

*Fardel, Bastien
Rue des Fleurettes 22
1400, Yverdon-les-Bains
bastien.fardel@cpnv.ch*

SI-C4b

31.05.2022
4^{ère} année

Glossaire

.pdf

Le Portable Document Format, communément abrégé en PDF, est un langage de description de page présenté par la société Adobe Systems en 1992 et qui est devenu une norme ISO en 2008., 35

agile

En ingénierie logicielle, les pratiques agiles mettent en avant la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients., 10, 23

bool

En informatique, un booléen ou bool est un type de données avec deux valeurs possibles vrai ou faux., 28

boucle if

En informatique, une instruction conditionnelle, (aussi appelé expression conditionnelle), est une fonction d'un langage de programmation, qui effectue différents calculs ou actions, en fonction de l'évaluation d'une condition booléenne, à savoir vraie ou fausse., 19, 27

C#

C# (C sharp [si:.ʃɑ:p] en anglais britannique) est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET., 8

collider

Les composants de collision définissent la forme d'un objet dans le but d'effectuer des collisions physiques., 30

diagramme de flux

Un diagramme de flux représente généralement un processus, un système ou un algorithme informatique qu'on utilise pour documenter, planifier, affiner ou visualiser un workflow en plusieurs étapes., 19, 27, 28, 30

GitHub

GitHub (/ɡɪt ˈhʌb/, 8

IceScrum

IceScrum est un outil de gestion de projet basé sur les principes Agile et la méthodologie Scrum., 8, 10, 22

Librairies

En programmation informatique, une librairie désigne une collection de fichiers, de programmes, de routines, de scripts ou de fonctions qui peuvent être référencés dans le code de programmation., 33

material

Un matériau physique 2D est utilisé pour ajuster la friction et le rebond qui se produisent entre les objets physiques 2D lorsqu'ils entrent en collision., 31

nomenclature

Une nomenclature est un système de classification servant de référence dans le cadre d'une activité professionnelle, industrielle ou d'une discipline donnée., 26

POO

La programmation orientée objet (POO), ou programmation par objet, est un paradigme de programmation informatique., 8, 9

prefab

Les Prefabs dans Unity sont des GameObjects réutilisables préconfigurés que vous créez dans la scène et stockez dans le projet., 31

revue de sprint

La revue de sprint est l'une des plus importantes cérémonies de Scrum. L'objectif de la réunion de revue de sprint, comme on l'appelle généralement, est l'événement qui a lieu à la fin du sprint. Toute l'équipe se réunit pour passer en revue ce qui a été construit., 10

RigidBody2D

Un composant 2D Rigidbody place un objet sous le contrôle du moteur physique. De nombreux concepts familiers du composant Rigidbody standard sont transférés à Rigidbody 2D, 27

SetActive

Active/désactive l'objet, en fonction de la valeur true ou false donnée., 30

sprite

En infographie, un sprite est un bitmap bidimensionnel intégré dans une scène plus vaste, le plus souvent dans un jeu vidéo en 2D., 26, 28

story

La définition d'une story Agile est aussi simple que de rassembler une ou deux phrases sur un produit dans une certaine phase de développement. Pas n'importe quelles phrases, cependant, 17, 18, 22, 23

Suite Office

Microsoft Office, ou simplement Office, est une famille de logiciels clients, de logiciels serveurs et de services développés par Microsoft., 8

tests unitaires

En programmation informatique, le test unitaire (ou « T.U. », ou « U.T. » en anglais) est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »)., 23, 35, 36, 37

tilemap

Une tilemap est une carte dans laquelle chaque région est représentée par une seule tuile de même forme et de même taille., 30

TPI

Travail Pratique Individuel, 8, 9, 10, 16, 22, 24, 26, 32

trame

Une "image" est généralement une image unique dans la série d'images présentées rapidement à votre écran afin de donner l'illusion du mouvement dans votre jeu. Les termes font donc généralement référence au nombre d'images que votre jeu peut simuler et produire en une seconde., 27

UI

L'interface utilisateur est un dispositif matériel ou logiciel qui permet à un usager d'interagir avec un produit informatique., 31, 32

underscore

Le tiret bas ou souligné, parfois appelé sous-tiret, blanc souligné, caractère de soulignement ou underscore (de l'anglais), est un

signe typographique, représenté
par le symbole « _ », 26

Unity

Unity est un moteur de jeu
multiplateforme (smartphone,
ordinateur, consoles de jeux vidéo
et Web) développé par Unity
Technologies., 8, 9, 23, 26, 27, 28,
30, 31, 32, 40

user-friendly

Le terme user-friendly décrit un
dispositif matériel ou une interface
logicielle facile à utiliser., 9

Vector2

En règle Général, les vecteurs
stockent des données numérique

Pour un Vector2

x , y les coordonnées de l'objet
dans une scène 2D., 28

Visual Studio

Microsoft Visual Studio est une suite
de logiciels de développement
pour Windows et mac OS conçue
par Microsoft., 8, 32

Table des matières

Glossaire	2
Table des matières	5
Informations générales	6
Informations de contact	6
Période de réalisation	6
Horaire de travail :	7
Matériel à disposition	8
Connaissances requises.....	8
Liens utiles.....	8
Méthode de sauvegarde	8
Introduction au projet.....	9
Présentation du projet.....	9
Public cible	9
Points techniques évalués spécifiques au projet	9
Planification du projet et conceptualisation.....	10
Gestion du projet	10
Planification initiale	10
Planification détaillée	12
Stories	17
Diagramme de flux	19
Conception du niveau	20
Planification sur IceScrum	22
Stratégie de test	23
Difficultés potentielles	23
Réalisation du projet et comparaison	24
Comparaison de la planification initiale et détaillée	24
Nomenclature des éléments du TPI	26
Implémentation du mouvement du joueur.....	27
Implémentation du retour dans le temps.....	28
Implémentation de l'interaction avec les leviers.....	30
Implémentation du niveau et des objets dans celui-ci	30
Arborescence du projet sur Unity	32
Librairies du projet	33
Erreurs restantes	35
Liste des documents fournis	35
Conclusion.....	35
Objectifs atteints	35
Objectifs non atteints	35
Points positifs / négatifs	36
Difficulté particulières.....	36

Annexes.....	37
Résumé du rapport de TPI	37
Sources – références bibliographiques.....	38
Temps passé par tâches	40
Manuel d'installation	40
Archives du projet.....	40
Table des illustrations.....	41

Informations générales

Informations de contact

- Candidat :
 - Nom : FARDEL
 - Prénom : Bastien
 - Email : Bastien.FARDEL@cpnv.ch
 - Tél : +41 79 892 43 10
- Chef de projet :
 - Nom : VIRET
 - Prénom : Loïc
 - Email : loic.viret@cpnv.ch
 - Tél : +41 79 344 07 35
- Expert 1 :
 - Nom : ROY
 - Prénom : Alain
 - Email : alain.tpi@bluewin.ch
 - Tél : +41 79 444 01 54
- Expert 2 :
 - Nom : CHEVILLAT
 - Prénom : Jérôme
 - Email : jerome@chevillat.ch
 - Tél : +41 79 762 76 09

Période de réalisation

Le candidat commence son travail pratique individuel le 2 mai 2022 à 8h00 et doit rendre celui-ci le 31 mai 2022 à 10h45.

Horaire de travail :

Le candidat possède à sa disposition environ 90 heures pour effectuer son travail, voici son horaire :

	Lu 02.05	Ma 03.05	Me 04.05	Je 05.05	Ve 06.05
Tout le jour	Année 2021-2022 2ème semestre 2021-2022 4ème trimestre 2021-2022				
h / p					
08:00	TPI SC-C214 08:00 - 11:25 CARREL Xavier SI-MI4b	TPI SC-C214 08:00 - 10:35 HURNI Pascal SI-MI4b			TPI SC-C214 08:00 - 09:35 VIRET Loïc SI-MI4b
08:45			Economie d'entreprise 08:50 - 09:35 SC-C214 ZEN-RUFFINEN Xavier	TPI SC-C214 08:50 - 10:35 BENZONANA Pascal SI-MI4b	
08:50			Mathématiques 09:50 - 10:35 SC-C214 DELAPORTE Stéphane		TPI SC-C214 09:50 - 12:15 VIRET Loïc SI-MI4b
09:35			Langue et communication 10:40 - 12:15 SC-C214 THIERY Yvan	TPI SC-C214 10:40 - 12:15 BENZONANA Pascal SI-MI4b	
09:50		TPI SC-C214 10:40 - 12:15 VIRET Loïc SI-MI4b			
10:35					
10:40					
11:25					
11:30	Anglais SC-C214 11:30 - 12:15 CHARRERE Yann				
12:15					
12:40					
13:25					
13:30	TPI SC-C214 13:30 - 15:05 VIRET Loïc SI-MI4b	TPI SC-C214 13:30 - 14:15 CARREL Xavier SI-MI4b	Société SC-C214 13:30 - 15:05 THIERY Yvan	Sport SC-Ancien Stand 13:30 - 15:05 DAFFLON Marc SI-MI4b	TPI SC-C214 13:30 - 15:05 CARREL Xavier SI-MI4b
14:15		TPI SC-C214 14:20 - 16:05 CARREL Xavier SI-MI4b			
14:20					
15:05					
15:20	TPI SC-C214 15:20 - 16:55 VIRET Loïc SI-MI4b	TPI SC-C214 16:10 - 16:55 HURNI Pascal SI-MI4b		TPI SC-C214 15:20 - 16:55 BENZONANA Pascal SI-MI4b	Rattrapages - TE - Retenues 15:20 - 16:55 SC-C131 ZEN-RUFFINEN Xavier ROCHAT Claude SI-T2a SI-T1b SI-T1a SI-MI4b SI-MI4a SI-MI3b SI-MI3a SI-MI2b SI-MI2a SI-MI1b SI-MI1a SI-CA2a SI-CA1a SI-C4a SI-C3b SI-C3a SI-C2b SI-C2a SI-C1b SI-C1a SI-T2b
16:05					
16:10					
16:55					
17:00					

