

IA DE JEU :

DEBLAECKER JEREMY B3 INFO LOGICIEL

Table des matières

Description du projet :.....	2
Clients, rôles de chacun :.....	2
Planning :.....	2
Technologies utilisées :	3
Fonctionnement de l'algorithme :.....	4
Fonctionnalités majeures	5
Maquettes :	6
Besoins fonctionnels	6
Utilisateurs du projet.....	6
Diagramme de classe.....	7
Diagramme de séquence.....	8
Captures écran	9
Annexe :.....	14

Description du projet :

L'objectif principal de ce projet est de développer un jeu vidéo de type "Labyrinthe" qui permettra à l'utilisateur de créer des niveaux, et voir comment l'IA du jeu trouve son chemin. Pour ce faire je vais me servir de l'algorithme de Dijkstra qui permet de résoudre le problème du plus court chemin.

Le projet comporte deux niveaux prédéterminés afin que l'utilisateur puisse tester le jeu sans avoir à créer son propre labyrinthe. Ce jeu contient également un éditeur de niveau qui permet de modifier un niveau préexistant ou de créer son propre niveau. Le labyrinthe peut ensuite être parcouru par une IA qui

Clients, rôles de chacun :

Pour ce projet de développement logiciel je travaille seul, le sujet fut proposé par Ynov parmi la liste des logiciels que nous pouvions choisir. Ce projet va me permettre d'améliorer mes compétences dans l'utilisation des algorithmes ainsi que sur le développement de jeu en python.

Planning :

Date	Jalon
30/01/21 :	Mise en place de l'environnement de dev.
15/02/21 :	Mise en place du labyrinthe et du premier niveau.
30/02/21 :	Création des obstacles.
05/03/21 :	Rédaction de la documentation.
29/03/21 :	Oral intermédiaire.
05/04/21 :	Création du deuxième niveau.
10/04/21 :	Lier le jeu et la BDD.
20/04/21 :	Création du logiciel de test de l'IA.
10/05/21 :	Création de l'éditeur de niveau.

15/05/21 :	Vérification de la fonctionnalité du projet.
15/05/21 :	Finaliser la documentation.
17/05/21 :	Oral final.

Tableau 1: Planning prévisionnel

Le planning initial prévoyait que je me charge de développement les highscores avant l'éditeur de niveau cependant j'ai préféré commencer par la création de l'éditeur de niveau qui me paraissait plus important pour ce projet.

Date	Jalon
30/01/21 :	Mise en place de l'environnement de dev.
15/02/21 :	Mise en place du labyrinthe et du premier niveau.
05/03/21 :	Rédaction de la documentation.
29/03/21 :	Oral intermédiaire.
05/04/21 :	Création du deuxième niveau.
10/04/21 :	Création de l'éditeur de niveau.
20/04/21 :	Création du logiciel de test de l'IA.
10/05/21 :	Intégration et test de l'éditeur de niveau.
12/05/21 :	Lier le jeu et la BDD.
15/05/21 :	Finaliser la documentation.
15/05/21 :	Vérification de la fonctionnalité du projet.
17/05/21 :	Oral final.

Tableau 2: Planning modifié après l'oral intermédiaire

Technologies utilisées :

Pour le développement de ce projet j'utilise Python grâce à l'IDE PyCharm et Visual Studio Code. Je me sers également de la librairie tinker afin de créer l'interface utilisateur.

Pour la création de maquette j'utilise AdobeXD qui me permet de créer l'interface graphique de mon application.

Concernant le gestionnaire de version je me sers d'un dépôt GitHub que vous trouverez à l'adresse suivante : https://github.com/jeremydeblaecker/IA_Jeu

Pour le suivi des tâches de ce projet j'utilise Trello qui me permet de savoir qu'elle tâche il me reste à faire. <https://trello.com/b/1C9QICSW/ia-de-jeu>

