

Cégep du Vieux Montréal

2022-09-02

Jeu de Daleks

Documentation du projet

Réalisé par

Théo Luquet & Justin Mayer

Présenté à

Godefroy Borduas

Dans le cadre du cours

C31 - GÉNIE LOGICIEL I : CONCEPTION ET GESTION | 420_IN - A22



TABLE DES MATIÈRES

Méthodes SCRUM

Analyse et modélisation

Implantation par sprints successifs

Analyse textuelle

Cas d'usage et Scénarios d'utilisation, ainsi que les rôles

Maquettes des interfaces graphiques

Menu principal

Menu des difficultés

Cours d'une partie

Fin de partie

Écran des scores

CRC

Modèle de données

Tables et/ou fichiers de données

Planning global

Sprint 1

MÉTHODES SCRUM

Analyse et modélisation

Obtention d'un mandat	Un mandat provient d'un client (externe, interne ou vous-même)
Analyse textuelle	Son but est d'obliger une lecture formelle du document
Identification des cas d'usage et des rôles	Ce que logiciel doit permettre de faire, du point de vue du client
Scénarios d'utilisation	La modélisation spécifiant séquentiellement dans le temps toutes les étapes pour réaliser un cas d'usage
Maquettes graphiques	Le design visuel des diverses pages-écrans (fenêtres) par lesquelles l'utilisateur manipulera le logiciel
Définition des classes et méthodes	L'identification des classes permettant d'instantier les objets chargés d'effectuer le traitement dans le programme
Modélisation des données	La définition des tables de données (ou autres modèles au besoin) dans lesquelles les informations devront être conservées
Planification globale <i>Incluant le découpage en sprints</i>	C'est le cadre temporel dans lequel on précise l'ordre de réalisation des classes, base de données, et autres composantes faisant partie du logiciel à créer

Implantation par sprints successifs

Un sprint est une période courte de production d'où doit sortir une partie du logiciel, le *livrable*.

Planification détaillée	À partir de la section du sprint courant dans la Planification globale, on attribue les tâches à des membres de l'équipe devant les réaliser. Chaque membre est responsable de préciser ses tâches en sous-tâches afin d'évaluer le temps requis à leur réalisation
SCRUMs quotidiens	Les scrums sont des réunions d'équipe quotidiennes où chaque membre doit indiquer les tâches complétées, les tâches à réaliser, et les problèmes rencontrés. Cette réunion doit se faire avec toute l'équipe afin d'assurer que tous comprennent comment le projet avance
Debriefing / Ajustement	Réunion de fin de sprint permettant d'évaluer comment le projet avance ou retarde, afin d'apporter les ajustements requis au prochain sprint
Livraison partielle	Le logiciel doit être utilisable pour faire partie d'une livraison

ANALYSE TEXTUELLE

L'analyse textuelle a pour but de forcer une lecture optimale du mandat du client. L'exercice consiste essentiellement à identifier les mots et expressions pertinents au logiciel à réaliser.

Comme la plupart des mandats sont écrits en langage ordinaire, on se concentre sur les noms des choses, les verbes qui dénotent des actions de ces choses, et sur les adjectifs qui les qualifient.

On ne répète pas inutilement les mots rencontrés plusieurs fois, on ne fait **pas de phrases**, les colonnes n'ont pas de lien l'une avec l'autre. On peut identifier des "expressions", tels "adresse de courriel", ou "envoyer courriel".

	Noms (objets)	Verbes (actions)	Adjectifs (attributs)
Explicites <i>(dans le texte)</i>	Docteur Daleks Tas de feraille Crédits cosmiques Aire de jeu Vagues Zapper Collision Cases (intersections) Téléporteur Liste Noms Console Progression "Run to end" "Run to end -1"	Déplacer Contrôler Volatiliser Téléporter Succombe Reviennent Disparaissent Attirer Échapper Afficher	Inexorablement Variable Aléatoire Facile Ordinaire Difficile Détruit Triée
Implicites <i>(requis par ce qui est explicite, mais jamais mentionné)</i>	Difficulté Tours Base de données Contrôles / input Menus	Conserver (noms & high scores) Réinitialiser (vagues / jeu entier) Actualiser (score, objets)	Individuel (joueur) Infini Diagonal
Supplémentaires <i>(suggestions pour le mandataire de choses possibles mais peut-être ignorées)</i>	Effets sonores Bosses / d'autres ennemis	Sauter (par dessus les Daleks) Ralentir (les Daleks)	Multijoueur Beau