Matériel à disposition

Le candidat possède à sa disposition le matériel suivant :

- 1x Dell OptiPlex 7040 – W10 21H2
- Unity v.2021.3.1f1
- Visual Studio 2019
- Suite Office 365
- Unity Hub 3.1.2

Connaissances requises

Le candidat possède les connaissances suivantes :

- Formation de base du CPNV
- Connaissances en POO
- Maîtrise de Unity
- Maîtrise de C#

Liens utiles

- Répertoire du projet : [BFL-CPNV/TPI Projet B-lock BFL: Repository du projet B'lock pour le TPI \(github.com\)](#)
- Lien IceScrum pour la gestion : [iceScrum - B'lock](#)

Méthode de sauvegarde

Afin d'assurer la protection des données face à la suppression fortuite ou causée par tout autre accident, les données du projet sont sauvegardées et stockées des manières suivantes :

- Une clé USB de 120 GB contient la version la plus récente du travail effectué
 - Chaque fin de journée, ce travail est téléchargé sur un ordinateur sur lequel chaque version est stockée en cas de problème avec les autres moyens de sauvegarde
- GitHub permet le suivi du travail effectué en donnant accès aux multiples versions du code et permettant de stocker le travail en dehors du poste du candidat en cas de problème avec celui-ci

Introduction au projet

Présentation du projet

B'lock est un projet ayant pour but de concevoir, à l'aide de Unity, un jeu de plateforme 2D où le joueur incarnera un personnage pouvant se déplacer de gauche à droite, sauter et actionner des leviers permettant de faire réagir l'environnement en activant ou désactivant des ponts, plateformes et portes.

De plus le joueur sera capable de remonter le temps, une fonctionnalité qui n'impacte que le joueur et non son environnement, afin de résoudre des puzzles paraissant impossible à résoudre autrement.

Public cible

Le jeu a pour objectif de cibler un public d'un âge supérieur à 15 et plus, capable de réflexion et de lecture.

Points techniques évalués spécifiques au projet

Le travail du candidat est évalué par les critères généraux définis par la grille d'évaluation TPI.

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants :

1. Respect de conventions de nommage (à définir, documenter et argumenter) par le candidat
2. Architecture du code (SOC, POO, fonctions, ...)
3. Au moins un test unitaire est implémenté, à jour et pertinent
4. Qualité et lisibilité du code
5. Gestion des piles pour la fonction de remontage dans le temps
6. L'aspect « user-friendly » du jeu permettant au joueur une prise en main rapide
7. La réactivité du remontage dans le temps (début et fin de la capacité)

Planification du projet et conceptualisation

Gestion du projet

Le projet va être géré en suivant la méthode agile, séparant les tâches à effectuer dans des stories qui seront réparties par sprints.

Après discussion avec le chef de projet, une revue de sprint aura lieu chaque vendredi jusqu'au vendredi 20 mai 2022.

Étant donné l'absence de périodes de travail durant la 4^{ème} semaine de TPI, les revues de sprint restantes auront lieu les mardis matins.

Afin d'appuyer le choix de la méthode de gestion, IceScrum sera utilisé afin de gérer la progression du travail.

L'utilisation de la méthode agile sera bénéfique au projet actuel, permettant de vérifier le travail effectué auprès du chef de projet et permettant de s'assurer que les fonctionnalités telles qu'elles sont implémentées sont celles qui sont attendues pour ce projet.

Le résultat de ces revues de Sprint quotidiennes aura pour avantage de permettre au candidat de corriger les fonctionnalités et la vision du projet à chaque avancée dans celui-ci.

Planification initiale

Lors du premier jour de départ du travail pratique individuel du candidat, celui-ci a dû fournir une planification initiale du projet ayant pour but de répartir le travail à faire sur un axe temporel, afin d'illustrer le travail effectué et rendu aux experts lors du premier jour vous retrouverez celle-ci dans la page suivante.

La planification initiale a été créée dans l'optique de séparer le travail à effectuer parmi les "cinq" semaines à disposition, consacrant la première pour la planification et la conceptualisation du projet, la deuxième pour le début de l'implémentation des fonctionnalités de bases et finalement la troisième et les quatre derniers jours restants pour l'implémentation des fonctionnalités avancées telles que le retour dans le temps et les menus du jeu.

En ce qui concerne la répartition du temps entre les différentes tâches, celle-ci a été faite en se basant sur le temps qui fut nécessaire pour effectuer des tâches semblables lors du pré-TPI et ce afin de garder une certaine marge en dépendant de la difficulté des tâches et de leurs imprévus.

B'lock

CPNV
Bastien Fardel

Début du projet : lun, 2.5.2022
Semaine d'affichage : 1

Semaine d'affichage :			2 mai 2022							9 mai 2022							16 mai 2022							23 mai 2022							30 mai 2022						
TÂCHE	DÉBUT	FIN	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5
			l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d
Planification et Analyse du projet																																					
Planification initiale	2.5.22	2.5.22																																			
Création de l'environnement de travail	2.5.22	3.5.22																																			
Création des stories	3.5.22	3.5.22																																			
Conception des diagrammes de flux	3.5.22	5.5.22																																			
Conception de la maquette graphique du jeu	6.5.22	6.5.22																																			
Création des sprints, tâches et stories sur IceScrum	9.5.22	9.5.22																																			
Réalisation du projet																																					
Réalisation d'un niveau pour tester les fonctionnalités futurs et le personnage et ses contrôles basiques	9.5.22	13.5.22																																			
Réalisation des puzzles à levier	13.5.22	17.5.22																																			
Réalisation de la fonction de retour dans le temps et implémentation dans les puzzles	17.5.22	20.5.22																																			
Réalisation de 2 niveaux	20.5.22	23.5.22																																			
Réalisation d'un menu pause et d'un menu principal pour lancer le jeu et le quitter	23.5.22	30.5.22																																			
Réalisation de fonctionnalités supplémentaires si précédentes terminées	30.5.22	30.5.22																																			
Gestion du projet - Documentation et tests																																					
Mise à jour de la documentation journalière - Journal de travail + rapport quand pertinent	3.5.22	31.5.22																																			
Création des tests unitaires automatiques dans le code - au fur et à mesure	9.5.22	30.5.22																																			
Validation des tests d'acceptations et des stories - au fur et à mesure	9.5.22	31.5.22																																			
Sprint Review - tous les vendredi entre 08:00 - 09:00	2.5.22	20.5.22																																			

Planification détaillée

Semaine 1 – Consacrée à la planification pendant le premier jour et la conceptualisation pendant le reste de celle-ci.

Planification du TPI

Bastien Fardel
CPNV

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	
HORAIRE	02.05.2022	03.05.2022	04.05.2022	05.05.2022	06.05.2022	
08:00-08:45	Rencontre avec l'expert 1	Mise en place de l'environnement de travail	Aucune période TPI	Aucune période TPI	Revue du Sprint	
08:50-09:35		Conception des stories		Conception des diagrammes de flux		
09:35-09:50	Pause	Pause		Pause	Pause	
09:50-10:35	Planification du projet	Conception des stories		Conception des diagrammes de flux	Conception des diagrammes de flux	
10:40-11:25				Mise à jour du journal de travail		
11:30-12:15	Aucune période TPI					
12:15-13:30	Pause	Pause			Pause	
13:30-14:15	Planification du projet	Conception des stories		Aucune période TPI	Mise en place du projet sur IceScrum	
14:20-15:05		Mise à jour du Rapport			Mise à jour du Rapport, du journal de travail et envoi	
15:05-15:20	Pause	Pause				
15:20-16:05	Planification du projet	Conception des diagrammes de flux			Conception de la maquette graphique du jeu	Aucune période TPI
16:10-16:55		Mise à jour du journal de travail et envoi				

Heures à effectuer cette semaine : 22:30

Heures à disposition du candidat : 88:30

Figure 1 : Planification détaillée - Semaine 1

Semaine 2 – Consacrée à l'implémentation des fonctions de base, répartissant les mouvements du joueur et de la création d'un niveau de base pendant la première partie de la semaine et l'implémentation des leviers pour les puzzles dans la deuxième partie.

Planification du TPI

Bastien Fardel
CPNV

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
HORAIRE	09.05.2022	10.05.2022	11.05.2022	12.05.2022	13.05.2022
08:00-08:45	Réalisation d'un niveau de test	Ajout du joueur et de ses contrôles	Aucune période TPI	Aucune période TPI	Revue du Sprint
08:50-09:35				Ajout du joueur et de ses contrôles	
09:35-09:50	Pause	Pause		Pause	Pause
09:50-10:35	Réalisation d'un niveau de test	Ajout du joueur et de ses contrôles		Ajout du joueur et de ses contrôles	Ajout des leviers
10:40-11:25	Mise à jour du journal de travail			Ajout des leviers	
11:30-12:15	Aucune période TPI				
12:15-13:30	Pause	Pause			Pause
13:30-14:15	Ajout du joueur et de ses contrôles	Ajout du joueur et de ses contrôles		Aucune période TPI	Ajout des leviers
14:20-15:05					Mise à jour du journal de travail
15:05-15:20	Pause	Pause			
15:20-16:05	Ajout du joueur et de ses contrôles	Ajout du joueur et de ses contrôles		Ajout des leviers	Aucune période TPI
16:10-16:55		Mise à jour du journal de travail, du rapport et envoi		Mise à jour du journal de travail	

Heures à effectuer cette semaine : 22:30

Heures à disposition du candidat : 88:30

Figure 2 : Planification détaillée - Semaine 2

Semaine 3 – Consacrée à l’implémentation des fonctionnalités avancées, le retour dans le temps représente la majeure partie du travail à effectuer pendant la semaine, laissant place à un jour pour l’implémentation d’un niveau complet.