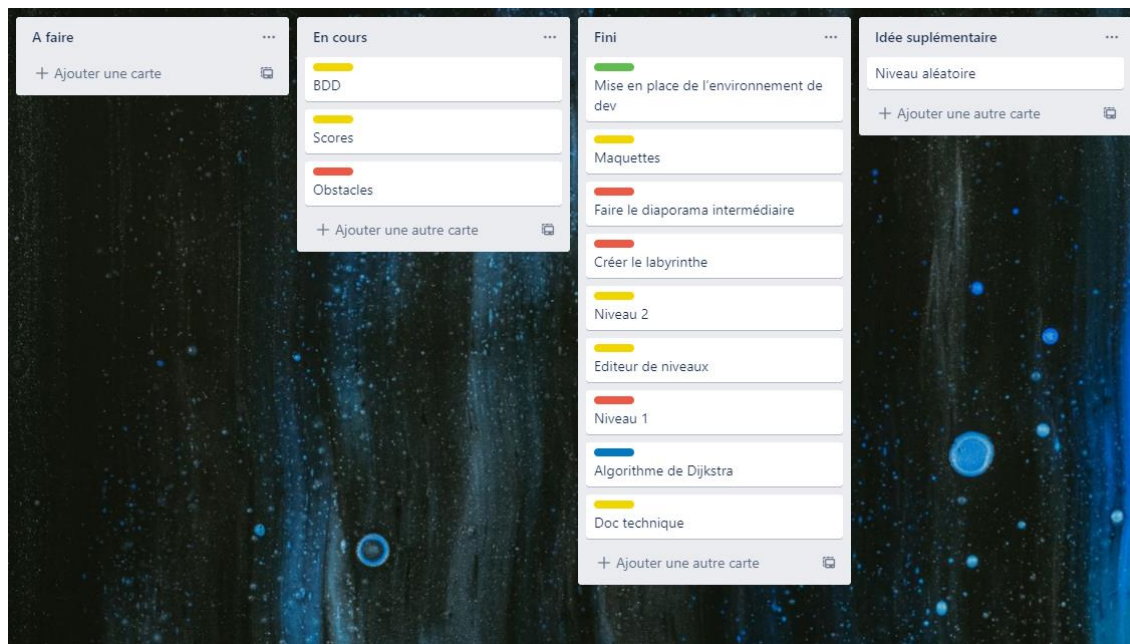


Figure 1: Trello du projet

Fonctionnement de l'algorithme :

Le problème du plus court chemin consiste à trouver le parcours le plus rapide entre un point A et un point B dans un graphe (ici notre labyrinthe) de manière optimiser, c'est-à-dire sans avoir besoin de repasser par un même endroit.

Pour ce faire j'ai décidé de développer ce jeu d'utiliser l'algorithme de Dijkstra qui permet de trouver le plus court chemin entre deux sommets d'un graphe.

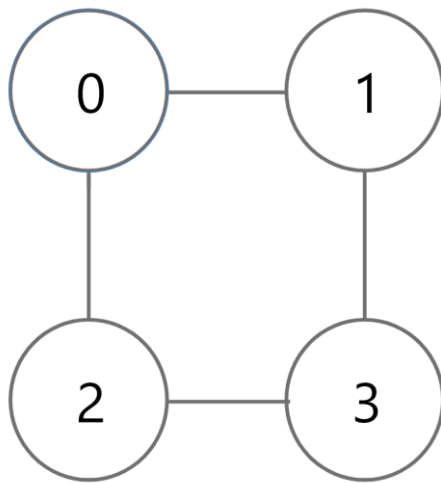
Un graphe est composé de deux éléments notables :

- Un ensemble de nœuds (en anglais « vertices ») aussi appelé sommets (voir les cercles numérotés de l'image, figure 2)
- Un ensemble d'arrêtes/nœuds (en anglais « edges »), qui relient les sections entre elles.

Afin de représenter notre graphe de manière compréhensible pour notre ordinateur nous devons nous servir de la représentation des matrices adjacentes, c'est-à-dire un tableau de nombre

L'algorithme de Dijkstra se base sur les matrices adjacentes par exemple (voir le graphique et le tableau ci-dessous), on peut voir que le sommet 0 est adjacent aux sommets 1 et 2 nous pouvons donc rentrer ces informations dans notre matrice (figure 1).

Pour ce projet j'ai donc décidé de créer un labyrinthe de 20 cases par 20 cases (400 sommets). Chacun de ces sommets se trouvent à la même distance les uns des autres, chaque sommet à le même poids ce qui permet ensuite à l'algorithme de trouver le chemin le plus court entre chacun de ces points.



	0	1	2	3
0	0	1	1	0
1	1	0	0	1
2	1	0	0	1
3	0	1	1	0

Tableau 3: Matrice du graphe à quatre nœuds

Figure 2: Graphe à quatre nœuds

Fonctionnalités majeures

Afin que ce projet soit fonctionnel il est nécessaire de développer plusieurs fonctionnalités majeures listé ci-dessous.