CAS D’USAGE ET SCÉNARIOS D’UTILISATION, AINSI QUE LES RÔLES

Identification des cas d'usage, indication des rôles concernés par ceux-ci (à qui ils vont s'appliquer : président, gestionnaire, vendeur, commis, etc), et précision des détails d'implantation de la dynamique de chaque cas d'usage dans un scénario d'utilisation, à l'égard des commandes de l'humain et du traitement opéré par l'ordinateur en réponse.

Une seule entrée par ligne et jamais dans les deux colonnes sur la même ligne; du haut vers le bas.

C'est la **représentation temporelles des actions** (soit de l'humain, soit de l'ordinateur) qui compte.

Un cas d'usage peut référer à un autre cas d'usage; ce dernier doit alors expliquer en détail cette partie de la tâche (par exemple la tâche d'identification de l'utilisateur (login)).

Utiliser le IF pour des sections conditionnelles, et le FOR pour des boucles.

RÔLE <i>Profil d'utilisateur</i>	CAS D’USAGE <i>Tâche complète</i>	HUMAIN <i>Manipulations humaines dans l’interface</i>	MACHINE <i>Opérations effectuées par l’ordinateur</i>
Joueur	Lancer le logiciel	Démarre le logiciel	
			Affiche le menu principal : Lancer une partie (Option 1), Afficher les scores (Option 2), Quitter (Option 3)
		Input une option	
			Valide & enregistre l’input
			Afficher l’option choisie
Joueur	Lancer une partie (Option 1)		Affiche le choix de difficulté : Facile, Normal, Difficile
		Input une option	
			Valide & enregistre l’input
			Initialise la partie : crée aire de jeu, place les Daleks, place le Dr Who
			Affiche la partie et ses règles, demande un input du joueur

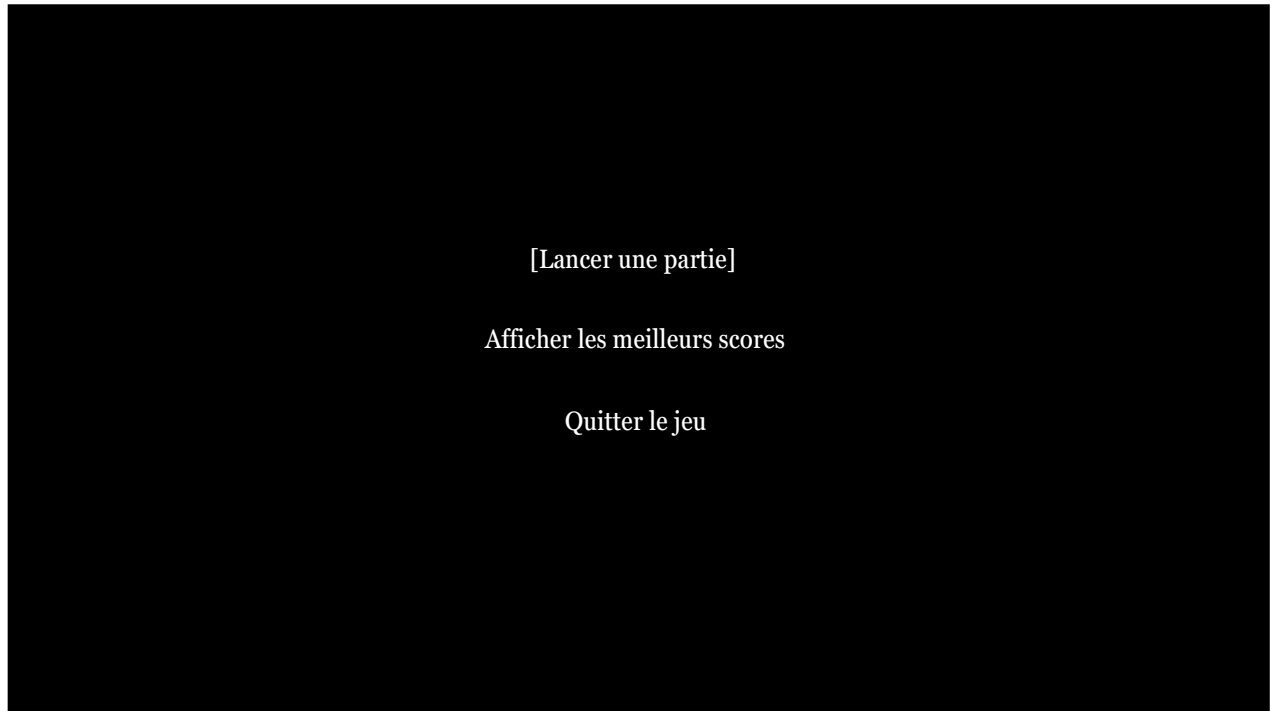
RÔLE <i>Profil d'utilisateur</i>	CAS D'USAGE <i>Tâche complète</i>	HUMAIN <i>Manipulations humaines dans l'interface</i>	MACHINE <i>Opérations effectuées par l'ordinateur</i>
Joueur	Jouer un tour	Input une commande de jeu	Déplacement/Attaque/Téléportation selon input Actualise aire de jeu : déplacer les Daleks, et vérifier les collisions Afficher le nouveau tour au joueur, ou afficher la fin de partie IF le Docteur est touché
Joueur	Fin de partie	Input son nom	Affiche son score et demande au joueur d'inscrire son nom Enregistre le score et le nom du joueur Affiche menu de fin : relancer une partie ou retourner au menu principal
		Input une option	Valide & enregistre l'input Affiche l'option choisie
Joueur	Afficher les meilleurs scores (Option 2)	Input pour retourner au menu principal	Affiche la liste des scores et noms de joueurs, et l'option de retourner au menu principal Valide & enregistre l'input Affiche l'option choisie
Joueur	Quitter (Option 3)	Input pour quitter le logiciel	Termine le processus du logiciel