Planification du TPI

Bastien Fardel
CPNV

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
HORAIRE	16.05.2022	17.05.2022	18.05.2022	19.05.2022	20.05.2022
08:00-08:45	Ajout du retour dans le temps	Ajout du retour dans le temps	Aucune période TPI	Aucune période TPI	Revue du Sprint
08:50-09:35				Mise à jour du rapport de TPI	
09:35-09:50	Pause	Pause		Pause	Pause
09:50-10:35	Ajout du retour dans le temps	Ajout du retour dans le temps		Mise à jour du rapport de TPI	Ajout du niveau
10:40-11:25					
11:30-12:15	Aucune période TPI				
12:15-13:30	Pause	Pause			Pause
13:30-14:15	Ajout du retour dans le temps	Ajout du retour dans le temps		Aucune période TPI	Ajout du niveau
14:20-15:05					Mise à jour du journal de travail, du rapport et envoi
15:05-15:20	Pause	Pause			Aucune période TPI
15:20-16:05	Ajout du retour dans le temps	Visite de l'expert 2		Mise à jour du journal de travail	
16:10-16:55	Mise à jour du journal de travail	Mise à jour du journal de travail, du rapport et envoi			

Heures à effectuer cette semaine : 22:30

Heures à disposition du candidat : 88:30

Figure 3 : Planification détaillée - Semaine 3

Semaine 4 – Consacrée à l'ajout des dernières fonctionnalités du jeu, le menu principal et le menu de pause sont séparés entre les deux derniers jours, étant plus simples à implémenter dans le projet.

Planification du TPI

Bastien Fardel
CPNV

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
HORAIRE	23.05.2022	24.05.2022	25.05.2022	26.05.2022	27.05.2022
08:00-08:45	Ajout du menu principal	Revue du Sprint	Aucune période TPI	Aucune période TPI	Aucune période TPI
08:50-09:35					
09:35-09:50	Pause	Pause			
09:50-10:35	Ajout du menu principal	Ajout du menu pause			
10:40-11:25					
11:30-12:15	Aucune période TPI				
12:15-13:30	Pause	Pause			
13:30-14:15	Ajout du menu principal	Ajout du menu pause			
14:20-15:05					
15:05-15:20	Pause	Pause			
15:20-16:05	Ajout du menu principal	Mise à jour du journal de travail, du rapport et envoi			
16:10-16:55	Mise à jour du journal de travail				

Heures à effectuer cette semaine : 12:45

Heures à disposition du candidat : 88:30

Figure 4 : Planification détaillée - Semaine 4

Semaine 5 (finale) – Consacrée à l'ajout du menu de pause si celui-ci devait ne pas être terminé et à la documentation afin de finir celle-ci et de la peaufiner, c'est aussi la dernière semaine de TPI, le rendu du projet ayant lieu le mardi à 10h45.

Planification du TPI

Bastien Fardel
CPNV

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
HORAIRE	30.05.2022	31.05.2022	01.06.2022	02.06.2022	03.06.2022
08:00-08:45	Ajout du menu pause	Revue du Sprint	Aucune période TPI	Aucune période TPI	Aucune période TPI
08:50-09:35					
09:35-09:50	Pause				
09:50-10:35	Ajout du menu pause	Mise à jour du rapport TPI			
10:40-11:25		RENDU FINAL			
11:30-12:15	Aucune période TPI				
12:15-13:30	Pause				
13:30-14:15	Ajout du menu pause				
14:20-15:05	Mise à jour du rapport TPI				
15:05-15:20	Pause				
15:20-16:05	Mise à jour du rapport TPI				
16:10-16:55	Mise à jour du journal de travail				

Heures à effectuer cette semaine : 08:15

Heures à disposition du candidat : 88:30

Figure 5 : Planification détaillée - Semaine 5

Stories

Afin de définir les tâches qui seront à effectuer de manière plus détaillée, des stories sont créées pour mieux comprendre le travail à effectuer et estimer l'effort nécessaire pour leur complétion.

Chaque story possède ses propres tâches et définit le travail à effectuer pour accomplir celles-ci.

B'lock, projet TPI de Bastien Fardel

STORIES

Déplacement dans l'environnement	
En tant que	utilisateur
Je veux	me déplacer
Pour	explorer mon environnement
Effort	5
Description	Je veux pouvoir me déplacer de gauche à droite et pouvoir sauter pour explorer mon environnement et découvrir les puzzles.

Interaction avec l'environnement	
En tant que	utilisateur
Je veux	interagir avec mon environnement
Pour	résoudre les puzzles
Effort	5
Description	Je veux pouvoir interagir avec les leviers pour résoudre les puzzles en

Retour dans le temps	
En tant que	utilisateur
Je veux	inverser le temps
Pour	résoudre les puzzles
Effort	8
Description	Je veux pouvoir revenir en arrière sur mes mouvements afin de pouvoir résoudre des puzzles qui sont autrement impossible à résoudre.

Niveaux	
En tant que	développeur
Je veux	créer un niveau
Pour	que le joueur puisse y résoudre des puzzles
Effort	3
Description	Je veux créer un niveau qui permet à l'utilisateur de résoudre des puzzles en utilisant les fonctionnalités du jeu.

TÂCHES ASSOCIÉES

Implémenter le déplacement à gauche	
Description	Le joueur doit pouvoir se déplacer à gauche lorsqu'il appuie sur "A".

Implémenter les collisions avec l'environnement	
Description	Le joueur ne doit pas pouvoir traverser les murs ou les obstacles solides.

Implémenter l'interaction entre le joueur et les leviers	
Description	Le joueur doit pouvoir actionner les leviers en se trouvant proche de ceux-ci et en appuyant sur "E".

Implémenter l'interaction des leviers avec l'environnement	
Description	Les leviers, lorsqu'ils sont baissés ou levés, ont un impact sur certaines plateformes de l'environnement ou sur les portes de celui-ci.

Implémenter le retour dans le temps lorsqu'une touche est maintenue	
Description	Le joueur peut remonter le temps en maintenant "R", reproduisant son parcours sans incidence sur l'environnement, lorsqu'il relâche celle-ci, le temps reprends son cours.

Implémenter le test unitaire de vérification pour s'assurer de l'ajout à la pile	
Description	Vérifier que les mouvements du joueur sont correctement ajoutés à la pile

Implémenter un niveau	
Description	Créer un niveau qui contient des puzzles utilisant toutes les fonctionnalités du jeu, principalement le retour dans le temps.

Implémenter une transition de zone	
Description	Lorsque le joueur change de zone, la caméra change de zone

Implémenter le saut	
Description	Le joueur doit pouvoir se déplacer à droite lorsqu'il appuie sur "B".

Implémenter le saut	
Description	Le joueur doit pouvoir sauter lorsqu'il appuie sur la touche "W", mais ne doit pas pouvoir sauter s'il n'est pas au sol.

Implémenter le test unitaire de vérification pour s'assurer du retrait de la pile	
Description	Vérifier que les mouvements du joueur sont correctement retirés (lus) depuis la pile

Implémenter une explication des contrôles dans la première partie du niveau	
Description	Le joueur doit pouvoir facilement apprendre les contrôles du jeu en ayant un rappel visuel de ceux-ci dans la première partie du niveau

PaLETTE graphique	
En tant que	développeur
Je veux	créer une palette graphique
Pour	implémenter les graphismes dans le jeu
Effort	1
Description	Je veux créer une palette graphique en noir et blanc uniquement contenant les illustrations du jeu pour les implémenter dans le jeu.

Menu principal	
En tant que	utilisateur
Je veux	accéder au menu principal lorsque je lance le jeu
Pour	pour jouer au jeu et le quitter
Effort	2
Description	Je veux accéder au menu principal pour ne pas directement être lancé dans le jeu et quitter celui-ci si nécessaire.

Menu Pause	
En tant que	utilisateur
Je veux	mettre le jeu en pause
Pour	y revenir plus tard ou le quitter
Effort	3
Description	Je veux pouvoir mettre le jeu en pause pour faire quelque chose d'autre et y revenir lorsque je le souhaite, je veux aussi pouvoir quitter le jeu depuis celui-ci.

Créer le sprite Idle du joueur	
Description	Créer le sprite d'inactivité du joueur

Créer le sprite des murs	
Description	Créer le sprite qui sera utilisé pour délimiter la zone de jeu et les plateformes

Implémenter un menu principal d'accueil	
Description	Implémenter un menu principal qui comportera les boutons suivants : "Jouer" et "Quitter".

Implémenter la fonctionnalité du bouton "Jouer"	
Description	Lorsque le bouton "Jouer" est appuyé, le jeu doit charger le niveau, lançant ainsi le jeu.

