

EUREKA!

GLOSSAIRE

A

AND	
porte logique ET.....	5, 9, 20, 24
Assets	
dossier maître dans unity.....	20

B

Buffer	
porte logique entrée = sortie	25

C

C#	
Language de programmation.....	5, 24

E

EXE	
programmes windows	27

F

FALSE	
FAUX	24, 25

G

gameobjects	
Objects du jeu	20
Gates	
Portes logique	20

I

icescrum	
logiciel de gestion de projet.....	7

N

NAND	
porte logique NON-ET	25
NOR	
Porte logique NON-OU	25
NOT	
Porte logique NON.....	25

O

OR	
Porte logique OU	9, 25

P

POO	
Programmation orientée object	5
préfabs	
game objects sauvegarder dans les assets	20

R

random	
aléatoire	13
release	
dernière version du programme.....	27

S

screenshot	
Instantané, photo	27, 30
slider	
glisseur.....	28, 30
sprites	
Image pour les éléments du jeu	20

T

TRUE	
VRAI	24, 25

U

UI	
User interface, interface graphique	15, 20, 29
Unity	
Moteur graphique de jeu	21

V

Visual Studio	
Application pour programmer	5, 24

W

winrar	
programme d'archivage de dossiers.....	27

X

XNOR	
Porte logique NON-OU exclusif	25
XOR	
Porte logique OU exclusif	5, 25

TABLE DES MATIÈRES

Glossaire.....	2
Table des matières	4
Analyse préliminaire	5
Introduction	5
Matériel à disposition	5
Prérequis	5
Objectifs	5
Planification initiale.....	6
Planification détaillée.....	6
Analyse / Conception	9
Analyse.....	9
Uses cases et scénarios	9
Concept	11
Diagramme de flux	11
Design de l'UI	15
Stratégie de test	18
Test Unitaire.....	18
Tests d'acceptation	18
Risques techniques	19
Risque techniques	19
Réalisation.....	19
Dossier de réalisation.....	19
Scripts.....	20
Dossier.....	23
Scripts.....	23
Version de windows	23
Version de Unity.....	24
Version de visual studio	25
Version du programme	25
Matériel utilisé	25
Librairies.....	25
Description des tests effectués	26
Tests d'acceptations.....	26
Test unitaire	27
Erreurs restantes.....	28
Liste des documents fournis	28
Conclusions	29
Annexes.....	30
Résumé du rapport du TPI / version succincte de la documentation	30
Sources – Bibliographie.....	30
Journal de travail.....	30
Manuel d'Installation	30
Manuel d'Utilisation.....	30
Archives du projet	33
Version de la documentation	33
Table des illustrations	34

ANALYSE PRÉLIMINAIRE

INTRODUCTION

Eurêka est un serious game de résolution de puzzle, créer avec le langage C# et le moteur graphique unity, il a pour but de proposer au joueur la possibilité de résoudre des puzzles sous la forme d'un schéma avec des portes logiques générées aléatoirement, le joueur devra deviner le résultat de tous les résultats afin de réussir le puzzle, plusieurs niveaux de difficulté s'offrent à lui, il peut aussi créer son propre puzzle.

Il a aussi la possibilité de prendre en photo son puzzle, afin de le sauvegarder sur le disque.

MATÉRIEL À DISPOSITION

Liste de matériel physique et de logiciel mis à disposition

- 1 PC du CPNV
- Unity 2020
- Visual Studio 2020
- Visual Studio Code
- Suite office

PRÉREQUIS

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

OBJECTIFS

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

Les objectifs du projet étant :

- Résolution de puzzle
Le but étant que le joueur puisse résoudre le puzzle en devinant la sortie(résultat) du puzzle (Exemple : une porte AND avec en entrée 1 et 0, donne 0 comme sortie)
Puzzle Aléatoire
Création Aléatoire des puzzles, chaque porte est générée aléatoirement, un nombre fixe de portes est présent, qui change en fonction du niveau de difficulté choisi par le joueur : en mode facile il y'a 3 portes logiques, en mode moyen, il y'a 9 portes logiques, en mode difficile, il y'a 19 portes logiques.
- Puzzle manuel
Création manuelle des puzzles, avec la possibilité de choisir les logiques portes dans le jeu, comme pour le puzzle aléatoire, les portes sont déjà prédéfinies par la difficulté, ils ne peuvent être déplacées, par contre le joueur peut choisir quelle type de porte (AND, XOR, etc...), et par la suite, de compléter le puzzle par la suite, comme un puzzle aléatoire
- Capture de puzzle
Permet de prendre en photo le puzzle courant, qui est sauvegardé sur le disque
- Théorie des portes logiques
Affiche sur l'écran comment les portes logiques fonctionnent sur le puzzle courant du comment fonctionne, leur entrée, sortie, avec une table de vérité

PLANIFICATION INITIALE

Eurêka!

Eureka!

1

lun, 5.2.2022

			2 mai 2022							9 mai 2022							16 mai 2022							23 mai 2022							30 mai 2022							
			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	
TÂCHE	DÉBUT	FIN	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 - Analyse																																						
Planification Initiale	2.5.22	2.5.22																																				
Création de l'environnement de travail	2.5.22	3.5.22																																				
Conception des uses cases et scénario	3.5.22	5.5.22																																				
Conception des diagrammes de flux	5.5.22	8.5.22																																				
Conception de l'interface graphique	8.5.22	9.5.22																																				
Création des sprints, tâches, tests sur IceScrum	9.5.22	10.5.22																																				
Réalisation																																						
Réalisation de l'interface graphique avec la théorie des portes	10.5.22	12.5.22																																				
Réalisation du système de portes aléatoires	12.5.22	17.5.22																																				
Réalisation du système de portes manuelle	17.5.22	22.5.22																																				
Réalisation du système de capture du puzzle	22.5.22	24.5.22																																				
Documentation et Tests																																						
Mise à jour du rapport de projet	24.5.22	30.5.22																																				
Création des test unitaire, validation des tests d'acceptation	30.5.22	31.5.22																																				

1 Planification initiale

La première partie étant la planification et l'analyse du projet, qui englobe, toute la partie de création de diagramme, uses cases et scénarios, et maquettes, elle s'étends sur une période d'une semaine et demi, après que la première partie, il y a la partie réalisation, là où tout le code est créé, l'aspect visuel, cette partie compte pour 2 semaine de travail, et ensuite, une partie documentation, qui prends 3 jours, à noter que la documentation du projet et mis à jour pendant la planification, et réalisation, la dernière partie n'est là que pour rentre la documentation plus jolie, et aussi pour tous ce qui est tests.

PLANIFICATION DÉTAILLÉE

Heure	lundi, 2 mai 2022	mardi, 3 mai 2022	mercredi, 4 mai 2022	jeudi, 5 mai 2022	vendredi, 6 mai 2022
08:00	Planification Initiale	Installation de l'environnement de travail			Conception des diagrammes de flux
08:45					
08:50		Conception des uses cases scénario		Conception des uses cases scénario	
09:35					
09:50					
10:35					
10:40					
11:25					
11:30					
12:15					
13:30	Planification Initiale	Conception des uses cases scénario		Conception des diagrammes de flux	Conception des diagrammes de flux
14:15					
14:20					
15:05					
15:20					
16:05					
16:10					
16:55	Installation de l'environnement de travail				

2 Planification détaillée semaine 1

Le premier jour du TPI, ce consacre principalement à la création de la planification initiale, mais aussi avec une partie d'installation de l'environnement de travail, cela prend en compte, l'installation d'unity, visual studio, installation de packages nécessaires, qui s'étend au début de la matinée de mardi.

À partir de mardi jusqu'à jeudi, sans compter mercredi, une partie importante du projet, étant la création des uses cases et scénario, qui aideras par la suite l'implémentation des fonctionnalités demandées dans le cahier des charges.

La conception des diagrammes de flux est mise en place jeudi après-midi et toute la journée de vendredi.

Heure	lundi, 9 mai 2022	mardi, 10 mai 2022	mercredi, 11 mai 2022	jeudi, 12 mai 2022	samedi, 0 janvier 1900
08:00	Conception de l'interface graphique	création des sprints, tâches, tests sur ice scrum			Réalisation du système de porte aléatoire
08:45					
08:50					
09:35					
09:50					
10:35					
10:40					
11:25					
11:30					
12:15					
13:30	création des sprints, tâches, tests sur ice scrum	Réalisation de l'interface graphique avec la théorie des portes			Réalisation du système de porte aléatoire
14:15					
14:20					
15:05					
15:20					
16:05					
16:10					
16:55					

3 Planification détaillée semaine 2

Le début de la deuxième semaine est consacrée pour la partie planification et un peu de l'implémentation, une partie, est donnée pour la conception de l'interface graphique, et de la création des sprints, stories, et tâches sur icescrum, mais aussi une partie d'implémentation, création de l'ui sur Unity, la création du système de théorie des portes logiques, qui affiche les portes logiques à l'écrans, qui prends en compte l'ui création du système de porte aléatoire, ou le puzzle créera un puzzle aléatoire afin que le joueur puisse le compléter.

Heure	lundi, 16 mai 2022	mardi, 17 mai 2022	mercredi, 18 mai 2022	jeudi, 19 mai 2022	vendredi, 20 mai 2022
08:00	Réalisation du système de porte aléatoire	Réalisation du système de porte aléatoire			Réalisation du système de portes manuelle
08:45					
08:50					
09:35					
09:50					
10:35					
10:40					
11:25					
11:30					
12:15					
13:30	Réalisation du système de porte aléatoire	Réalisation du système de portes manuelle			Réalisation du système de portes manuelle
14:15					
14:20					
15:05					
15:20					
16:05					
16:10					
16:55					

4 Planification détaillée semaine 3

La troisième semaine, porte seulement sur de l'implémentation, en commençant par la création du puzzle aléatoire qui s'étends sur 2 jours complet, et ensuite, la partie de création de porte manuelle, c'est-à-dire, la possibilité de créer son propre puzzle, en choisissant ses portes logiques, et ensuite le compléter, qui s'étends sur tout le reste de la semaine.

Heure	lundi, 23 mai 2022	mardi, 24 mai 2022	mercredi, 25 mai 2022	jeudi, 26 mai 2022	vendredi, 27 mai 2022			
08:00	Réalisation du système de capture de puzzle	Réalisation du système de capture de puzzle						
08:45								
08:50								
09:35								
09:50								
10:35								
10:40								
11:25								
11:30								
12:15								
13:30	Réalisation du système de capture de puzzle	Mise à jour du rapport de projet						
14:15								
14:20								
15:05								
15:20								
16:05								
16:10								
16:55								

5 Planification détaillée semaine 4

La quatrième semaine est coupée par le pont de l'ascension, seulement le lundi et mardi compte, le lundi, se porte sur la création du système de capture de puzzle, et mardi après-midi, la mise à jour de la documentation, qui prends en compte, l'amélioration de la qualité de la documentation, ajout d'images, de description, des choses pas documenté lors de l'implémentation, la conceptions et analyse.