Menu :

Le menu permet à l'utilisateur de sélectionner le niveau, les obstacles, le point de départ et la destination.

Deux niveaux prédéterminés :

Les deux niveaux prédéterminés permettent d'avoir un labyrinthe déjà prérempli d'obstacles, il suffit ensuite de rajouter le point de départ et d'arriver grâce au menu.

Une IA qui parcourt le labyrinthe :

Une fois le labyrinthe réalisé avec des obstacles, un point de départ et d'arriver nous pouvons ensuite utiliser l'IA qui trouvera le chemin le plus court jusqu'à la fin du jeu.

Un éditeur de niveau :

L'éditeur de niveau permet quant à lui de modifier un niveau préexistant en y ajoutant des obstacles, nous pouvons également utiliser la grille de 400 cases vierges afin de créer notre propre labyrinthe.

Scores :

La page de score quant à elle permet à l'utilisateur de voir en combien de temps et de coups l'IA a pu parcourir le labyrinthe.

Maquettes :

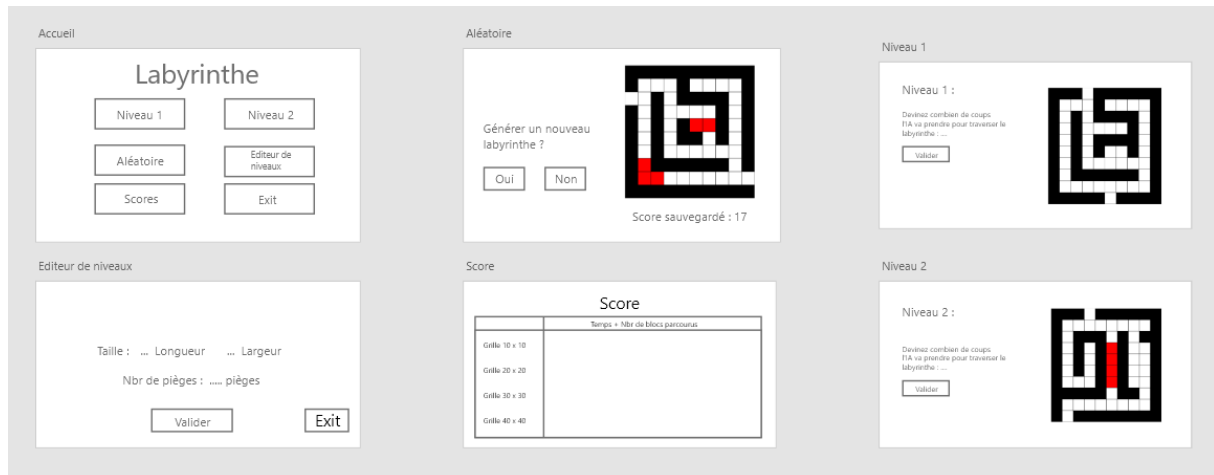


Figure 3: Maquette du projet avant l'oral intermédiaire

Pour la première version de ce projet j'avais réalisé des maquettes afin de m'aider à développer le labyrinthe, cependant j'ai préféré modifier ces maquettes après le rendu intermédiaire afin de rendre mon projet plus simple d'utilisation en affichant tous les éléments sur une même page à l'exception de la page des highscores.