Implémenter la fonction de pause du jeu	
Description	Lorsque le joueur appuie sur "P", le jeu est mis en pause, les vecteurs de vitesse du joueur doivent rester les mêmes

Implémenter la fonctionnalité du bouton "Reprendre"	
Description	Lorsque le bouton "Reprendre" est appuyé, le jeu n'est plus en pause et le menu n'est plus affiché.

Créer le sprite de marche du joueur	
Description	Créer le sprite de déplacement du joueur

Créer le sprite des leviers	
Description	Créer le sprite utilisé par les leviers

Implémenter la fonctionnalité du bouton "Quitter"	
Description	Lorsque le bouton "Quitter" est appuyé, le jeu doit se fermer.

Implémenter la fonctionnalité du bouton "Quitter"	
Description	Lorsque le bouton "Quitter" est appuyé, le jeu doit se fermer.

Implémenter le menu pause	
Description	Lorsque le jeu est mis en pause, un menu s'affiche avec les boutons "Reprendre" et "Quitter".

Figure 6 : Stories du projet B'lock

Le travail a été séparé en sept stories qui contiennent chacune deux à quatre tâches qui sont à accomplir pour que la story soit considérée comme étant terminée.

Ces stories séparent les fonctionnalités de mouvement, de retour dans le temps, de menu principal et de pause, le niveau ainsi que l'aspect graphique du projet.

L'effort à fournir total qui a été calculé est de vingt-sept, en sachant que vingt-sept heures sont à la disposition du candidat pour l'implémentation, il est donc acceptable de considérer que l'effort est l'équivalent du temps en heures qui est estimé pour l'implémentation des stories.

Diagramme de flux

Avant de programmer les fonctionnalités nécessaires pour le projet, il est important de commencer par les conceptualiser afin de pouvoir facilement convertir le diagramme de flux en code, possédant toute la logique que celui-ci devra suivre.

On peut voir sur les deux diagrammes de flux suivants les fonctionnalités du joueur et des leviers.

Le joueur doit pouvoir se déplacer de gauche à droite lorsque celui-ci appuie sur la touche appropriée et ce de même pour sauter, pour ce qui concerne le retour dans le temps, le joueur doit pouvoir revenir en arrière sur ses déplacements et ne doit pas pouvoir se déplacer lorsque celui-ci remonte le temps, il est donc important d'enregistrer chacun de ses mouvements lorsque celui-ci se déplace afin de pouvoir retracer le chemin parcouru par celui-ci de manière exacte et de bloquer l'accès au reste des fonctionnalités du joueur lorsque celui-ci remonte le temps.

Dans le deuxième diagramme de flux, il est question de permettre au joueur d'interagir avec les leviers uniquement lorsque celui-ci se trouve à proximité.

Une fois un levier actionné par le joueur, l'état de celui-ci doit être changé en fonction de sa valeur actuelle, si celui-ci est activé alors il doit être désactivé et les obstacles qui seraient connectés à ce levier doivent eux aussi changer d'état.

Lors de la création du diagramme de flux pour le contrôleur du joueur, il a été décidé de ne pas insérer les boucles if les unes dans les autres mais de les laisser à part afin de permettre une certaine réactivité aux contrôles, il est important que le joueur puisse se déplacer vers la gauche et la droite même si celui-ci saute déjà, de même pour l'inverse.

Le joueur, lorsqu'il remonte le temps, ne lis qu'une seule donnée à la fois afin de permettre une bonne réactivité de la fonctionnalité, ne laissant presque aucun délai entre le début de l'appui de la touche et le relâchement de celle-ci.

Les deux diagrammes de flux dont nous avons discuté ci-dessus sont disponibles en annexe à ce document afin d'offrir une meilleure lisibilité.

Conception du niveau

Afin de présenter au joueur un environnement dans lequel celui-ci pourra utiliser les fonctionnalités du jeu, il a été décidé de créer un seul niveau qui est séparé en trois zones distinctes.

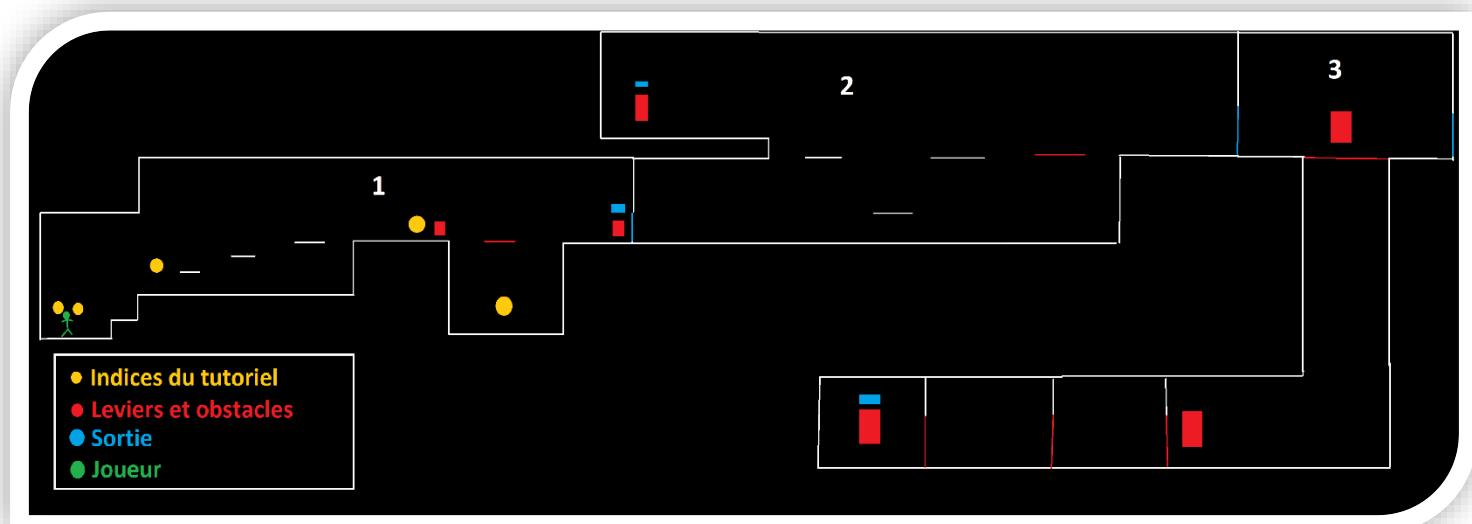


Figure 7 : Maquette Conceptuelle du niveau

Chaque zone met le joueur à l'épreuve pour résoudre un puzzle qui nécessite l'utilisation du retour dans le temps, la première zone servant de tutoriel.

La première zone est destinée à servir d'espace de découverte des commandes pour le joueur, celui-ci sera accompagné par des indices lui expliquant les commandes afin que celui-ci puisse facilement comprendre comment se déplacer et sauter.

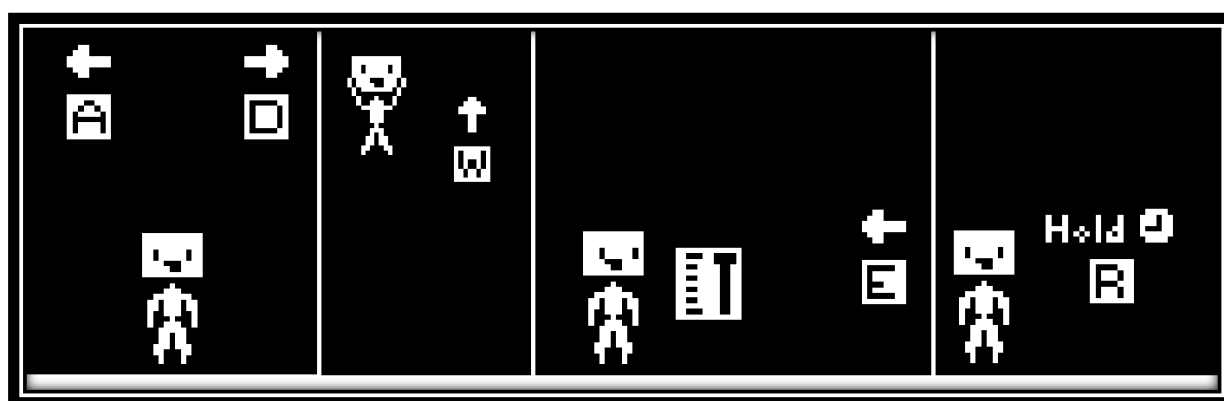


Figure 8 : Indices des commandes pour le joueur

Ensuite le joueur découvrira les leviers lui permettant de modifier son environnement, n'ayant d'autre choix que de descendre dans le trou nouvellement ouvert, le joueur se retrouvera coincé et sera accueilli par un indice lui expliquant comment remonter le temps pour sortir du trou.

Les deux autres zones permettent au joueur de mettre à l'épreuve leur contrôle des commandes et les plateformes sont disposées de manière à rendre les sauts compliqués, offrant l'occasion au joueur de faire usage du retour dans le temps lorsque celui-ci tombe afin de gagner du temps.

La deuxième zone semble simple en apparence mais nécessite au joueur de faire usage du retour dans le temps pour résoudre le puzzle de la zone.



Figure 9 : La sortie est fermée mais la plateforme est activée

Lorsque le joueur ouvre la sortie, la plateforme qui permet à celui-ci de l'atteindre disparaît, le joueur doit donc trouver un moyen d'atteindre la sortie malgré l'absence de la plateforme.

La troisième zone démontre au joueur que celui-ci peut traverser les obstacles grâce à la fonction de retour dans le temps, lui permettant de sortir de la cage dans laquelle celui-ci se retrouve piégé lorsqu'il ouvre la sortie.