Heure	lundi, 30 mai 2022	mardi, 31 mai 2022	mercredi, 1 juin 2022	jeudi, 2 juin 2022	vendredi, 3 juin 2022
08:00	Mise à jour du rapport de projet	Création des tests unitaire, validation des tests d'acceptation			
08:45					
08:50					
09:35					
09:50					
10:35					
10:40					
11:25					
11:30					
12:15					
13:30	Création des tests unitaire, validation des tests d'acceptation	Création des tests unitaire, validation des tests d'acceptation			
14:15					
14:20					
15:05					
15:20					
16:05					
16:10					
16:55					

6 Planification détaillée semaine 5

La cinquième semaine, est la dernière semaine de travail, qui prends en compte, les tests unitaire, validations des texte d'acceptation avec une partie de mise à jour de projet, le rendu du projet, étant le mardi 31 mai, il est prise en compte

ANALYSE / CONCEPTION

ANALYSE

USES CASES ET SCÉNARIOS

Afin de comprendre comment, chaque fonctionnalité doit être programmée, et aussi d'avoir un scénario type pour gérer les exceptions j'ai fait des uses cases et scénario, afin de pouvoir bien se mettre d'accord sur ce qu'il faut faire

PUZZLE ALÉATOIRE

Un puzzle aléatoire est généré lors du lancement d'une partie en tant que puzzle aléatoire.

Si le jeu est en échec, c'est-à-dire que le résultat attendu, n'est pas trouvé par le joueur, le jeu ne peut pas se terminer, ce use case et scénario ne gère pas la résolution de puzzle. À l'inverse, si le résultat attendu, est trouvé par le joueur, le jeu est considéré comme trouvé, et un nouveau puzzle est généré, sauf si le joueur décide de quitter le jeu ou de passer sur un puzzle manuel.

Si le joueur décide de fermer le jeu quand un puzzle est en cours, le jeu demande une confirmation, si le joueur décide quand même de quitter le jeu, le puzzle ne sera pas sauvegardé, et un nouveau puzzle sera généré au prochain lancement du jeu, s'il n'y a pas de jeu en cours, le jeu se ferme sans donner de confirmation.

en tant que	Joueur	
Je veux	Avoir un puzzle aléatoire à chaque partie	
Pour	finir le puzzle	
Priorité	M	
Action	Condition	Réaction
Lancement du jeu	Le joueur choisit un puzzle aléatoire	Un puzzle aléatoire est généré
La partie est en échec		la partie ne peut pas se terminer
La partie est gagnée		la partie est terminée
La partie courante est terminée		Un puzzle aléatoire est généré
Fermeture du jeu	la partie est terminée	le jeu se ferme
Fermeture du jeu	une partie est en cours	le jeu demande une confirmation avant de se fermer

7 cas d'utilisation : puzzle aléatoire

PUZZLE MANUEL

Un puzzle manuel est généré lors du lancement de la partie, les ports sont vides mais déjà présents, les liaisons sont aussi déjà présentes.

Le joueur peut donc choisir le type de portes (porte AND, OR, par exemple).

Après avoir confirmé son choix, le puzzle passe en mode résolution de puzzle, le joueur peut donc compléter son propre puzzle.

Une partie peut être considérée comme en échec si le joueur ne choisit pas toutes les portes logiques (laisse une vide par exemple) et le jeu peut être considéré comme terminé, si toutes les portes ont été choisies et le joueur aille confirmer son puzzle. Comme pour les puzzles aléatoires, le jeu demande une confirmation, si une partie est en cours, et ne demande pas si le jeu est déjà terminé.

Je veux	Avoir une puzzle manuel à chaque partie	
Pour	finir le puzzle	
Priorité	M	
Action	Condition	Réaction
Lancement du jeu	Le joueur choisi un Manuel	Un puzzle Manuel est généré
Le joueur choisit une porte de la liste		la porte est placé dans le jeu
Le joueur choisit une porte de la liste	il choisit la même porte	rien ne change
La partie est en échec		la partie ne peux pas se terminer
La partie courante est terminée		Un puzzle Manuel est généré
Fermeture du jeu	la partie est terminée	le jeu se ferme
Fermeture du jeu	une partie est en cours	le jeu demande une confirmation avant de ce fermer

8 cas d'utilisation : puzzle manuel

RÉSOLUTION DE PUZZLE

C'est la partie la plus importante du jeu, c'est ce qui gère toute la partie de résolution des puzzles, aléatoire et manuel, le résultat que le joueur doit trouver, et soit 1 ou 0 (true ou false), il faut donc vérifier si la valeur est dans cette fourchette de données ou que la valeur entrée est bien du décimal, si ce n'est pas le cas, le résultat est considéré comme faux, et un erreur est du coup affichée, si le joueur ne rentre aucun résultat rien ne se passe, si le joueur trouve le bon résultat, le jeu se termine.

Nom	RésolutionPuzzle-01	
en tant que	Joueur	
Je veux	Avoir une puzzle	
Pour	finir le puzzle	
Priorité	H	
Action	Condition	Réaction
Le joueur entre une résultat	le résultat est faux	une erreur est affichée
Le joueur entre une résultat	le résultat est juste	le jeu se termine
Le joueur n'entre pas résultat		rien ne se passe
Le joueur entre une résultat	la valeur est supérieur à un	le résultat est considéré comme faux
Le joueur entre une résultat	la valeur est inférieur à un	le résultat est considéré comme faux
Le joueur entre une résultat	la valeur n'est pas en décimal	le résultat est considéré comme faux

9 cas d'utilisation : résolution de puzzle

CAPTURE D'ÉCRAN

Le système de capture d'écran, permet la sauvegarde en photo du puzzle actuel, le jeu ne peux pas relire l'image pour charger une partie, cela est juste une photo. Si la théorie des portes logique, ou le menu démarrée est affiché, le bouton pour prendre en capture le jeu, n'est ni actif, ni visible, à l'inverse on peut y accéder sans soucis.

Si le jeu ne peut pas avoir accès au fichier de sauvegarde d'image pour quelconques raisons, le jeu retourne une erreur.

Nom	CaptureD'écran-01	
en tant que	Joueur	
Je veux	pouvoir avoir en image le puzzle	
Pour	l'observer	
Priorité	S	
Action	Condition	Réaction
Le joueur prends en photo le puzzle		la photo est sauvegardée dans le disque
Le joueur prends en photo le puzzle	les résultat de la partie est affiché	le bouton n'est pas visible ni actif
Le joueur prends en photo le puzzle	le chemin d'accès est inaccessible	un message d'erreur est retourné au joueur

10 case d'utilisation : capture d'écran

THÉORIES DES PORTES LOGIQUES

A chaque lancement de partie, soit en lançant un puzzle aléatoire, ou après la création d'un puzzle manuel, le jeu affiche un jeu d'icône de portes logiques, et en appuyant sur une icône, un descriptif détaillé, leur icone, et une table de vérité sera possible d'afficher. Il est aussi possible de réafficher cette fenêtre en appuyant sur F1.

Nom	ThéoriePortLogiques-01	
en tant que	Joueur	
Je veux	Avoir afficher sur comment marche les portes logique	
Pour	connaître/être sur des fonctionnement des portes logique	
Priorité	S	
Action	Condition	Réaction
Lancement d'une partie		la fonctionnement sur les porte logique courante est affichée
en appuyant sur F1		la fonctionnement sur les porte logique courante est affichée

11 cas d'utilisation : Théorie des porte logique

CONCEPT

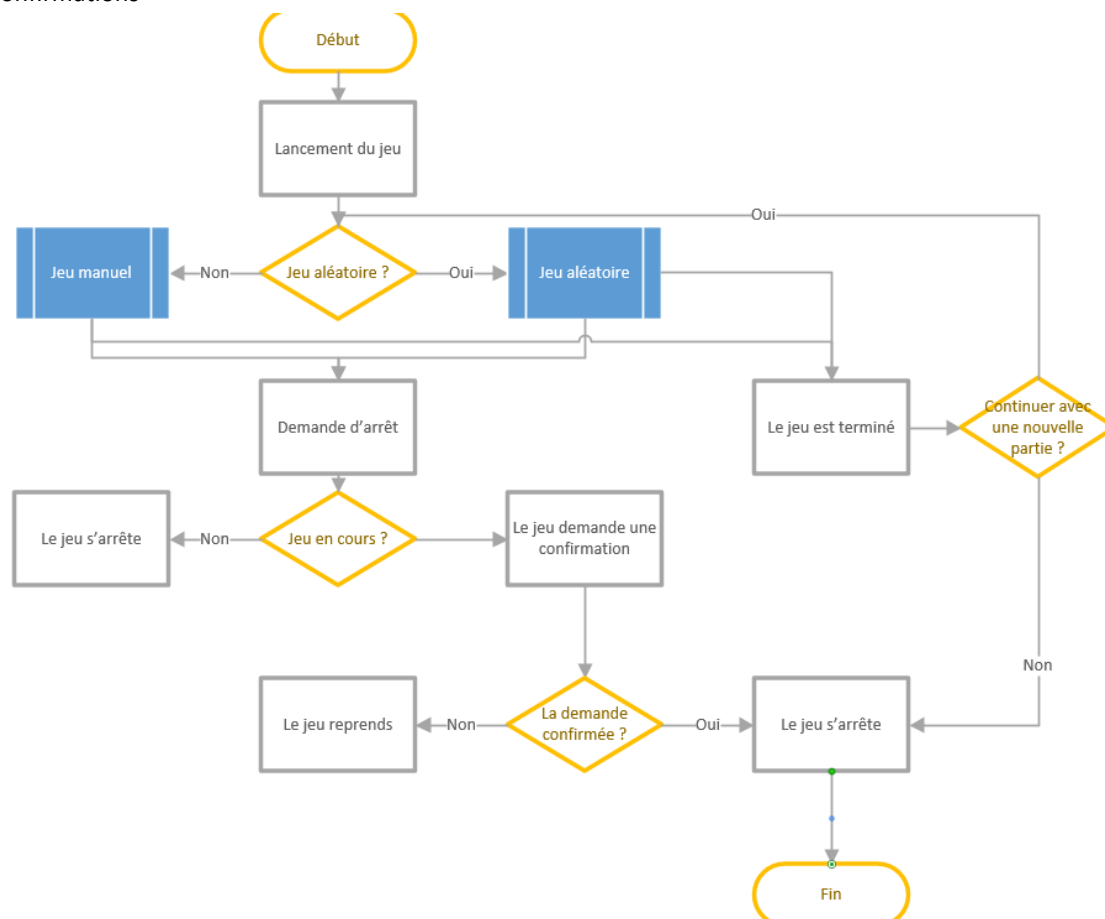
DIAGRAMME DE FLUX

Pour avoir une idée plus précise du comment une fonctionnalité doit être implémentée, j'ai fait plusieurs diagrammes de flux, dont on peut voir comment les fonctions interagissent entre elles

MAIN

Le code main, est le script maître qui gère l'appelle des autres fonctions, et gère aussi la fermeture du programme, le jeu commence toujours par demander si le joueur veut commencer par un puzzle manuel ou un puzzle aléatoire, et appelle la fonction qui gère le type de puzzle choisit. Le joueur peut choisir de continuer ou pas après chaque partie, si le joueur choisit de continuer, le jeu repose la question au joueur sur quel type de puzzle il veut, s'il ne veut pas, le jeu se ferme.