MAQUETTES DES INTERFACES GRAPHIQUES

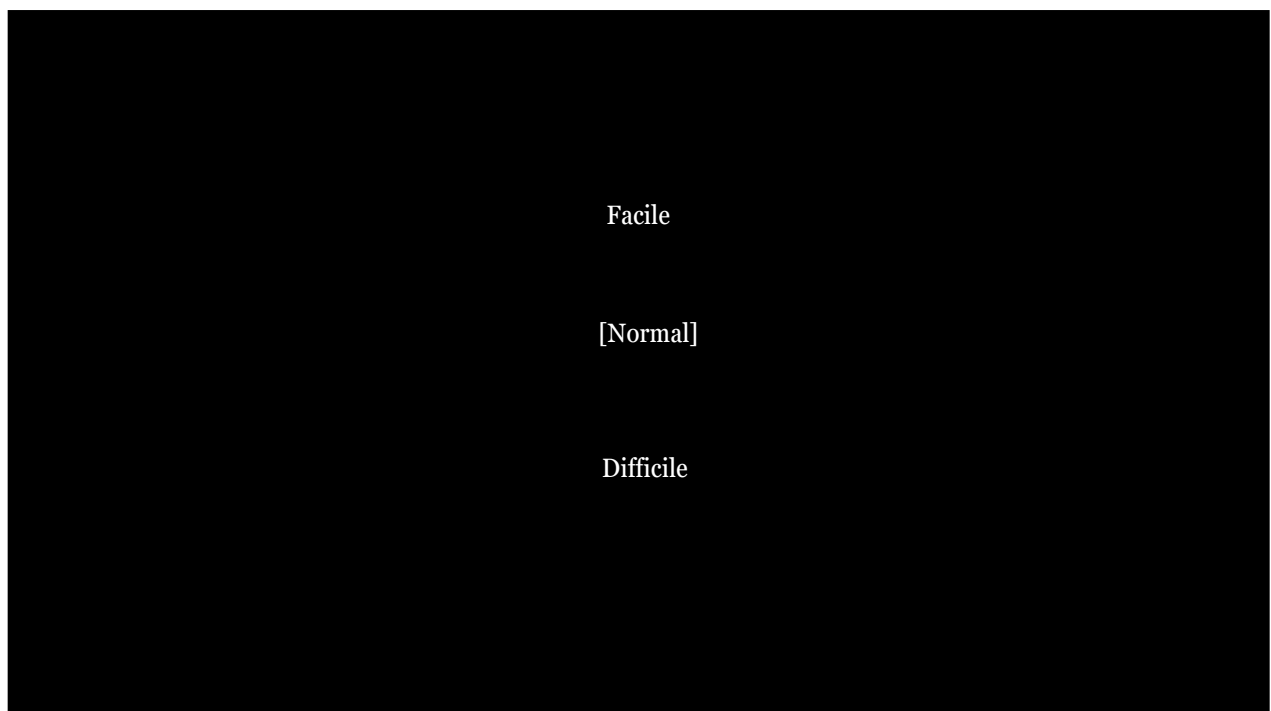
Maquettes graphiques représentant les pages-écrans du logiciel prévu. Cela permet d'identifier des ensembles fonctionnels de widgets offerts à l'utilisateur.