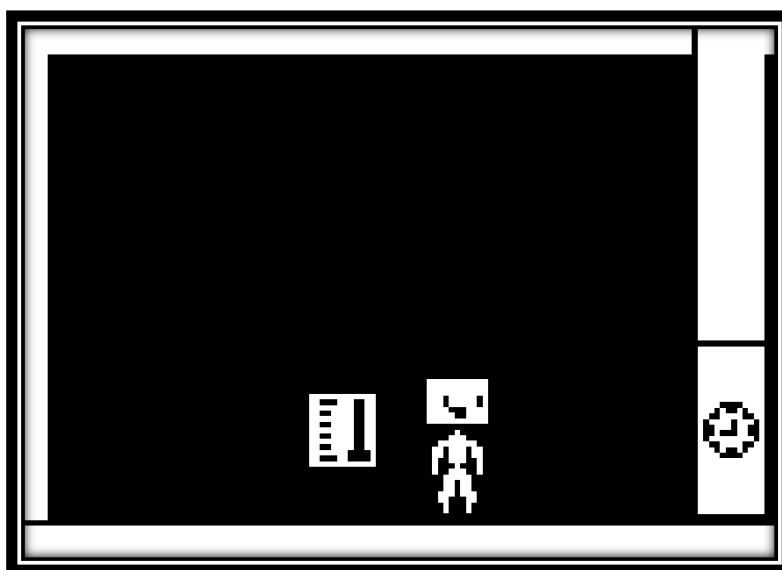


Figure 10 : Le joueur est enfermé dans une salle

Planification sur IceScrum

Les stories ont été créées sur IceScrum, afin de pouvoir gérer celles-ci, 3 sprints ont été créés pour finalement produire une release finale qui sera présentée pour le TPI.



Figure 11 : Stories sur IceScrum

SPRINT 1 :

« Ajout des fonctionnalités de base »

- Déplacement dans l'environnement
- Interaction avec l'environnement
- Palette graphique

SPRINT 2 :

« Ajout des fonctionnalités avancées et du niveau »

- Retour dans le temps
- Niveaux

SPRINT 3 :

« Ajout des menus »

- Menu principal
- Menu Pause

Chaque Sprint possède une durée d'une semaine, permettant d'effectuer une revue de Sprint en fin de semaine.

Pendant chaque revue de Sprint, les tests d'acceptation des stories seront vérifiés par le chef de projet afin de confirmer la fermeture des stories accomplies et de permettre le lancement du prochain Sprint, déplaçant les éventuelles stories non terminées dans le prochain Sprint.

Afin de considérer une story comme accomplie, les règles suivantes doivent être respectées :

- Les tâches de la story doivent être **Done**
- Les tests d'acceptation de la story doivent être passé en **Success**

Si la story n'est pas accomplie, celle-ci sera déplacée au prochain Sprint (*Shift to next Sprint*).

Stratégie de test

Tests unitaires

Afin de vérifier le bon fonctionnement de la fonctionnalité de retour dans le temps, des tests unitaires seront effectués pour assurer le bon fonctionnement de la pile utilisée pour gérer les données de retour dans le temps.

De plus ces tests unitaires sont demandés par le projet dans l'un des huit points techniques.

Tests d'acceptation

En raison de la gestion agile de mon projet, chaque story possèdera ses propres tests d'acceptation permettant de confirmer l'accomplissement de celle-ci.

Ceux-ci couvrent les fonctionnalités attendues de la story.

Difficultés potentielles

Retour dans le temps

La fonctionnalité de retour dans le temps n'est pas quelque chose d'inconnu pour moi, cependant n'ayant jusqu'à maintenant jamais fait usage d'une telle fonctionnalité dans un environnement faisant usage du système de physique d'Unity, je suppose rencontrer quelques imprévus pendant la création de cette fonctionnalité.

Cependant, je suis convaincu de pouvoir réussir à implémenter une telle fonctionnalité dans ce projet.

Documentation

La documentation de mon projet dans sa partie de planification et de conceptualisation n'est pas ce qui m'effraie le plus, l'implémentation risque d'être compliquée pour moi, n'ayant pas vraiment d'expérience sur la documentation de code en dehors des commentaires de ceux-ci.

Tests unitaires

Cela fait longtemps depuis la dernière fois où j'ai dû créer des tests unitaires, étant donné que je ne me souviens pas entièrement de comment ceux-ci fonctionnent et sont créés, il se peut que je rencontre des difficultés dans leur création.

Réalisation du projet et comparaison

Comparaison de la planification initiale et détaillée

Vous retrouverez ci-dessous une comparaison entre la planification initiale et détaillée par rapport au déroulement réel du projet, cette comparaison a pour objectif de me permettre de me rendre compte de la différence entre le temps prévu pour les tâches et le temps qui a réellement été nécessaire, ceci pour me permettre de mieux évaluer le temps nécessaire pour ce genre de travail dans le futur et surtout pour avoir une idée de la fiabilité de ma planification de base.

En se référant à la planification initiale disponible à la page onze (11) et de la planification détaillée aux pages douze (12) à seize (16), et au journal de travail à la page disponible en annexe, on remarque quelques changements, certains sont moins prononcés que d'autres.

En général, la planification initiale n'était pas si loin de la réalité de la chose. La première semaine a été respectée dans la majorité, celle-ci ayant été entièrement dédiée à la planification et la conceptualisation du travail futur.

Cependant, l'implémentation a pris de l'avance pendant la deuxième semaine, alors que la création des contrôles de base du joueur n'ont pris que deux (2) jours avant d'être terminés alors qu'il était prévu de prendre la semaine entière pour implémenter ceux-ci.

Peut-être avais-je sous-estimé mes capacités suite au fiasco de mon pré-TPI, il y avait sûrement eu faute de jugement sur le temps nécessaire pour l'implémentation des mouvements du joueur mais cela ne s'est pas avéré être un point négatif pour le projet.

Ce gain de temps a permis le début de l'implémentation de l'interaction avec les leviers et les puzzles deux (2) jours avant le début planifié pour celle-ci, et son implémentation complète le jour même où celle-ci devait être commencée.

Avec un gain de temps pareil, le début de l'implémentation du retour dans le temps a pu voir le jour le lundi 16 mai 2022 et se terminer le mardi 17 mai 2022, le jour où son implémentation aurait dû commencer.

Malgré le déraillement par rapport à la planification initiale, on remarque que cela n'est qu'une question d'un (1) ou deux (2) jours au plus, la planification étant donc plutôt précise à quelques jours près.

On remarque aussi quelque chose de différent entre la planification initiale et le travail réellement accompli, il était question d'implémenter deux (2) niveaux mais après discussion avec le chef de projet une idée est venue remplacer l'ancienne, l'ajout d'un niveau séparé en trois (3) zones.

Son implémentation a débuté à la date planifiée mais s'est terminé le jour même au lieu du jour suivant.

Dernièrement, l'implémentation du menu principal et du menu pause a commencé le 23 mai 2022 à la date planifiée mais s'est terminé le même jour au lieu du 30 mai 2022.

En ce qui concerne la documentation, le journal de travail a été mis à jour régulièrement contrairement au rapport qui n'a pas été mis à jour de manière journalière.

Les tests d'acceptations ont été vérifiés à chaque revue de Sprint avec le chef de projet entre huit heures quinze et neuf heures.

Comme remarqué avant, la planification en elle-même n'a pas été parfaite mais n'a pas non plus été entièrement erronée, les tâches planifiées ayant lieu ou se terminant un (1) jour en avance au plus.

Pour ce qui est de la planification détaillée, on remarque comme nous l'avons expliqué avant que certaines tâches possèdent de l'avance par rapport à la planification et le changement le plus visible est l'utilisation de la journée entière du jeudi 19 mai 2022 pour la mise à jour de la documentation.

Une chose importante à noter est la perte de la matinée complète le lundi 30 mai 2022 suite au placement de l'examen d'ECG (culture générale) pendant celle-ci.

Nomenclature des éléments du TPI

Pour la nomenclature de mon projet j'ai décidé de faire en sorte d'avoir des noms en anglais étant donné la nature du code qui est écrit en anglais, les fonctions, variables, objets, dossiers et fichiers possèdent des noms spécifiques qui permettent de s'assurer que ceux-ci soit faciles à différencier et à retrouver.

Les règles de cette nomenclature sont les suivantes :

- Les fichiers en rapport avec la documentation à rendre doivent suivre le format suivant : **Nom_Prénom_Titre_Version_TPI**
- Les dossiers du projet doivent être nommés en anglais avec des noms succincts qui permettent de facilement comprendre ce qu'ils contiennent en généralité, exemples : **Materials, Artworks, Scripts, Scenes...**
- Les fichiers de script doivent être nommés en anglais en utilisant une majuscule pour la première lettre et des minuscules ensuite, les espaces doivent être remplacés par le caractère *underscore*, exemples : **Object_controller, Player_controller, Lever_controller...**
- Les variables doivent être composées de minuscules uniquement et doivent posséder un nom qui permet de comprendre ce qu'elle représente, exemple : **is_grounded, player_sprite_renderer, feet_position, connected_objects...**
- Les constantes doivent être écrites en majuscules et ne doivent pas comporter d'espace, exemple : **REWINDINDEX, MOVEMENTSPEED, JUMPFORCE, CHECKRADIUS...**
- Les objets dans Unity doivent être nommés en anglais et doivent posséder une majuscule pour chaque première lettre d'un mot, les espaces doivent être remplacés par le caractère *underscore*, exemples : **Zone1_Obstacle, Zone1_Exit_Lever, Zone1_Exit, Player, Zone1_Exit_Trigger...**
- Les fonctions doivent être nommées en anglais et ne doivent pas comporter d'espaces, la première lettre de chaque mot doit être en majuscule, exemple : **MovePlayer, GroundCheck, InteractWithLever, ResumeGame, QuitGame...**

Implémentation du mouvement du joueur

Il avait été imaginé, dans le diagramme de flux du code, d'implémenter les mouvements à l'aide de boucles **if** testant si une touche spécifique était appuyée pour déplacer le joueur, cependant cette mise en œuvre n'est pas optimisée et produit beaucoup de répétition de code pour le déplacement, c'est pour cela que j'ai fait usage des librairies offertes par Unity.