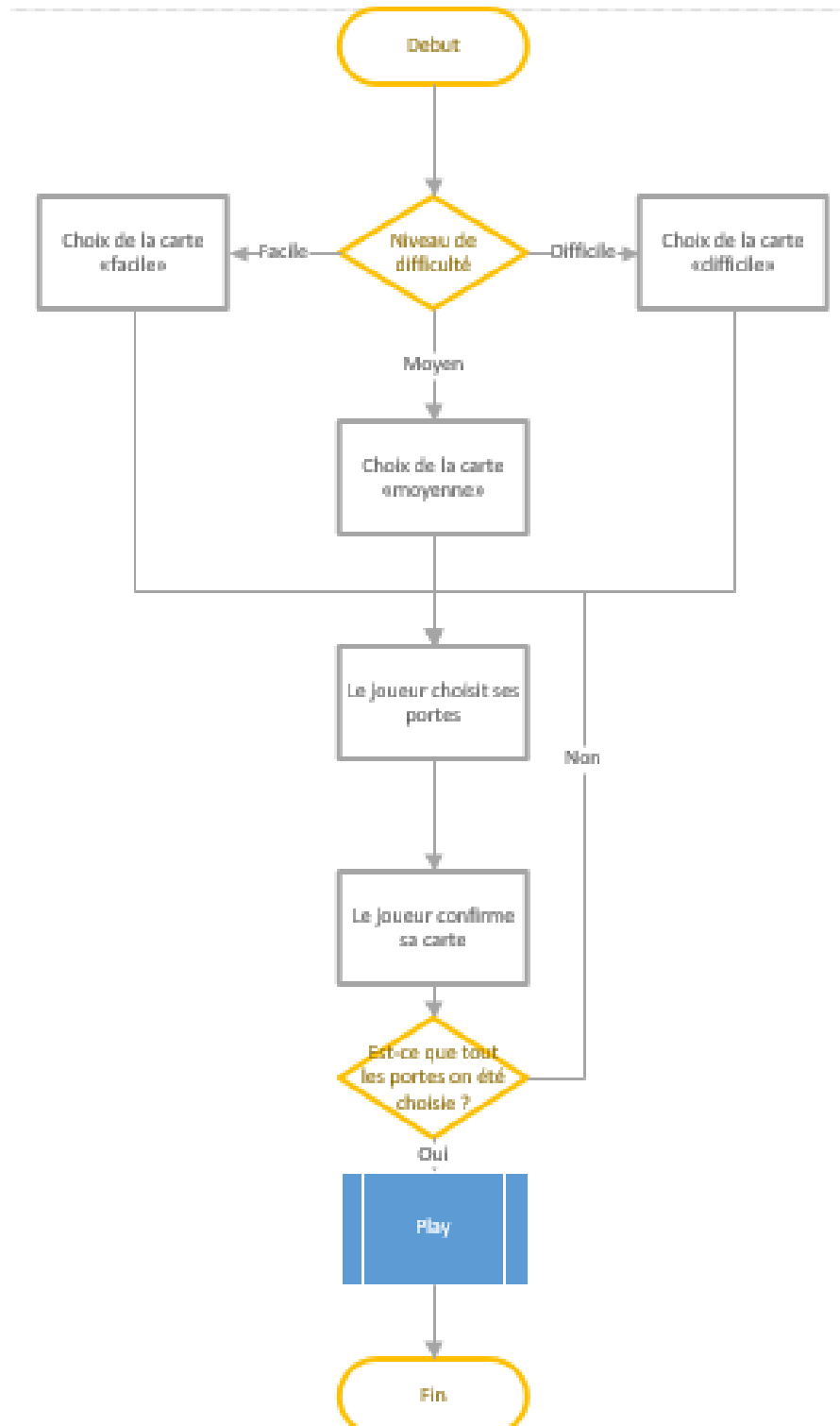
Il peut arriver que le joueur veut quitter le jeu, suivant le cas, si un puzzle est en cours, le jeu demande une confirmation au joueur. Au contraire, si aucun puzzle est en cours, le jeu se quitte sans demander de confirmations



12: Diagramme de flux global de l'application

JEU MANUEL

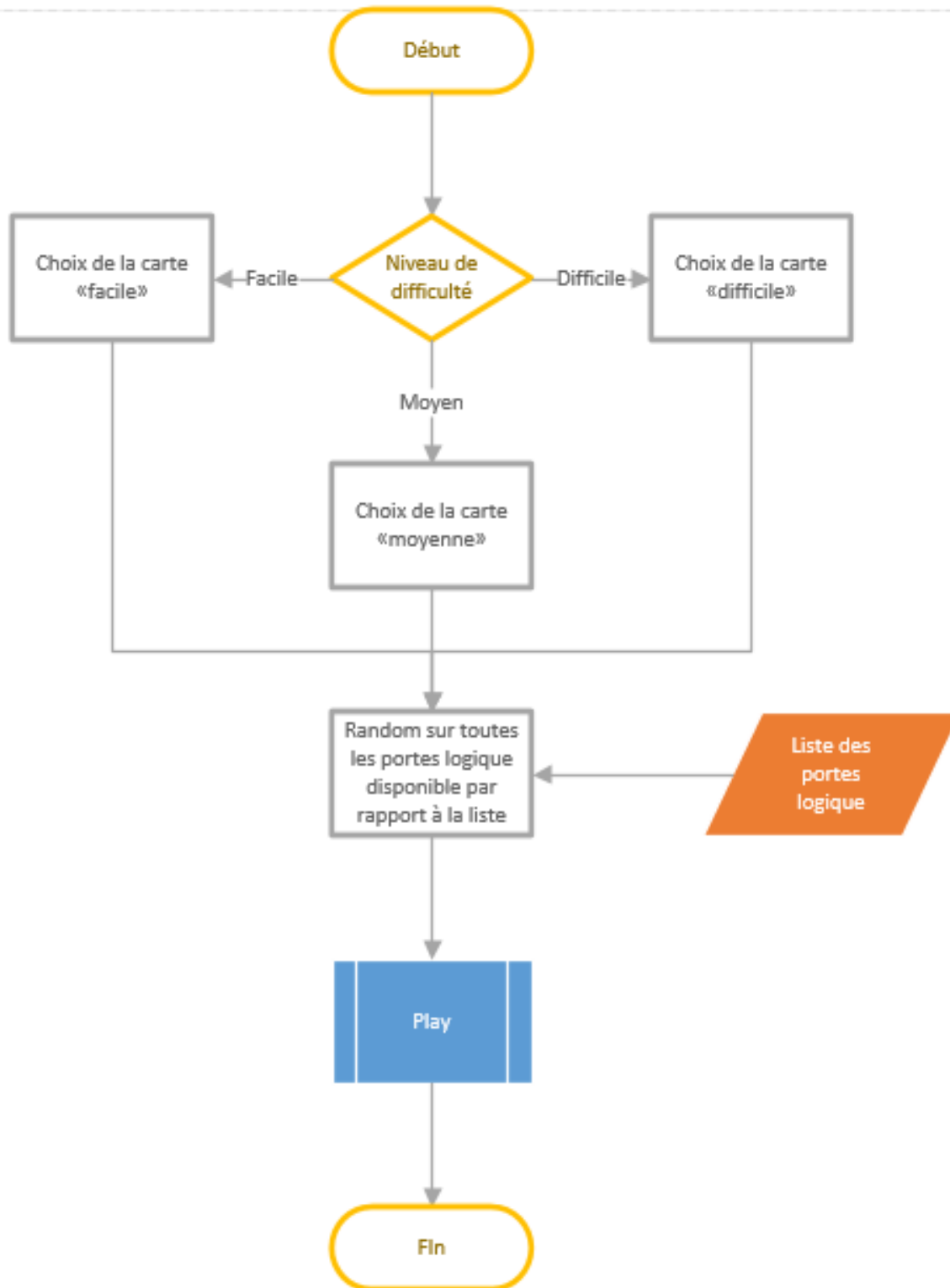
Si le joueur choisit un puzzle manuel, il doit aussi choisir la difficulté du jeu (un puzzle facile a 3 portes, moyen 9 portes et difficile 19 portes), les portes et fils sont déjà présent, le joueur peut juste choisir le type de portes par portes présente, si le joueur ne choisit pas tous ses portes, le jeu ne continue pas, à l'inverse s'il choisit toute ces portes, le jeu continue et appelle la fonction play.



13 Diagramme du puzzle manuel

JEU ALÉATOIRE

Comme pour le puzzle manuel, le puzzle aléatoire commence par un choix de niveau de difficulté, avec le même nombre de portes, le jeu après sélection de la difficulté prend toutes les portes du puzzle, et la liste de porte disponible et fait un random sur toutes les portes, et ensuite le joueur peut enfin participer au puzzle

