite en dure dans l'application, une différente pour chaque version de l'application. Des clés de License sont générés à partir de la clé privée, et sont vendus aux clients de l'application. La licence est créée à partir de la clé privée + l'adresse MAC de la machine cliente par exemple, pour que le client aie uniquement un accès illimité à l'application sur **cette** version et **cette** machine.

Cette solution peut être intéressante dans le cas où peu de mises à jours sont ajoutés à l'application dans le futur, et avec un retour sur l'investissement sur le court terme.

### *Option 2 : Abonnement*

Accès illimité à l'application, pour autant de machines que le client désire, avec un compte payant par mois. Permet d'accéder aux nouvelles mises à jours constamment.