Unity permet de récupérer une valeur représentant l'axe horizontal qui est obtenu lorsque le joueur appuie sur les touches **A**, **D** et **les flèches directionnelles** du clavier, ceci permet ainsi de récupérer la direction dans laquelle le joueur se dirige.

Grâce à cela il ne reste plus qu'à appliquer cette valeur, multipliée par la vitesse à laquelle nous souhaitons le voir se déplacer et l'intervalle de secondes entre la trame précédente et la nouvelle, au **Rigidbody2D** du joueur, faisant usage du système de physique de Unity afin de mouvoir notre joueur dans son environnement.

```
/// <summary>
/// MovePlayer est une fonction qui applique une vitesse au joueur afin que celui-
/// ci puisse se déplacer de gauche à droite
/// </summary>
/// <param name="horizontal_movement"></param>
private void MovePlayer(float horizontal_movement)
{
    Vector3 targetVelocity = new Vector2(horizontal_movement,
    player_rigidbody2d.velocity.y); // Création de la vitesse cible du joueur
    FlipSprite(horizontal_movement);
    player_animator.SetFloat("speed", Mathf.Abs(horizontal_movement));

    player_rigidbody2d.velocity = Vector3.SmoothDamp(player_rigidbody2d.velocity,
    targetVelocity, ref velocity, .05f); //déplacement du joueur en appliquant une
    vitesse à son rigidbody2D par rapport à la vitesse actuelle et à celle ciblée
}
```

En ce qui concerne l'implémentation du saut, celle-ci n'a pas beaucoup changée par rapport au diagramme de flux. Si le joueur est en train de remonter le temps alors celui-ci ne peut pas sauter, dans le cas contraire celui-ci peut sauter s'il se trouve au sol et qu'il appuie sur la touche **W**.

```
/// <summary>
/// Update est appelé une fois par mise à jour de trame
/// </summary>
private void Update()
{
    CheckIfGrounded();

    if (!is_rewinding) // Bloque les actions du joueurs si celui-ci remonte le
    temps
    {
        if (is_grounded && Input.GetKey(KeyCode.W))
        {
            player_rigidbody2d.velocity = Vector2.up * JUMPFORCE; // Applique une
            force de saut au joueur
        }
    }
    ...
}
```

La fonctionnalité qui vérifie si le joueur est au sol est très simple et fait usage des outils mis à disposition par Unity, un objet nommé **Feet_Position** est attaché au joueur dans la scène du jeu, cet objet est placé au pieds du joueur comme son nom l'indique. Sa position est ensuite récupérée et stockée, les parties de l'environnement du joueur représentant le sol sont définies et la fonction **Physics2D.OverlapCircle()** est appelée pour créer un cercle avec un rayon défini autours de la position des pieds du joueur en cherchant si le sol est détecté dans ce cercle.

```
/// <summary>
/// CheckIfGrounded est une fonction qui vérifie si le joueur est actuellement au
/// sol en se basant sur la position de ses pieds et la distance de ceux-ci par
/// rapport au sol
/// </summary>
private void CheckIfGrounded()
{
    is_grounded = Physics2D.OverlapCircle(feet_position.position, CHECKRADIUS,
    what_is_ground); // crée un cercle autours des pieds du joueur avec un radius
    constant afin de vérifier si le sol est touché
}
```

Cela permet de savoir de manière constante si le joueur est au sol ou non et cela permet de s'assurer que le joueur est bien au sol avant de lui permettre de sauter.

Implémentation du retour dans le temps

Le retour dans le temps reste fidèle à l'idée présentée dans le diagramme de flux en étant la clé qui autorise les autres fonctionnalités à fonctionner, tout mouvement du joueur est bloqué pendant que celui-ci remonte le temps.

Afin d'enregistrer les mouvements du joueur pour pouvoir remonter le temps par la suite, une classe a été créée pour lister les trois éléments qui seront sauvegardés.

```
/// <summary>
/// RewindData est une classe utilisée pour instancier une liste en précisant les
/// éléments qu'elle contient
/// </summary>
public class RewindData
{
    public Vector2 player_position;
    public bool is_flipped;
    public float player_speed;
}
```

Un **Vector2** permet l'enregistrement de la position du joueur sur l'axe **x** et **y**, un **bool** enregistre l'état (*true*, *false*) du retournement du **sprite** du joueur afin de savoir dans quelle direction celui-ci regarde et finalement un dernier **bool** enregistre la vitesse actuelle du joueur pour assurer l'animation de celui-ci pendant le retour dans le temps.

Une liste basée sur la classe est donc créée et utilisée par le retour dans le temps, l'ajout de ces données est effectué dans la fonction **FixedUpdate** qui permet un enregistrement continu de la position du joueur lorsque celui-ci se déplace étant donné que son déplacement est géré dans la même fonction.

Finalement une vérification est effectuée afin de vérifier si le joueur a appuyé et maintient la touche **R**, provoquant ainsi le blocage des autres fonctionnalités pour remonter le temps.

```
/// <summary>
/// RewindData est une fonction qui permet au joueur de remonter le temps en
/// lisant ses actions enregistrées une à la fois
/// </summary>
private async void RewindData()
{
    if (recorded_data.Count > 0) // S'assure que la liste n'est pas vide
    {
        await Task.Delay(200); // Permet de ralentir et rendre le retour dans le
        temps "un peu plus lisse"
        int index = recorded_data.Count - REWINDINDEX;

        transform.position = recorded_data[index].player_position;
        player_sprite_renderer.flipX = recorded_data[index].is_flipped;
        player_animator.SetFloat("speed",
            Mathf.Abs(recorded_data[index].player_speed));

        recorded_data.RemoveAt(index); // Retire la donnée qui a été lue

        player_rigidbody2d.velocity = Vector2.zero ;
    }
}
```

Le dernier élément de la liste est lu et effacé de celle-ci tant que le joueur maintient la touche appuyée, bien sûr afin de permettre à l'utilisateur d'arrêter la lecture à n'importe quel moment, une seule donnée est lue à la fois.

Implémentation de l'interaction avec les leviers

Comme pour la fonctionnalité de retour dans le temps, l'interaction avec les leviers est une copie conforme de l'idée présentée dans le diagramme de flux, vérifiant si le joueur est assez proche pour interagir avec le levier, si celui-ci appuie sur **E** et modifiant l'environnement en fonction de l'interaction.

Le joueur est capable d'activer et de désactiver les leviers, l'interaction est effectuée en vérifiant si le **collider** du joueur est détecté dans celui du levier, permettant ainsi à l'utilisateur d'interagir à l'aide de la touche **E** si celui-ci le désire.

Une fois l'interaction effectuée, un tableau contenant les objets associés au levier (comme les sorties et plateformes) est ensuite lu et les valeurs **SetActive** de ceux-ci sont inversées.

```
/// <summary>
/// InteractWithLever est une fonction qui permet au joueur d'interagir avec les
leviers du jeu, modifiant ainsi l'environnement de celui-ci pour résoudre des
puzzles
/// </summary>
private void InteractWithLever()
{
    is_lever_off = !is_lever_off;
    lever_animator.SetBool("is_lever_off", is_lever_off);
    lever_animator.SetTrigger("interact");
    foreach (var obstacle in connected_obstacles)
    {
        obstacle.SetActive(!obstacle.activeSelf);
    }
}
```

Implémentation du niveau et des objets dans celui-ci

Afin d'offrir un environnement permettant au joueur d'utiliser la fonctionnalité de retour dans le temps pour résoudre des puzzles, une conception a été effectuée (voir page 17 à 18).

Le niveau fait usage de **tilemaps** mises à disposition par Unity et permettant d'appliquer les graphismes de l'environnement dans un canevas quadrillé, une fois l'environnement "peint" il ne reste plus qu'à ajouter un **collider** adapté aux tilemaps pour générer les collisions physiques de l'environnement.

Afin d'éviter au joueur de se retrouver coincé entre chaque **collider** de chaque tuile, l'utilisation en **composite** est appliquée au **collider** de la **tilemap**, permettant une fusion de toutes les collisions de la **tilemap**.

Une autre chose qui a été entreprise afin d'enrichir l'environnement et l'expérience du joueur est la création d'une couche ténébreuse cachant les zones que le joueur n'a pas encore débloqués.

Pour créer cette illusion, des **tilemaps** ont été ajoutées par-dessus les autres et se retrouve automatiquement retirées les unes après les autres lorsque leur zone est débloquée.

Les leviers retrouvés à différents endroits dans le niveau sont créés à partir d'un **prefab** que j'ai créé, il existe deux **prefabs** dans le projet, l'**Exit_Trigger** et le **Lever**.