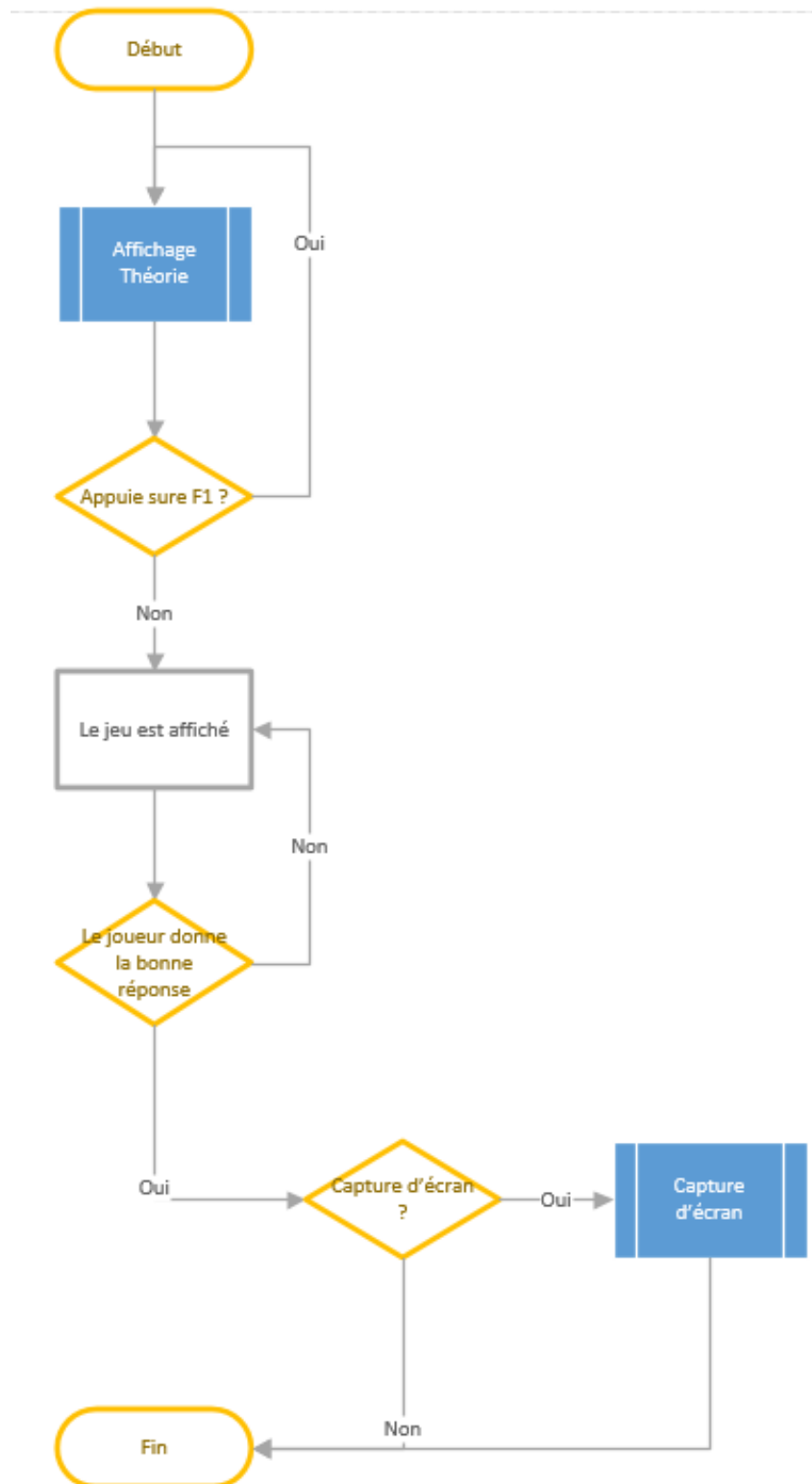


14 Diagramme du puzzle aléatoire

PLAY

Au début de chaque partie, la théorie sur chaque porte présente sur le puzzle courant est affichée, et peut être réaffichée en appuyant sur F1.

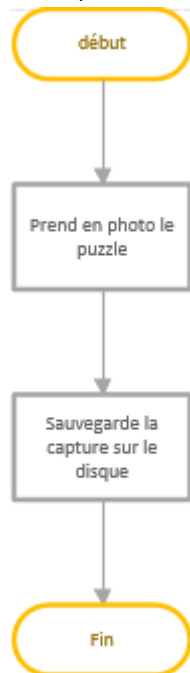
Après, cela, le jeu est affiché, et tant que le joueur n'a pas trouvé la bonne réponse, le joueur ne pourra pas passer à autre chose, s'il trouve la bonne réponse, il peut prendre en photo le puzzle, et le jeu est ensuite considéré comme terminé et retourne au main.



15 diagramme de la résolution de puzzle

CAPTURE D'ÉCRAN

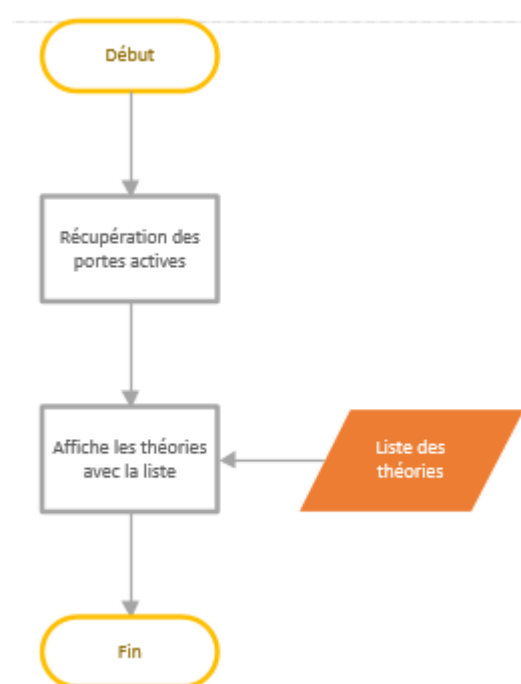
Permet la sauvegarde du puzzle courant par le biais d'une photo de ce dernier



17 Diagramme pour la capture d'écran

THÉORIE DES PORTES LOGIQUES

Permet l'affichage de la théorie des port logiques



16 diagramme pour la théorie des portes logique

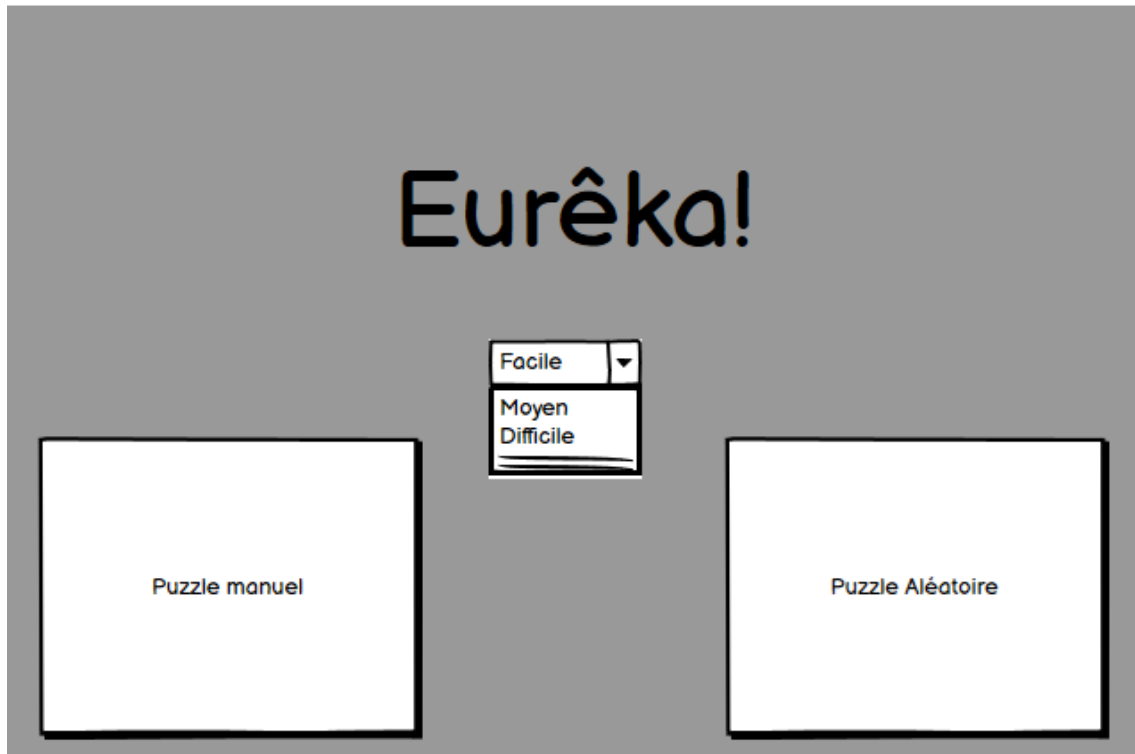
DESIGN DE L'UI

Afin d'avoir une idée de comment l'UI du serious game, va ressembler, il est important d'avoir un design à quoi on peut s'attacher pour plus tard designer l'UI du jeu.

L'UI est pensé pour être accessible par PC et téléphone, les boutons sont grands, facile à voir.

MENU PRINCIPALE/JEU SUIVANT

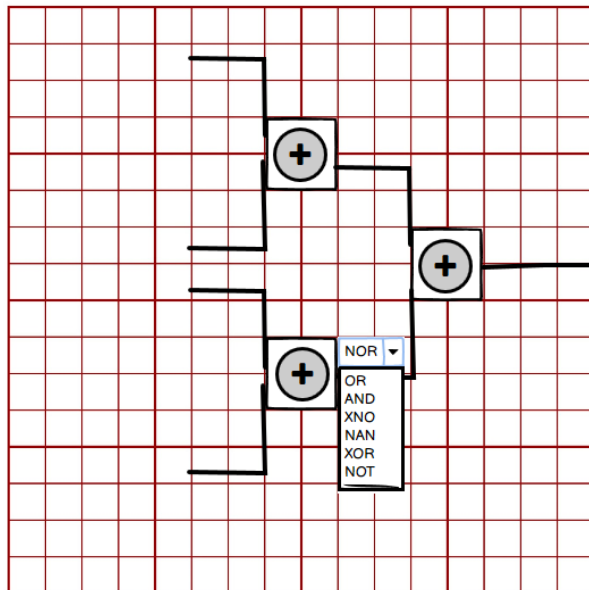
Cette « page » peut être utilisée pour le menu principale, mais aussi comme sélection de mode de jeu pour le prochain puzzle



18 maquette menu principale

MENU DE CRÉATION DE PUZZLE

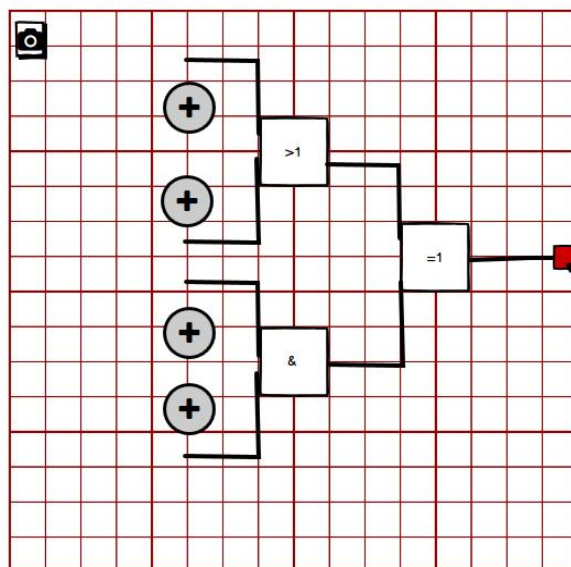
Le joueur aura la possibilité de créer son propre puzzle, les portes sont déjà présents, et le nombre de porte ne change que si le niveau de difficulté change, dans ce cas, le niveau de difficulté est facile, il peut, en cliquant sur la porte, changer de type de porte.