Regroupement conceptuel, plutôt qu'agencement esthétique, pour l'instant. On parle donc d'**ergonomie** du logiciel.

Menu principal



Menu des difficultés



Cours d'une partie

Daleks restants : 3
Crédits cosmiques : 488

Écrivez votre prochaine action : _____

HAUT, BAS, GAUCHE ou DROITE pour se déplacer d'une case dans la direction voulue

ZAPPEUR pour vaporiser les Daleks à proximité

TÉLÉPORTEUR pour se téléporter au loin

Fin de partie

TOTAL : 1666 Crédits Cosmiques

Entrez votre nom : J O J O ✓

[Quitter]

Écran des scores

NOM	SCORE	DATE
JOJO	1666	02/09/22
JAJA	488	04/09/22
JUJU	123	06/09/22
...

CRC

Identification des classes d'objets à programmer qui vont remplir les attentes du client, telles qu'identifiées par les tâches inscrites dans les scénarios d'utilisation provenant de l'analyse textuelle, selon les cas d'usage. Ces tâches deviennent ici des méthodes d'objet (*des fonctions*) : un objet est responsable des tâches qu'il peut effectuer. La collaboration est un lien vers des objets qui seront en relation avec le présent, et donc qui nous orientera vers son propriétaire pour paufiner le code de cette relation.

Classe : Contrôleur	Collaboration
Propriétaire : Justin	
Responsabilités : <ul style="list-style-type: none">• Récupérer l'input (au clavier) du joueur pour les menus ainsi que lors du jeu• Valider l'input• Enregistrer les scores	

Classe : Vue	Collaboration
Propriétaire : Théo	
Responsabilités : <ul style="list-style-type: none">• Afficher les menus• Afficher l'aire de jeu• Afficher les scores	

Classe : Jeu	Collaboration
Propriétaire : Justin / Théo	
Responsabilités : <ul style="list-style-type: none">• Créer l'aire de jeu• Placer les Daleks et le Docteur• Gérer les éléments de jeu selon leurs règles• Actualiser l'état de jeu	

Classe : Dalek	Collaboration
Propriétaire : Théo	
Responsabilités : <ul style="list-style-type: none"> • Se rapprocher du Docteur • Se changer en tas de ferraille lors d'une collision • Stocker position de l'élément • Stocker état de l'élément • Vérifier le déplacement 	<ul style="list-style-type: none"> • Jeu • Docteur

Classe : Docteur	Collaboration
Propriétaire : Justin	
Responsabilités : <ul style="list-style-type: none"> • Se déplacer • Se téléporter • Utiliser le zappeur • Stocker position de l'élément • Stocker état de l'élément • Vérifier le déplacement 	<ul style="list-style-type: none"> • Jeu • Daleks

MODÈLE DE DONNÉES

On établit notre diagramme de données. Fichiers plats, BD et SGBD, Big Data, etc. On indique comment nos données sont préservées.

Tables et/ou fichiers de données

Fichier CSV (comma-separated values) :

nom,	score,	date,
joueur1	10500	25/12/222
...

PLANNING GLOBAL

Cette section sert à préciser les scénarios d'utilisation qui seront développés lors des sprints : vous devez également les ordonner afin de produire les choses plus importantes le plus vite possible (afin notamment de réaligner le tir au besoin).

La liste des tâches doit représenter l'ensemble des éléments à programmer tel qu'indiqué dans les CRCs et la définition des modèles de données.