En créant l'objet une première fois et en le transformant en **prefab**, je gagne du temps sur l'implémentation puisque je n'ai plus besoin de recréer l'objet dont j'ai besoin.

Finalement j'ai créé l'objet qui représente le joueur, l'objet auquel j'ai assigné le script **Player_controller.cs** afin de permettre au joueur de le contrôler.

Ayant rencontré un problème causé par la friction entre le joueur et son environnement, lui permettant de se coller à des murs lorsque celui-ci se déplaçait continuellement contre ceux-ci, j'ai créé un **material** avec une friction de zéro, assignant celui-ci au joueur afin de corriger le problème de friction.

Finalement l'implémentation du menu de pause au niveau visuel est effectué à l'aide des objets proposés par Unity, en effet celui-ci propose des objets dans une catégorie libellée **UI**.

Arborescence du projet sur Unity

Afin d'être en mesure de trouver facilement tous les éléments nécessaires au cours de la création du projet, une arborescence a été créée pour classer les éléments. L'arborescence générale est la suivante :

- **Fardel_Bastien_BLOCK_TPI** (Contient le projet dans son ensemble)
 - **.vs** (Dossier généré automatiquement par Visual Studio)
 - **Assets** (Dossier contenant l'ensemble des ressources du projet)
 - **Artworks** (Dossier contenant les graphismes et animations)
 - **Icon** (Contient les icônes du jeu)
 - **Lever** (Contient les graphismes et animations du levier)
 - **Player** (Contient les graphismes et animations du joueur)
 - **World** (Contient les graphismes de l'environnement)
 - **Materials** (Contient les matériaux)
 - **Prefabs** (Contient les préfabriqués)
 - **Scenes** (Contient les différentes scènes)
 - **Scripts** (Dossier contenant les scripts du projet)
 - **Classes** (Contient les scripts concernant les classes)
 - **Lever** (Contient les scripts concernant les leviers)
 - **Menus** (Contient les scripts concernant des menus)
 - **Main** (Contient le script du menu principal)
 - **Pause** (Contient le script du menu de pause)
 - **Player** (Contient les scripts concernant le joueur)
 - **Zone** (Contient les scripts concernant la gestion de zones)
 - **TextMesh Pro** (Dossier utilisé contenant les éléments UI utilisés)

Librairies du projet

Librairies installées dans Unity

- com.unity.2d.animation@7.0.4
- com.unity.2d.common@6.0.3
- com.unity.2d.path@5.0.2
- com.unity.2d.pixel-perfect@5.0.1
- com.unity.2d.psdimporter@6.0.3
- com.unity.2d.sprite@1.0.0
- com.unity.2d.spriteshape@7.0.4
- com.unity.2d.tilemap.extras@2.2.1
- com.unity.2d.tilemap@1.0.0
- com.unity.burst@1.6.5
- com.unity.collab-proxy@1.15.15
- com.unity.ext.nunit@1.0.6
- com.unity.feature.2d@1.0.0
- com.unity.ide.rider@3.0.13
- com.unity.ide.visualstudio@2.0.14
- com.unity.ide.vscode@1.2.5
- com.unity.mathematics@1.2.6
- com.unity.modules.ai@1.0.0
- com.unity.modules.androidjni@1.0.0
- com.unity.modules.animation@1.0.0
- com.unity.modules.assetbundle@1.0.0
- com.unity.modules.audio@1.0.0
- com.unity.modules.cloth@1.0.0
- com.unity.modules.director@1.0.0

- com.unity.modules.imageconversion@1.0.0
- com.unity.modules.imgui@1.0.0
- com.unity.modules.jsonserialize@1.0.0
- com.unity.modules.particlesystem@1.0.0
- com.unity.modules.physics@1.0.0
- com.unity.modules.physics2d@1.0.0
- com.unity.modules.screencapture@1.0.0
- com.unity.modules.subsystems@1.0.0
- com.unity.modules.terrain@1.0.0
- com.unity.modules.terrainphysics@1.0.0
- com.unity.modules.tilemap@1.0.0
- com.unity.modules.ui@1.0.0
- com.unity.modules.uielements@1.0.0
- com.unity.modules.uielementsnative@1.0.0
- com.unity.modules.umbra@1.0.0
- com.unity.modules.unityanalytics@1.0.0
- com.unity.modules.unitywebrequest@1.0.0
- com.unity.modules.unitywebrequestassetbundle@1.0.0
- com.unity.modules.unitywebrequestaudio@1.0.0
- com.unity.modules.unitywebrequesttexture@1.0.0
- com.unity.modules.unitywebrequestwww@1.0.0
- com.unity.modules.vehicles@1.0.0
- com.unity.modules.video@1.0.0
- com.unity.modules.vr@1.0.0
- com.unity.modules.wind@1.0.0
- com.unity.modules.xr@1.0.0

- com.unity.services.core@1.0.1
- com.unity.test-framework@1.1.31
- com.unity.textmeshpro@3.0.6
- com.unity.timeline@1.6.4
- com.unity.ugui@1.0.0
- com.unity.visualscripting@1.7.6

Erreurs restantes

Le jeu ne possède pas de fin

Le jeu ne possède actuellement pas de fin, le joueur se retrouve donc dans une salle vide.

Liste des documents fournis

- Planification initiale (.pdf électronique et papier physique)
- Rapport de projet TPI (ce document en .pdf et papier physique)
- Journal de travail (.pdf électronique et papier physique)
- Code source de l'application (.pdf électronique et papier physique)
- Exécutable du produit (électronique uniquement)
- Diagramme de flux (.pdf électronique et papier physique)

Conclusion

Objectifs atteints

L'ensemble des objectifs ont été atteints à l'exception de l'objectif numéro trois concernant l'ajout d'au minimum un test unitaire sur la gestion de la pile pour le retour dans le temps.

Objectifs non atteints

Le retour dans le temps ne possède pas de tests unitaires

Il était demandé de produire des tests unitaires pour la fonctionnalité de retour dans le temps, cependant en vue de l'approche de la date limite pour le rendu et la concentration principale sur clôture de la documentation, ces tests n'ont malheureusement pas vu le jour.

Points positifs / négatifs

Points positifs

Je suis plutôt heureux avec le résultat “final” que j’ai produit, le jeu est appréciable et a du potentiel, mon code est propre et quand je compare celui-ci à celui de mon ancien projet, je me rends compte que j’ai beaucoup évolué et que j’ai réussi à grandement optimiser mon code.

De plus je suis heureux d’avoir pu retourner à la programmation en C# et sur Unity, je pense continuer dans le futur sur le jeu ou commencer mes propres projets, j’ai toujours rêver de créer un jeu offrant à l’utilisateur un moyen de relaxer en observant son environnement.

En ce qui concerne le déroulement du projet, mon opinion est mixé, je suis vraiment fier d’avoir mieux géré ce projet comparé à celui de mon pré-TPI et même s’il me reste sûrement quelques modifications à effectuer à ma gestion, j’en reste fier.

La communication entre les experts et le chef de projet m’a semblé être positive et fluide, particulièrement celle entre mon chef de projet et moi, celui-ci s’est montré présent dès que le besoin se faisait sentir et n’a pas hésité à assurer le bon déroulement de ce TPI du fait de sa présence et son assistance.

Points négatifs

Malgré l’aspect positif de ce projet, je dois avouer avoir un opinion mixé sur le déroulement du projet, il m’est arrivé de commettre quelques erreurs en perdant du temps précieux, principalement sur la documentation que j’ai malheureusement négligée.

En dehors de cela, j’ai aussi le regret de ne pas avoir pu implémenter une fin au projet mais je ne regrette pas d’avoir sacrifié celle-ci pour m’assurer de la bonne documentation de ce projet.

Difficulté particulière

J’ai particulièrement rencontré une difficulté au niveau de la documentation que j’ai sous-estimé en terme de travail nécessaire pour la compléter.

Heureusement j’ai pu obtenir assez de temps pour la compléter et terminer celle-ci dans les temps et je suis relativement fier du résultat de celle-ci.

Je peux aussi citer les tests unitaires que je n’ai malheureusement pas réussi à implémenter dans le temps imparti, un de mes regrets puisque ceux-ci étaient mon opportunité pour m’améliorer et terminer tous les points techniques spécifiques demandés.

Annexes

Résumé du rapport de TPI

Situation de départ

Ce projet a commencé le 02 mai 2022 à 08h00, après avoir reçu mon cahier des charges et l'avoir lu auprès du premier expert que j'ai eu le plaisir de rencontrer le matin même, j'ai signé celui-ci, signant le début officiel de mon travail.

J'ai donc commencé à travailler sur ma planification pour ce projet et sur sa conception afin de mettre en place les fondations de mon projet.

Mise en œuvre

Une fois la première semaine de planification et d'analyse terminée, j'ai commencé l'implémentation du projet, créant B'lock et ses toutes premières fonctionnalités, implémentant les mouvements du joueur, un niveau pour les tester et les premiers puzzles.

Après une semaine de travail sur l'implémentation des premières fonctionnalités, j'ai enfin pu commencer à implémenter le retour dans le temps et les menus dans le jeu.

J'ai pris soin de remplir mon journal de travail de manière journalière afin de m'assurer d'avoir un suivi de mon travail tout au long de l'implémentation.

Résultat

Finalement après quatre semaines de travail laborieux, j'ai enfin obtenu un produit jouable, bien qu'il lui manque une fin, et je peux dire que j'en suis fier, le jeu contient un niveau divisé en trois zones qui mettent le joueur à l'épreuve, l'incitant à utiliser toutes les fonctionnalités qui ont été implémentées dans le jeu.