19 maquette : création de puzzle

MENU DE RÉOLUTION DE PUZZLE

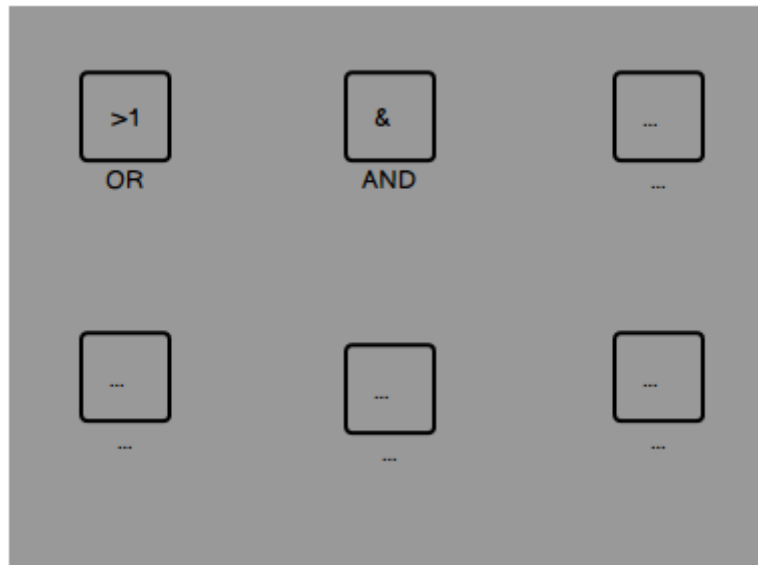
Après avoir terminé la création manuel d'un puzzle ou d'avoir choisi un puzzle aléatoire, le joueur devra compléter le puzzle, si le joueur choisit la bonne réponse, la sortie passe au vert, et le joueur à gagner



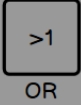
20 maquette : puzzle aléatoire

THÉORIE DES PORTES LOGIQUES

Un choix de portes logique seras affiché, en début de partie et en appuyant sur F1, en appuyant sur les différentes icônes de la première fenêtre, il est possible d'avoir un descriptif détaillé de la porte logique.



21 théorie : jeu d'icônes de porte logique



OR

1	0
0	1
1	0

Lorem ipsum dolor sit amet, consectetur
 adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.
 Ullamcorper velit sed ullamcorper morbi.
 Pellentesque habitant morbi tristique senectus
 et. Tincidunt dui ut ornare lectus sit. Adipiscing
 enim eu turpis egestas pretium aenean
 pharetra. Vulputate dignissim suspendisse in
 est ante in nibh. Commoda quis imperdiet
 massa tincidunt nunc pulvinar. Pellentesque
 nec nam aliquam sem et tortor consequat.
 Auctor urna nunc id cursus metus. Lorem
 ipsum dolor sit amet consectetur adipiscing.
 Amet luctus venenatis lectus magna fringilla
 urna. Neque ornare aenean euismod
 elementum nisi quis eleifend quam adipiscing.
 Montes nascetur ridiculus mus mauris.
 Bibendum enim facilisis gravida neque. Ut sem
 viverra aliquet eget sit amet tellus cras.

22 théorie détaillé des portes logique

STRATÉGIE DE TEST

TEST UNITAIRE

Un test Unitaire seras effectué sur la fonctionnalité de la résolution de puzzle, afin de bien vérifier le bon fonctionnement de cette fonctionnalité, avec le module de tests proposer par visual studio.

TESTS D'ACCEPTATION

Plusieurs tests d'acceptation seront mis sur IceScrum, je demanderais à plusieurs camarades de classe et membres de ma famille pour tester le produit, et de vérifier que toutes les tests d'acceptation passent

RISQUES TECHNIQUES

RISQUE TECHNIQUES

Quelque chose dont je n'ai pas l'habitude de faire, du au faite que j'ai très rarement pratiqué, étant les tests unitaires en C#, est j'en n'ai jamais fait avec tout l'infrastructure d'Unity

DOCUMENTATION

Un risque qui reste majeur pour moi, étant savoir comment documenter toute la partie implémentation mon projet de la bonne manière

RÉALISATION

DOSSIER DE RÉALISATION

TEMPS ACTUEL

Heure	lundi, 2 mai 2022	mardi, 3 mai 2022	mercredi, 4 mai 2022	jeudi, 5 mai 2022	vendredi, 6 mai 2022
08:00					modifications des uses cases et scénario, plus changement des priorité modification des diagrammes de flux
08:45					ajout de descriptif aux images sur les uses cases et scénario, diagrammes de flux
08:50		Création des diagrammes de flux (code pour puzzle aléatoire, puzzle manuel, capture d'écran, affichage théorie, affichage scores) + création du projet unity et git		intégration des diagrammes de flux, théorie des porte, capture d'écran, schéma du design de l'UI	
09:35	Création de la planification initiale				modification des diagrammes de flux modification du journal de travail pour prendre en compte l'avancement de tâche pas terminée, et ajout du numéro de tâches ajout de descriptif aux images sur les uses cases et scénario, diagrammes de flux et maquettes
09:50	Installation de unity 2020	Entretien pour technicien développement		mise à jour du design pour intégrer le niveau de difficulté intégration du schéma du design de l'UI, changement du menu principale, ajout des schémas des puzzle aléatoire et manuel intégration du schéma du design de l'UI, ajout des schémas des puzzle aléatoire et manuel, théorie des	ajout de descriptif aux images sur les diagrammes de flux et maquettes ajout de descriptif aux stories et tâches sur iceScrum Aide à Arthur bourgue
10:35					ajout de descriptifs aux tâches des stories sur icescrum
10:40	création des uses case et scénario (Puzzle aléatoire et puzzle manuelle)	Création du design de l'UI (puzzle aléatoire et puzzle manuel)			
11:25					Sprint review avec M. Viret
11:30		Recherches sur les normes des porte logiques (IEC) Création du design de l'UI (main, prise de screenshot)		ajout des tâches sur icescrum pour toutes les stories création de sprites pour le jeu (porte logiques, câbles, effets)	ajout de descriptions sur certaines tâches qui en manque, ajout de la durée des tâches, changement du format de cellule dans le journal de travail
12:15					

13:30	Création du rapport de projet	Réalisation de l'interface graphique avec la théorie des portes mise à jour du rapport de projet			ajout de légendes, changement, modifications du style, mise en forme
14:15					
14:20		Création du projet sur IceScrum + mise en place des sprints et création des stories + mise à jour du rapport de projet			Aide à Arthur bourgue Changement du des maquette, afin que la théorie des portes logique soit plus facile à lire sur téléphone intégration des nouvelles maquettes dans la documentation, mise en page plus correcte
15:05	création des uses case et scénario (Résolution de puzzle, capture d'écran, théorie des porte logiques)				
15:20	Création des diagrammes de flux (code main)	mise à jour du rapport de projet			ajout d'une stories pour l'ui dans ice scrum, avec des tâches
16:05					création de l'ui du menu démarré sur Unity
16:10					création de l'ui du menu démarré sur Unity ajout de tâches sur ice scrum pour l'ui, création de plusieurs tâches pour ajout de scènes, intégration de boutons
16:55					recherches sur les conventions de nommage sur c# création du script main menu, qui gère les boutons, et la liste déroulante du menu démarré

Comme pour la planification initiale

SCRIPTS

CLASSES

LogicGate

+LogicGateType : enum
+type : LogicGateType

23 Diagramme de la classe LogicGate

- LogicGate : une classe contenant des types de porte logique

GATES

GateManager

+output : bool
+logicGate : LogicGate
-text : Text
-GateSprites : List<Sprite>
-InversedGateSprites : List<Sprite>
-ColorState : List<Sprite>
-ImageInversed : Image
-CurrentSate : Image
-input1 : bool
-input2 : bool
-Start
+ChangeInput1(bool value)
+ChangeInput2(bool value)
-changeOutput()

CustomGateManager

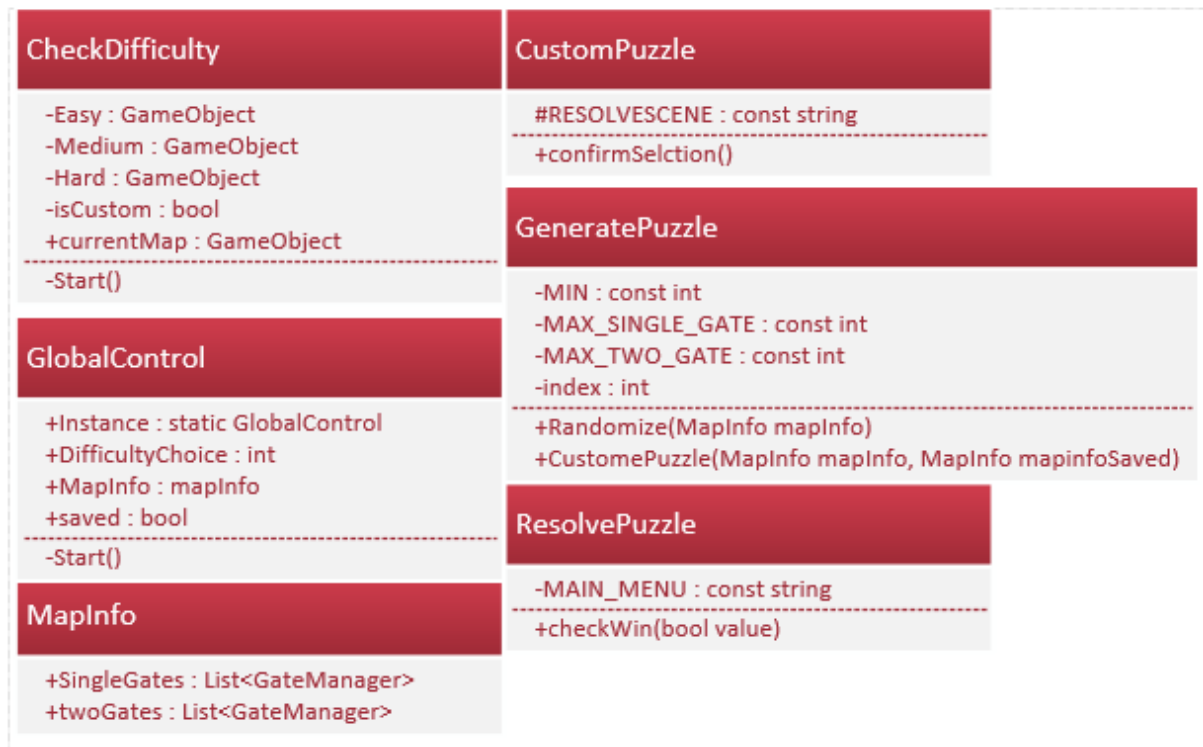
-dropdown : Dropdown
-background : Image
-GateSprites : List<Sprite>
-InversedGateSprites : List<Sprite>
+isSingleGate : bool
-gateManager : GateManager
+OnSelectGate(int value)