Placer les lignes de démarcation des sprints, par couleurs ou autre.

Membres de l'équipe : Justin & Théo

Fonctionnalité	Priorité	Temps prévu	Sprint visé
Créer le contrôleur	Haute	5min	1
Créer l'aire de jeu	Haute	5min	1
Créer les Daleks	Haute	5min	1
Placer les Daleks	Haute	5min	1
Créer le Docteur	Haute	5min	1
Placer le Docteur	Haute	5min	1
Déplacements Dalek	Haute	15min	1
Déplacement Docteur	Haute	10min	1
Collisions	Haute	15min	1
Inputs de jeu	Haute	20min	1
Fin de partie	Moyenne	10min	2
Gestion des vagues	Moyenne	30min	2
Difficultés	Moyenne	10min	2
Menu principal	Moyenne	20min	2
Menu difficultés	Moyenne	5min	2
Menu fin de partie	Moyenne	5min	2
Enregistrer scores (input & CSV)	Basse	20min	2
Ajouter le score au menu de fin de partie	Basse	5min	2
Menu de l'écran des scores	Basse	5min	2

SPRINT 1

Planification détaillée			Gestion des scrums								
Étapes (fonctionnalités haut niveau) <i>Litéralement faire un copier-coller à partir de la planification globale</i>	Sous-étapes <i>Découper l'étape en sous-tâches pour mieux indiquer le moment du livrable</i>	Responsable <i>Un membre de l'équipe</i>	Semaine 1			Semaine 2			Semaine 3		
			Prévu	Fait	Problème	Prévu	Fait	Problème	Prévu	Fait	Problème
Créer contrôleur	Coder & instancier la classe Contrôleur	Justin	30/08/22	30/08/22							
Créer l'aire de jeu	Coder la classe Jeu	Justin / Théo	30/08/22	30/08/22							
	Créer matrice de jeu	Justin	02/09/22	02/09/22							
	Coder la classe Vue	Théo	02/09/22	02/09/22							
	Lier la Vue au Contrôleur	Justin	02/09/22	02/09/22							
Créer les Daleks	Coder la classe Dalek	Théo	02/09/22	02/09/22							
	Initialiser avec des coordonnées aléatoires	Théo	02/09/22	02/09/22							
Placer les Daleks	Placer les Daleks dans l'aire de jeu en les instanciant dans la classe Jeu	Théo	02/09/22	02/09/22							
Créer le Docteur	Coder la classe Docteur	Justin	02/09/22	02/09/22							
	Initialiser avec des coordonnées au centre de l'aire de jeu	Justin	02/09/22	02/09/22							
	Méthodes du zappeur	Justin	02/09/22								
	Méthodes du téléporteur	Justin	02/09/22								
Placer le Docteur	Placer le Docteur dans l'aire de jeu en les instanciant dans la classe Jeu	Justin	02/09/22	02/09/22							
Déplacement Daleks	Méthode de déplacement des Daleks, selon la position du Docteur	Théo	02/09/22								
Déplacement Docteur	Méthode de déplacement du Docteur, 1 case à droite / gauche / en haut / en bas	Justin	02/09/22								
Collisions	Méthode de détection des collisions entre Daleks, résultant en tas de ferraille	Théo	02/09/22								
	Méthode de détection des collisions avec le Docteur, résultant en fin de partie	Justin	02/09/22								

Planification détaillée			Gestion des scrums								
Étapes (fonctionnalités haut niveau) <i>Litéralement faire un copier-coller à partir de la planification globale</i>	Sous-étapes <i>Découper l'étape en sous-tâches pour mieux indiquer le moment du livrable</i>	Responsable <i>Un membre de l'équipe</i>	Semaine 1			Semaine 2			Semaine 3		
			<i>Prévu</i>	<i>Fait</i>	<i>Problème</i>	<i>Prévu</i>	<i>Fait</i>	<i>Problème</i>	<i>Prévu</i>	<i>Fait</i>	<i>Problème</i>
Inputs de jeu	Définir les commandes valides	Justin / Théo	02/09/22								
	Validation des inputs	Justin / Théo	02/09/22								
	Modifier les méthodes du Docteur pour utiliser les inputs	Justin / Théo	02/09/22								