Je suis aussi heureux de ressortir de ce TPI avec une meilleure compréhension de la gestion d'un projet pareil.

Cependant, je dois noter que je suis attristé de ne pas avoir pu implémenter les tests unitaires à mon projet.

Sources – références bibliographiques

Sites web

GameDevTraum. (Consultation le 12 mai 2022). *ACTIVATE and DEACTIVATE GameObjects through CODE in Unity*. Consulté le 12 mai 2022, sur gamedevtraum.com : { [ACTIVATE and DEACTIVATE GameObjects through CODE in Unity](https://gamedevtraum.com) } (gamedevtraum.com)

Unity. (Consultation le 12 mai 2022). *GameObject.SetActive*. Consulté le 12 mai 2022, sur docs.unity3d.com: [Unity - Scripting API: GameObject.SetActive](https://docs.unity3d.com/2020.1/Documentation/ScriptReference/GameObject.SetActive.html) ([unity3d.com](https://docs.unity3d.com))

Unity Forums. (12 janvier 2018). *How to add new value to array*. Consulté le 16 mai 2022 sur forum.unity.com : [How to add new value to array - Unity Forum](https://forum.unity.com/threads/how-to-add-new-value-to-array-144459)

Unity. (Consultation le 16 mai 2022). *Array*. Consulté le 16 mai 2022, sur docs.unity3d.com : <https://docs.unity3d.com/2020.1/Documentation/ScriptReference/Array.html>

Unity Forums. (20 juillet 2020). *C# : Arrays with multiple types of data at each points?*. Consulté le 16 mai 2022, sur forum.unity.com : [https://forum.unity.com/threads/c-arrays-with-multiple-types-of-data-at-each-point.144459/](https://forum.unity.com/threads/c-arrays-with-multiple-types-of-data-at-each-point-144459/)

Unity Forums. (17 septembre 2018). *Trying to add multiple values into a list*. Consulté le 16 mai 2022, sur forum.unity.com : [Trying to add multiple values into a list - Unity Forum](https://forum.unity.com/threads/trying-to-add-multiple-values-into-a-list-144459)

Samual Sam. (20 août 2018). *How to pop the first element from a C# List?*. Consulté le 16 mai 2022, sur www.tutorialspoint.com : [How to pop the first element from a C# List? \(tutorialspoint.com\)](https://www.tutorialspoint.com/csharp/csharp_remove_first_element_from_list.htm)

willa. (25 janvier 2014). *Add multiple items to a list*. Consulté le 16 mai 2022, sur stackoverflow.com : [c# - Add multiple items to a list - Stack Overflow](https://stackoverflow.com/questions/214459/c-add-multiple-items-to-a-list)

vittu1994. (28 septembre 2015). *how to know when list is empty?*. Consulté le 16 mai 2022, sur answers.unity.com : [how to know when list is empty? - Unity Answers](https://answers.unity.com/questions/144459/how-to-know-when-list-is-empty-unity-answers)

lakshganga. (18 septembre 2013). *Check if list is empty in C# [closed]*. Consulté le 16 mai 2022, sur stackoverflow.com : <https://stackoverflow.com/questions/18867180/check-if-list-is-empty-in-c-sharp>

Microsoft. (Consultation le 17 mai 2022). *Enumerable.Any Méthode*. Consulté le 17 mai 2022, sur docs.microsoft.com : <https://docs.microsoft.com/fr-fr/dotnet/api/system.linq.enumerable.any?view=net-6.0>

Vidéos disponibles sous YouTube

TUTO UNITY FR. (22 février 2020). *Créer un jeu en 2D facilement avec Unity (#1 : Création du premier niveau)*. Consultée le 9 mai 2022, sur YouTube : [\(291\) Créer un jeu en 2D facilement avec Unity \(#1 : Création du premier niveau\) - YouTube](#)

JGN Games. (2 juillet 2016). *Making a Lever in Unity – 2. Scripting*. Consultée le 12 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=t400AgozXds>

SpeedTutor. (23 octobre 2016). *Lever / Switch Puzzle System Demo – Unity*. Consultée le 12 mai 2022, sur YouTube : https://www.youtube.com/watch?v=p-7g1_15IHQ

Trever Mock. (26 janvier 2022). *How to create a Replay System like in Super Meat Boy using Unity*. Consultée le 12 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=ilQQstDnX2I>

Chloe Nothing. (1^{er} novembre 2022). *[Unity 2d]Use of the time rewind mechanism*. Consultée le 12 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=hyr0ifmWzA4>

Superjustin5000. (10 décembre 2021). *TIME REWIND like Braid | Control Time in Unity | Part 1*. Consultée le 16 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=SQNpWGxaxVY>

TUTO UNITY FR. (13 juin 2020). *Créer un jeu en 2D facilement avec Unity (#17 : Menu Principal 1/2)*. Consultée le 20 mai 2022, sur YouTube : https://www.youtube.com/watch?v=4LkiX_XioXg

Brackeys. (29 novembre 2017). *START MENU in Unity*. Consultée le 20 mai 2022, sur YouTube : https://www.youtube.com/watch?v=zc8ac_qUXQY

Brackeys. (20 décembre 2017). *PAUSE MENU in Unity*. Consultée le 20 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=JivuXdrIHK0>

TUTO UNITY FR. (27 juin 2020). *Créer un jeu en 2D facilement avec Unity (#19 : Menu Pause)*. Consultée le 20 mai 2022, sur YouTube : <https://www.youtube.com/watch?v=xakb-Jok5Xg>

Temps passé par tâches

Ceci est basé sur le temps calculé dans mon journal de travail, le diagramme suivant en est le résultat.

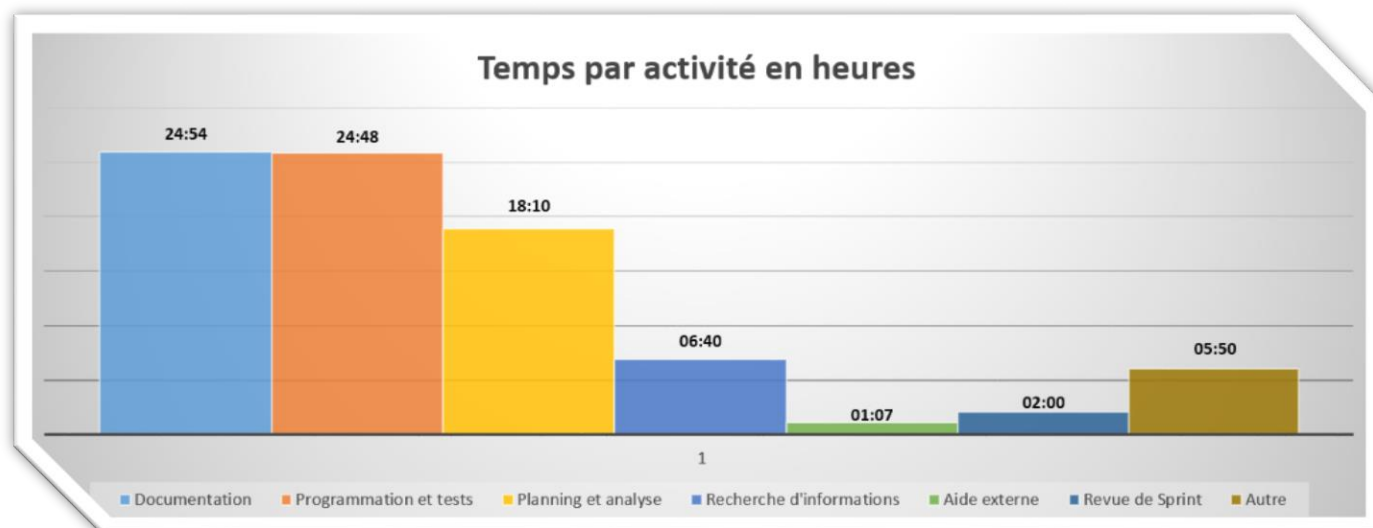


Figure 12 : Diagramme en barres du temps passé par activité

Au total, le temps passé à travailler sur ce projet est de 83 heures et 29 minutes.

Manuel d'installation

Afin de pouvoir jouer au jeu, il faut installer la release finale disponible sous Github : [Release Version finale \(TPI\) de B'lock - BFL-CPNV/TPI Projet B-lock BFL \(github.com\)](#)

Vous téléchargerez donc un fichier compressé, extrayez-le sur votre bureau et ouvrez-le, il ne vous reste plus qu'à lancer l'exécutable qui est reconnaissable grâce à l'icône du jeu.

Archives du projet

Les archives du projet sont disponibles sous le répertoire du projet : [BFL-CPNV/TPI Projet B-lock BFL: Repository du projet B'lock pour le TPI \(github.com\)](#)

Table des illustrations

Figure 1 : Planification détaillée - Semaine 1.....	12
Figure 2 : Planification détaillée - Semaine 2.....	13
Figure 3 : Planification détaillée - Semaine 3.....	14
Figure 4 : Planification détaillée - Semaine 4.....	15
Figure 5 : Planification détaillée - Semaine 5.....	16
Figure 6 : Stories du projet B'lock.....	18
Figure 7 : Maquette Conceptuelle du niveau	20
Figure 8 : Indices des commandes pour le joueur	20
Figure 9 : La sortie est fermée mais la plateforme est activée.....	21
Figure 10 : Le joueur est enfermé dans une salle.....	21
Figure 11 : Stories sur IceScrum	22
Figure 12 : Diagramme en barres du temps passé par activité	40