24 Diagramme des classe GateManager et CustomGateManager

- GateManager : le script gérant une porte logique, chaque porte logique à se script, il gère entrées et sorties

-
- CustomGateManager : le script gérant la création des portes logique dans la création de puzzle

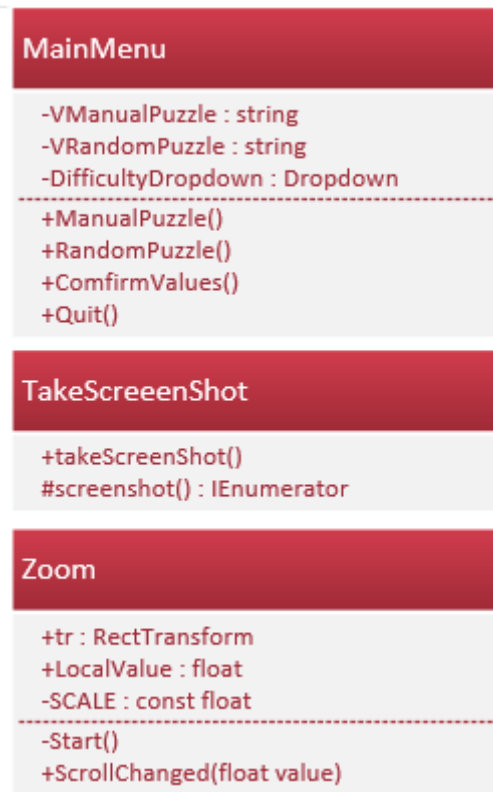
PUZZLE



25 Diagramme des classe pour la gestion de puzzle

- CheckDifficulty : script qui prends en compte la difficulté choisit par le joueur, et choisit la bonne carte
- GlobalControl : script qui permet de garder en mémoire des variables, pour qu'elles ne soient pas perdue au changement de scène
- MapInfo : script qui donne les infos, de toutes les portes logique courante pour la difficulté choisie
- GeneratePuzzle : récupère toutes les portes logiques de la difficulté choisie et change toutes les port (sauf pour moyen et difficile ou la dernière porte est de tout de façon un AND), si le joueur a créer un puzzle auparavant, il génère le puzzle par rapport au choix du joueur
- ResolvePuzzle : Script qui gère la complétion du puzzle, si la dernière porte est en « Vert », le jeu retourne au menu principale, et il est possible de rechoisir un nouveau puzzle.
- CustomPuzzle : gère la confirmation du choix du joueur pour la création du puzzle, et sauvegarde le tout dans GlobalControl

UI



26 diagramme des classe de l'UI

- MainMenu : script qui gère le menu démarré, notamment le changement de scène et la difficulté
- Zoom : script qui permet de zoomer sur le puzzle
- TakeScreenShot : permet la capture d'écran du puzzle actuel, et sauvegarde le sur le disque

DOSSIER

- Assets : répertoire principal, englobent tous les autres dossiers, fichier et script

RESSOURCES

- GameObjects : répertoire contenant différents gameobjects utiles pour le projet.
- Prefabs : répertoire contenant les préfab utilisés pour le projet
- Sprites : répertoire contenant tous les sprites du jeu

SCENES

- Répertoire contenant toutes les scènes du jeu

SCRIPTS

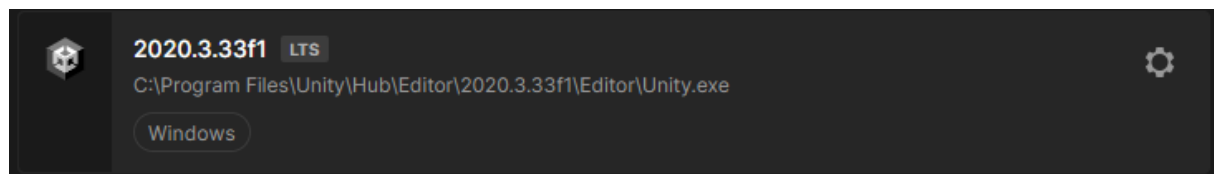
- Classes : répertoire contenant toutes les classes utilisées dans le jeu, notamment LogicGate
- Gates : répertoire avec les scripts gérant les portes logiques
- Puzzle : répertoire avec les scripts gérant les puzzles
- UI : répertoire contenant les scripts qui gèrent les UI

VERSION DE WINDOWS



27 Version de windows

VERSION DE UNITY



28 Version de Unity

VERSION DE VISUAL STUDIO

À propos de Microsoft Visual Studio

Visual Studio

État de la licence
Termes du contrat de licence

Microsoft Visual Studio Community 2019
Version 16.11.10
© 2021 Microsoft Corporation.
Tous droits réservés.

Microsoft .NET Framework
Version 4.8.04084
© 2021 Microsoft Corporation.
Tous droits réservés.

Produits installés :

Gestionnaire de package NuGet – 5.11.0

Outils C# – 3.11.0-4.21602.3+fc14a1355c0461af5110b74b26f6478f22d26565

Outils Visual Basic – 3.11.0-4.21602.3+fc14a1355c0461af5110b74b26f6478f22d26565

Package hôte de l'adaptateur de débogage de Visual Studio Code – 1.0

ProjectServicesPackage Extension – 1.0

Visual Studio Tools pour Unity – 4.11.4.0

Copier les informations

Informations système

DxDiag

Détails sur le produit :

Gestionnaire de package NuGet dans Visual Studio. Pour plus d'informations sur NuGet, visitez <https://docs.nuget.org/>

Avertissement : ce logiciel est protégé par la loi relative au droit d'auteur et par les conventions internationales. Toute reproduction ou distribution partielle ou totale de ce logiciel, par quelque moyen que ce soit, est strictement interdite. Toute personne ne respectant pas ces dispositions se rendra coupable du délit de contrefaçon et sera passible des sanctions pénales prévues par la loi.

OK

29 Version de Visual Studio

VERSION DU PROGRAMME

2.00

MATÉRIEL UTILISÉ

PROCESSEUR

Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41 GHz

RAM

16.0 Go

OS

Windows 10 Éducation 21H2 x64

LIBRAIRIES

DANS LE PROJET

- 2D pixel perfect
- 2D Sprite
- JetBrains Rider Editor
- Test Framework
- TextMeshPro
- Timeline
- Unity UI
- Version Control
- Visual Studio Code Editor

- Visual Studio Editor

LIBRAIRIES BUILT-IN

- AI
- Android JNI
- Animation
- Asset Bundle
- Audio
- Cloth
- Director
- Image Conversion
- IMGUI
- JSONSerialize
- Particle System
- Physics
- Physics 2D
- Screen Capture
- Subsystems
- Terrain
- Terrain Physics
- Tilemap
- UI
- UIElements
- UIElements Native
- Umbra
- Unity Analytics
- Unity Web Request
- Unity Web Request Asset Bundle
- Unity Web Request Audio
- Unity Web Request Texture
- Unity Web Request WWW
- Vehicles
- Video
- VR
- Wind
- XR
- les répertoires où le logiciel est installé
- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels
- la description exacte du matériel
- le numéro de version de votre produit !
- Programmation et scripts : librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

DESCRIPTION DES TESTS EFFECTUÉS

TESTS D'ACCEPTATIONS

RÉSOLUTION DU PUZZLE

- Les portes logiques fonctionnent correctement
 - Quand le joueur change l'entrée des portes, la sortie des portes logique réagisse correctement
- Les entrées changent quand le joueur veut
 - Si le joueur change l'entrée, le changement est pris en compte et affecte les portes logiques
- Le système détecte quand le puzzle est terminé
 - Quand le puzzle est réussi, et que la dernière porte logique est en vert, le jeu se termine correctement

CRÉATION DU PUZZLE ALÉATOIRE

- Le système change aléatoirement toutes les portes
 - Le système génère aléatoire les portes logiques, en prenant en compte les portes à 2 entrées et 1 entrée
- Le système génère aléatoirement à chaque fois
 - Le système génère aléatoirement, et ne régénère jamais le même puzzle

CRÉATION DU PUZZLE CUSTOMISÉ

- Quand le joueur choisit sa porte logique la porte change
 - Le visuel change, et le système prend en compte
- Quand le joueur confirme son choix le système prend en compte le choix
 - Quand le joueur a terminé de créer son puzzle, et confirme son choix, le système passe en mode création, avec les portes logique à la bonne place, qui fonctionne correctement

CRÉATION DU SYSTÈME DES THÉORIE

- Les boutons réagissent correctement
 - Les boutons affichent et caches les bonnes dalles

SYSTÈME DE CAPTURE D'ÉCRAN

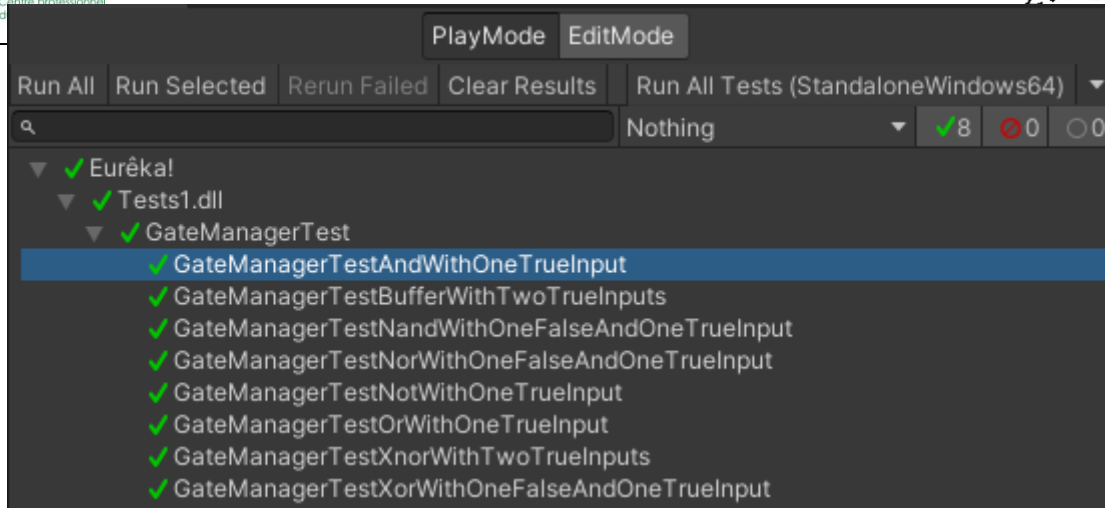
- Quand le bouton est pressé une photo est prise
 - Une photo est prise au moment du pressage du bouton, et le l'image est bien capturée
- Si pas de dossier existe, un dossier est créé
 - Si le joueur prend une photo et le dossier de photo n'est pas présent, le dossier est créé

TEST UNITAIRE

Le test unitaire se porte sur le script de la gestion de porte logique, c'est-à-dire, le script, qui gère les entrées, et sortie des portes, le test est réalisé en C# sur Visual Studio.