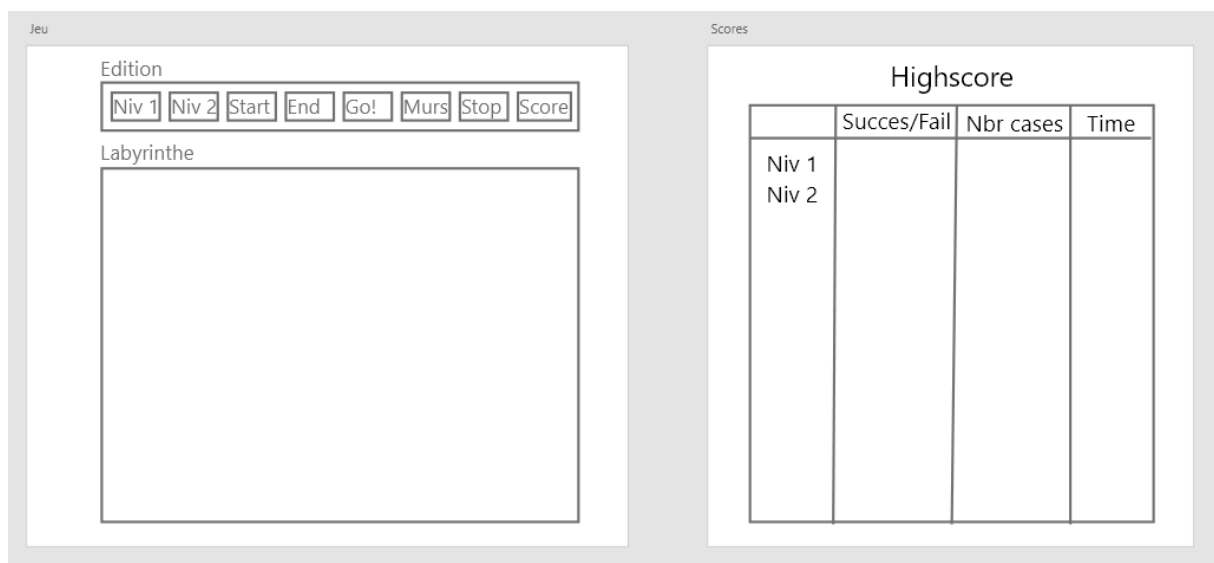


Figure 4: Maquette du projet après l'oral intermédiaire

Avec ce nouveau design nous pouvons voir que l'éditeur du labyrinthe et les deux niveaux sont regroupés sur la même page rendant ce projet plus ergonomique, seul les scores sont visibles sur une autre page afin de ne pas trop distraire l'utilisateur.

Besoins fonctionnels

Ce paragraphe va décrire les besoins fonctionnels du mini-projet

Utilisateurs du projet

Le mini-projet sera utilisé par un seul type d'acteur et qui sera nommé joueur.

Le joueur devra pouvoir :

- Choisir un des deux niveaux prédéterminés

- Créer son propre niveau
- Afficher les highscores
- Décider du point d'arrivée et de départ
- Lancer l'IA pour résoudre le labyrinthe