GATEMANAGER TEST AND WITH ONE TRUE INPUTS

Test sur une porte AND avec une entrée TRUE, et qui retourne un FALSE, les entrées qui ne sont pas spécifiés, sont de base false



GATEMANAGER TEST BUFFER WITH TWO TRUE INPUTS

Test sur une porte Buffer, avec deux entrées TRUE, et qui retourne un TRUE, les portes logiques avec une seule entrée, ne prennent en considération seulement que la première entrée

GATEMANAGER TEST NAND WITH ONE FALSE AND ONE TRUE INPUT

Test sur une porte NAND, avec une entrée TRUE et une entrée FALSE, et qui retourne un FALSE

GATEMANAGER TEST NOR WITH ONE FALSE INPUT AND ONE TRUE INPUT

Test sur une porte NOR, avec une entrée FALSE et une entrée TRUE, qui retourne un FALSE

GATEMANAGER TEST NOT WITH ONE TRUE INPUT

Test sur une porte NOT, avec une entrée TRUE, qui retourne un false

GATEMANAGER TEST OR WITH ONE TRUE INPUT

Test sur une porte OR, avec une entrée TRUE, qui retourne un TRUE

GATEMANAGER TEST XNOR WITH TWO TRUE INPUTS

Test sur une porte logique XNOR, avec deux entrée TRUE, qui retourne un TRUE

GATEMANAGER TEST XOR WITH ONE FASLE AND ONE TRUE INPUT

Test sur une porte logique XOR, avec une entrée TRUE, et une entrée FALSE, qui retourne un TRUE

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

ERREURS RESTANTES

S'il reste encore des erreurs:

- Description détaillée
- Conséquences sur l'utilisation du produit
- Actions envisagées ou possibles

LISTE DES DOCUMENTS FOURNIS

- Diagrammes de flux
- Journal de travail
- Maquettes
- Uses case et scénario
- Résumé du projet

CONCLUSIONS

- Développez en tous cas les points suivants:
-
- Objectifs atteints / non-atteints
- Points positifs / négatifs
- Difficultés particulières
- Suites possibles pour le projet (évolutions & améliorations)

ANNEXES

RÉSUMÉ DU RAPPORT DU TPI / VERSION SUCCINCTE DE LA DOCUMENTATION

SOURCES – BIBLIOGRAPHIE

- tutorialspoint. «Logic gates.» *tutorialspoint.* 16 05 2022.
https://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm.
- Unity. «Screenshot, but from a certain camera.» *Unity Forum.* 16 05 2022.
<https://forum.unity.com/threads/screenshot-but-from-a-certain-camera.69645/#:~:text=%20%20%201%20Disable%20all%20activated%20cameras,is%20done%2C%20you%20can%20switch%20cameras...%20More%20>.
- . «Unity - Application.persistentDataPath.» *Unity documentation.* 16 05 2022.
<https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>.
- . «Unity - ScreenCapture.CaptureScreenshot.» *Unity Documentation.* 16 05 2022.
<https://docs.unity3d.com/ScriptReference/ScreenCapture.CaptureScreenshot.html>.

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

JOURNAL DE TRAVAIL

Date	Durée	Activité	Remarques

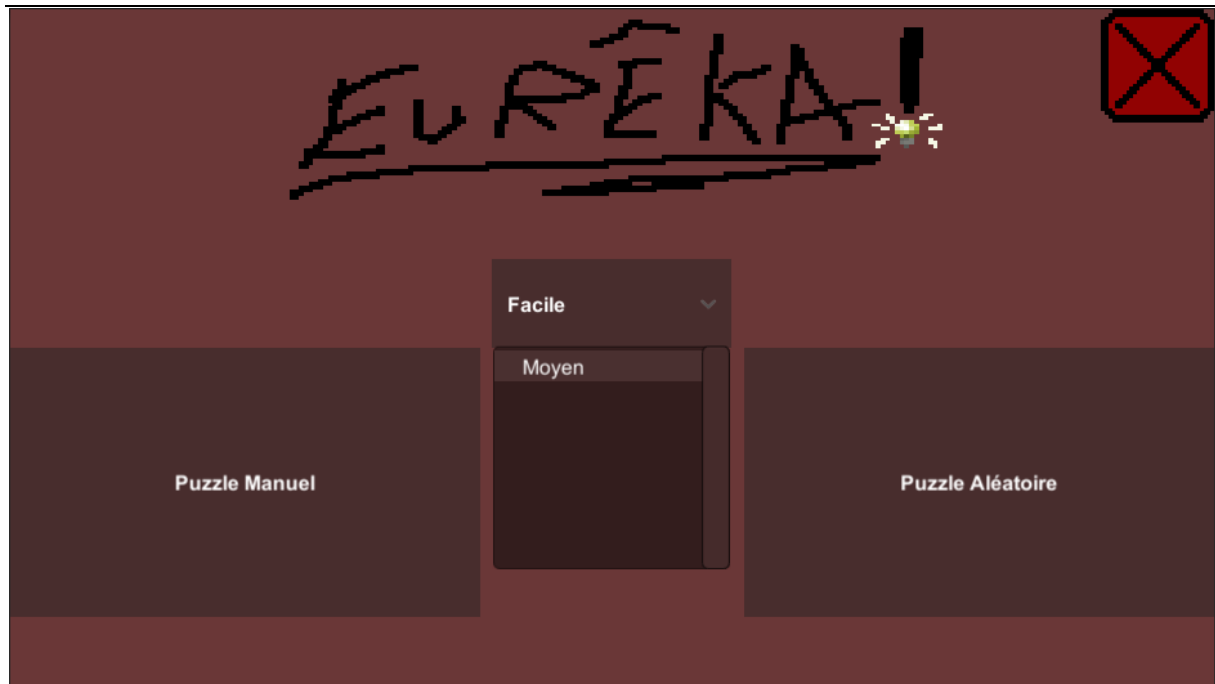
MANUEL D'INSTALLATION

Sous [tristepin222/Eureka: projet TPI \(github.com\)](https://github.com/tristepin222/Eureka), installez la dernière release disponible, à faire attention : le programme est compatible seulement sur les version Windows 10.

Après avoir télécharger le jeu, feuillez le désarchiver, avec un programme comme winrar, et simplement lancer le jeu, en cliquant sur l'icône se terminant par EXE.

MANUEL D'UTILISATION

Après avoir, installer le jeu, et d'avoir lancé le jeu pour la première fois, vous trouverez sur le menu principal, ou plusieurs choix se présente à vous, le choix de difficulté, puzzle manuel ou puzzle aléatoire, le choix de difficulté est par défaut, facile.



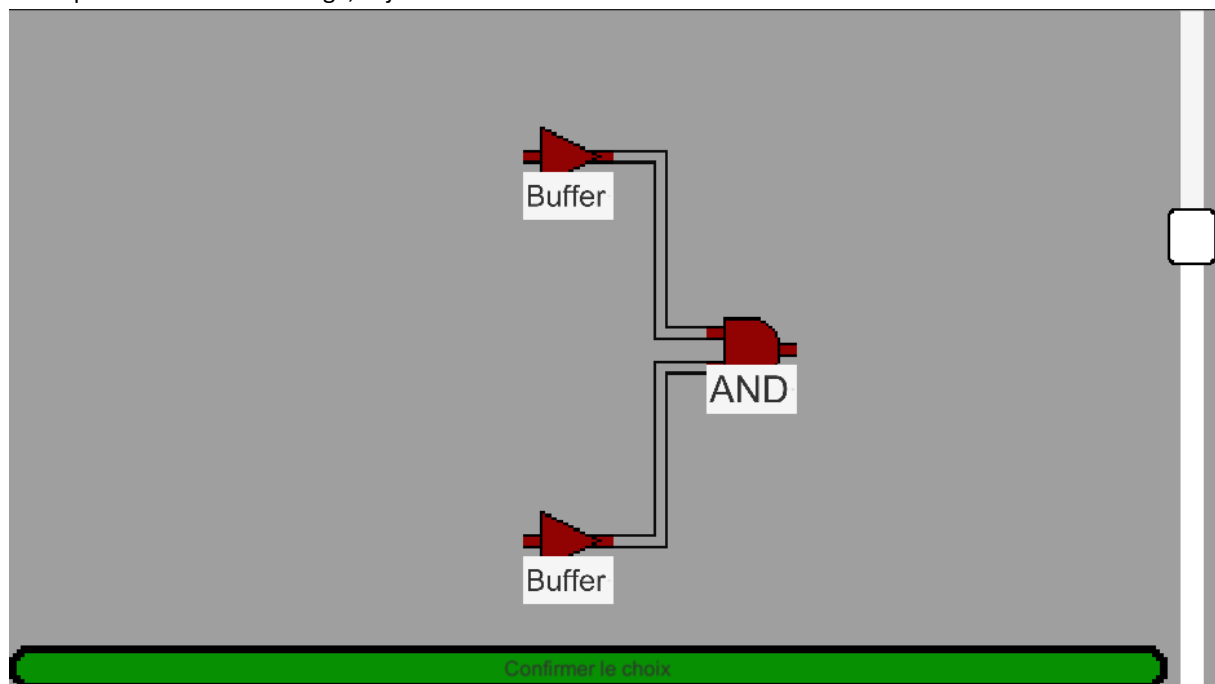
31 Menu principale

Le bouton « puzzle manuel », vous permet de créer votre propre puzzle.

La liste déroulante, vous permet de choisir la difficulté.

Le bouton « puzzle » aléatoire, vous permet de générer un puzzle aléatoirement.

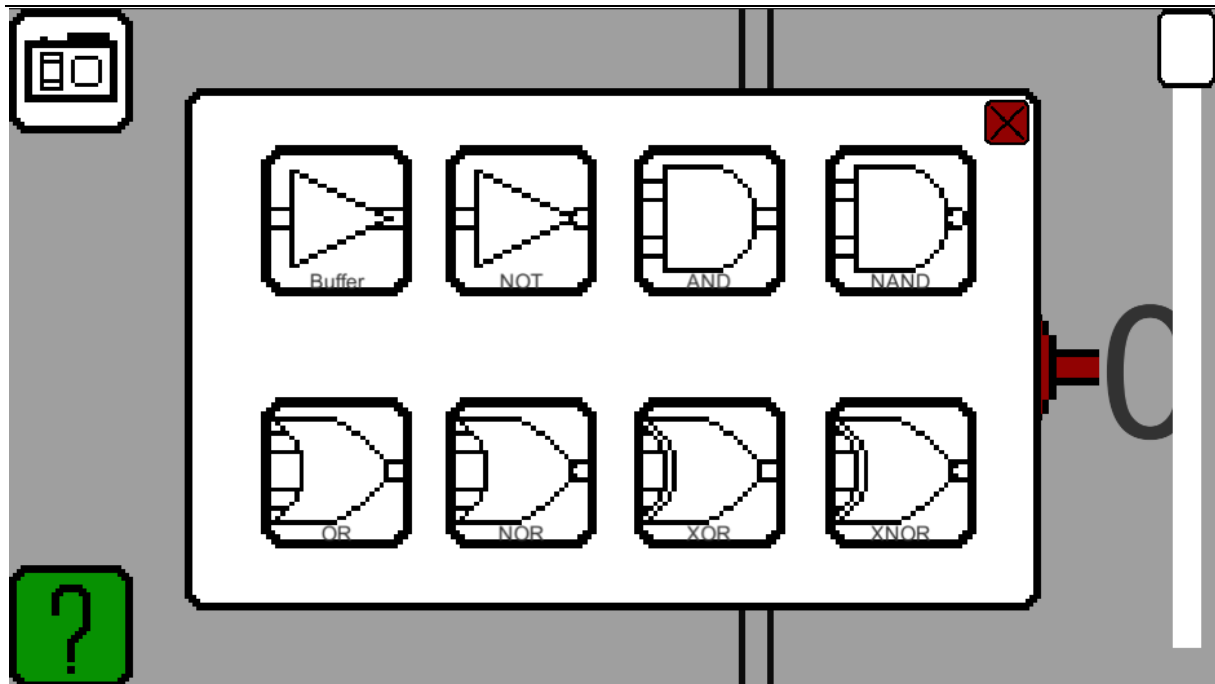
En cliquant sur le bouton rouge, le jeu se ferme



32 Système de création de puzzle

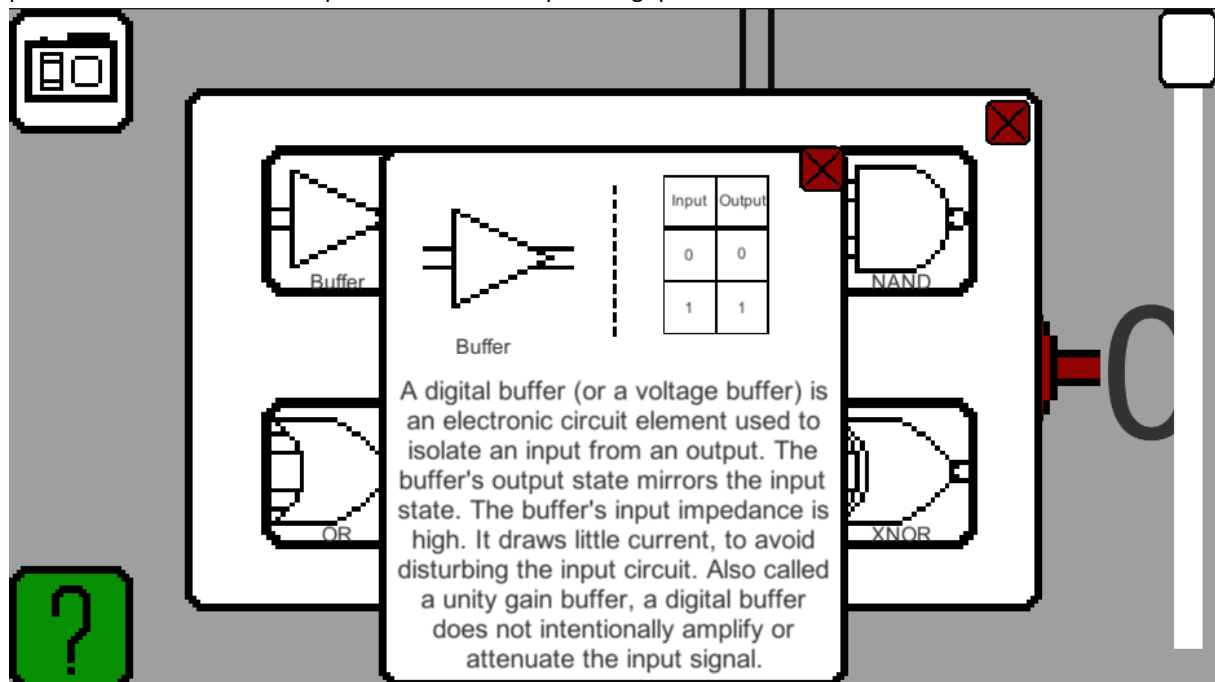
Les listes déroulantes en dessous des portes logiques, permettent de choisir la porte de votre choix, et en cliquant sur « confirmer le choix », votre choix est sauvegardé, et vous passez en mode résolution de puzzle.

Il y a aussi possibilité de zoomer et dézoomer, avec le slider à droite.



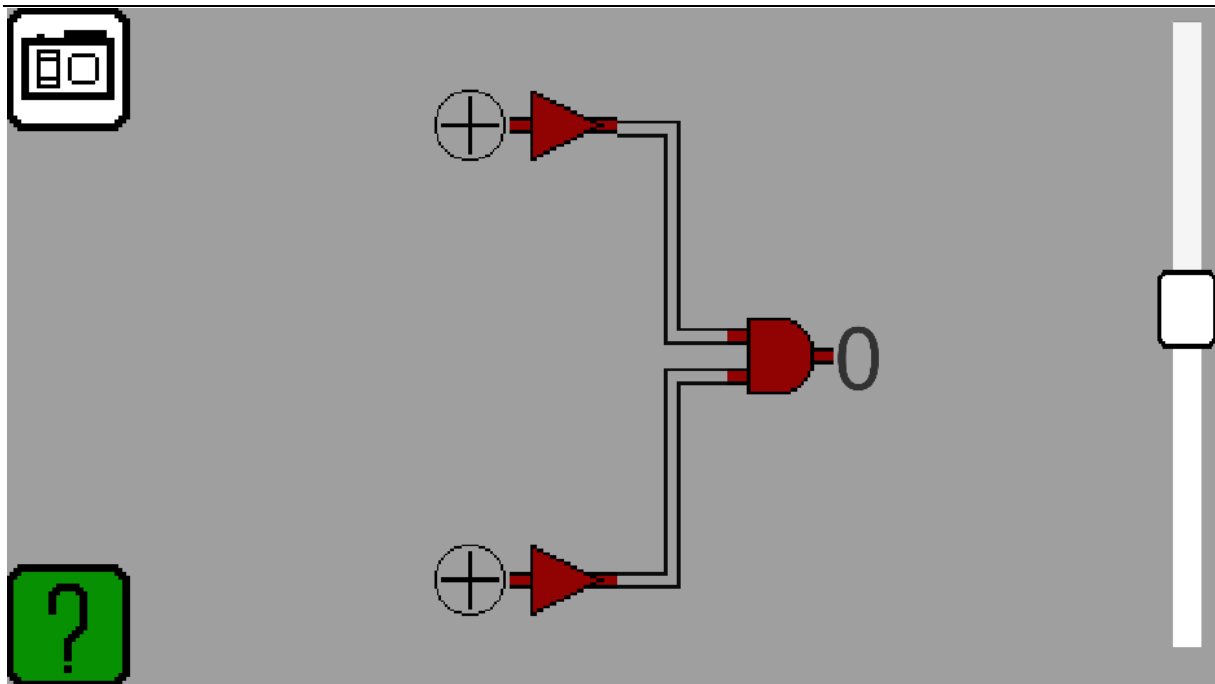
33 Boutons théorie des portes

Vous êtes présenté par une UI, avec plusieurs portes logiques, en appuyant sur n'importe qu'elle bouton, il est possible d'afficher une description détaillée de la porte logique.



34 Théorie détaillée des portes logiques

Il est possible de fermer ces fenêtres en appuyant sur la croix rouge



35 Résolution de puzzle, aléatoire ou manuel

En appuyant sur le point d'interrogation, il est possible de réafficher la page avec les portes logique et ses descriptions.

En appuyant sur le bouton avec l'appareil photo, il est possible de prendre un screenshot du puzzle courant, avec les boutons à droite des portes logique, il est possible de changer l'entrée, si la sortie, finale est en verte, le puzzle est gagné, et vous retournerez au menu principale. Il y a aussi possibilité de zoomer et dézoomer, avec le slider à droite.

ARCHIVES DU PROJET

VERSION DE LA DOCUMENTATION

- 1.00
 - Canevas de base
- 2.00
-
- 3.00
- 4.00
- 5.00

Media, ... dans une fourre en plastique

TABLE DES ILLUSTRATIONS

1 PLANIFICATION INITIALE	6
2 PLANIFICATION DÉTAILLÉE SEMAINE 1	6
3 PLANIFICATION DÉTAILLÉE SEMAINE 2	7
4 PLANIFICATION DÉTAILLÉE SEMAINE 3	7
5 PLANIFICATION DÉTAILLÉE SEMAINE 4	8
6 PLANIFICATION DÉTAILLÉE SEMAINE 5	8
7 CAS D'UTILISATION : PUZZLE ALÉATOIRE	9
8 CAS D'UTILISATION : PUZZLE MANUEL.....	10
9 CAS D'UTILISATION : RÉOLUTION DE PUZZLE	10
10 CASE D'UTILISATION : CAPTURE D'ÉCRAN.....	10
11 CAS D'UTILISATION : THÉORIE DES PORTE LOGIQUE	11
12: DIAGRAMME DE FLUX GLOBAL DE L'APPLICATION.....	11
13 DIAGRAMME DU PUZZLE MANUEL	12
14 DIAGRAMME DU PUZZLE ALÉATOIRE.....	13
15 DIAGRAMME DE LA RÉOLUTION DE PUZZLE	14
16 DIAGRAMME POUR LA THÉORIE DES PORTES LOGIQUE.....	15
17 DIAGRAMME POUR LA CAPTURE D'ÉCRAN.....	15
18 MAQUETTE MENU PRINCIPALE	16
19 MAQUETTE : CRÉATION DE PUZZLE	17
20 MAQUETTE : PUZZLE ALÉATOIRE	17
21 THÉORIE : JEU D'ICÔNES DE PORTE LOGIQUE.....	18
22 THÉORIE DÉTAILLÉ DES PORTES LOGIQUE	18
23 DIAGRAMME DE LA CLASSE LOGICGATE	20
24 DIAGRAMME DES CLASSE GATEMANAGER ET CUSTOMGATEMANAGER.....	20
25 DIAGRAMME DES CLASSE POUR LA GESTION DE PUZZLE.....	22
26 DRIAGRAMME DES CLASSE DE L'UI	23
27 VERSION DE WINDOWS	24
28 VERSION DE UNITY	24
29 VERSION DE VISUAL STUDIO	25
30 TEST UNITAIRES.....	27
31 MENU PRINCIPALE	31
32 SYSTÈME DE CRÉATION DE PUZZLE.....	31
33 BOUTONS THÉORIE DES PORTES.....	32
34 THÉORIE DÉTAILLÉE DES PORTES LOGIQUES	32
35 RÉOLUTION DE PUZZLE, ALÉATOIRE OU MANUEL	33