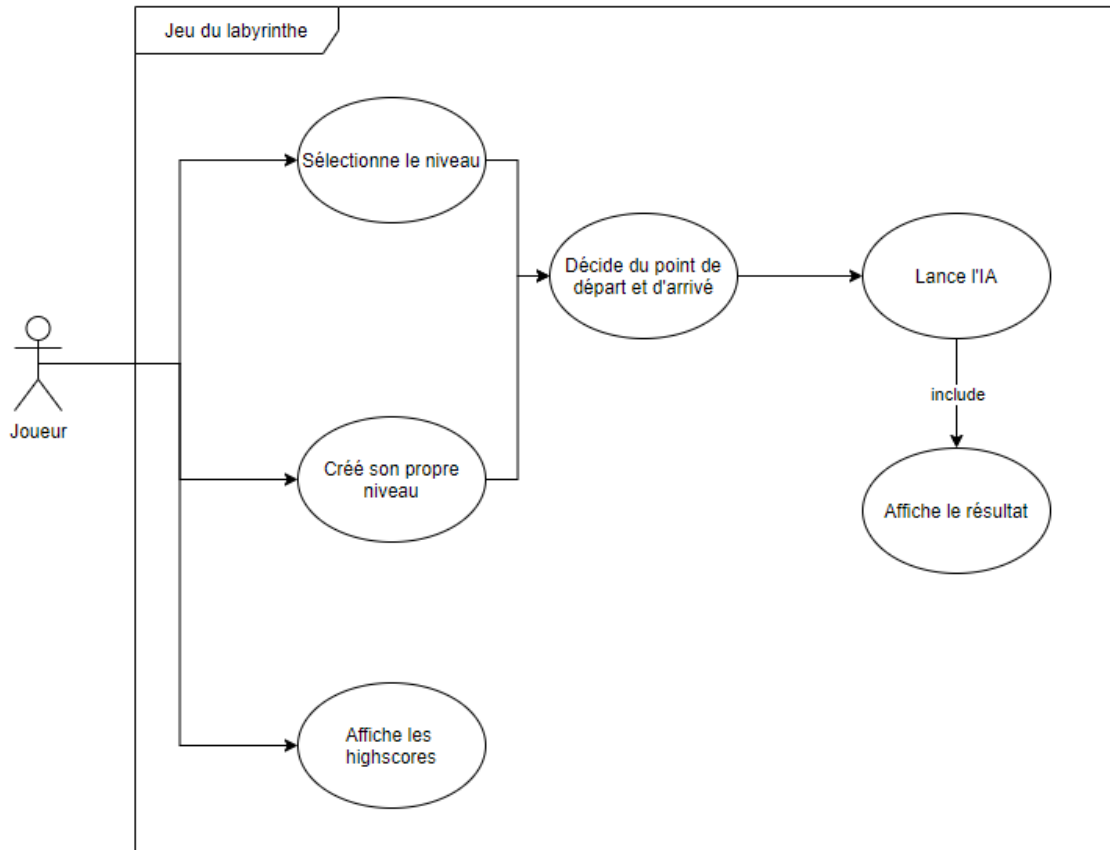


Figure 5: Use case, la vue du joueur

Diagramme de classe

Le logiciel d'IA de jeu sera constitué d'un package IA_DE_JEU. Ce package contient 3 classes

- Interface labyrinthe : classe qui permet de :
 - Choisir le niveau
 - Ajouter des obstacles
 - Recommencer le jeu
 - Lancer IA
- Labyrinthe : classe qui contient l'emplacement des obstacles, départ et fin.
- Solution labyrinthe : classe qui résout le labyrinthe

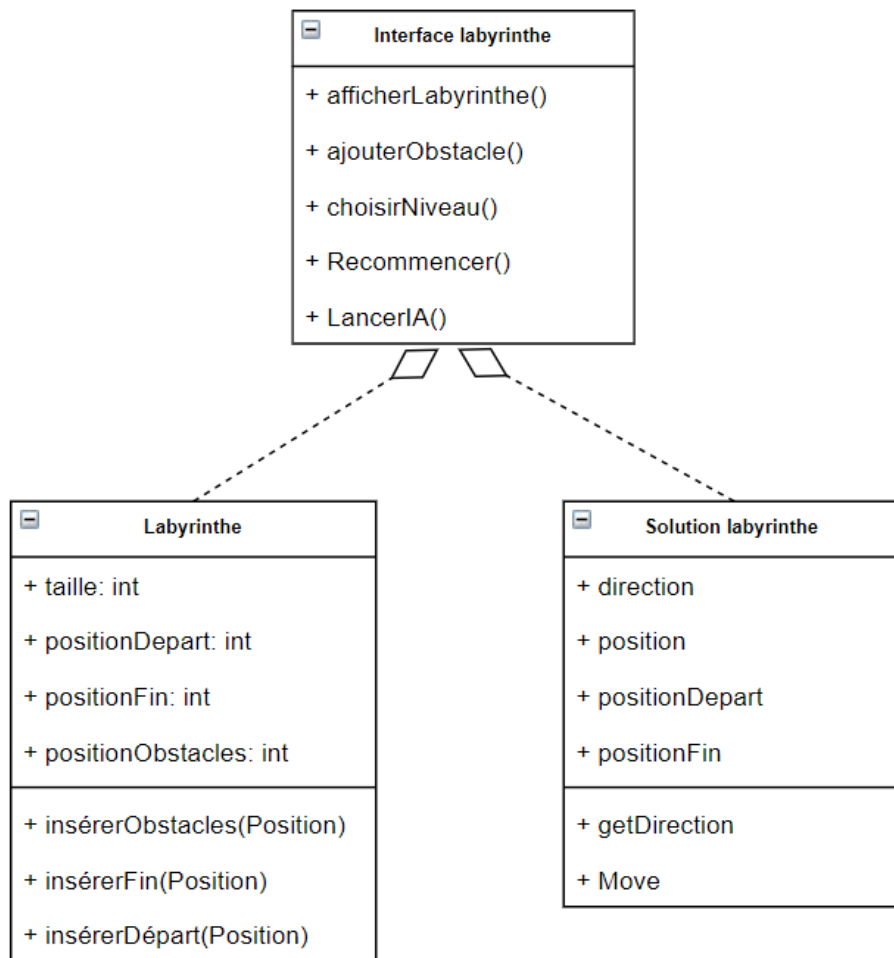


Figure 6: Diagramme de classe, package IA_DE_JEU

Diagramme de séquence

Le logiciel qui va permettre de générer le labyrinthe fait de 2 enchaînement principaux :

- La génération du labyrinthe
- L'IA qui résout ce labyrinthe

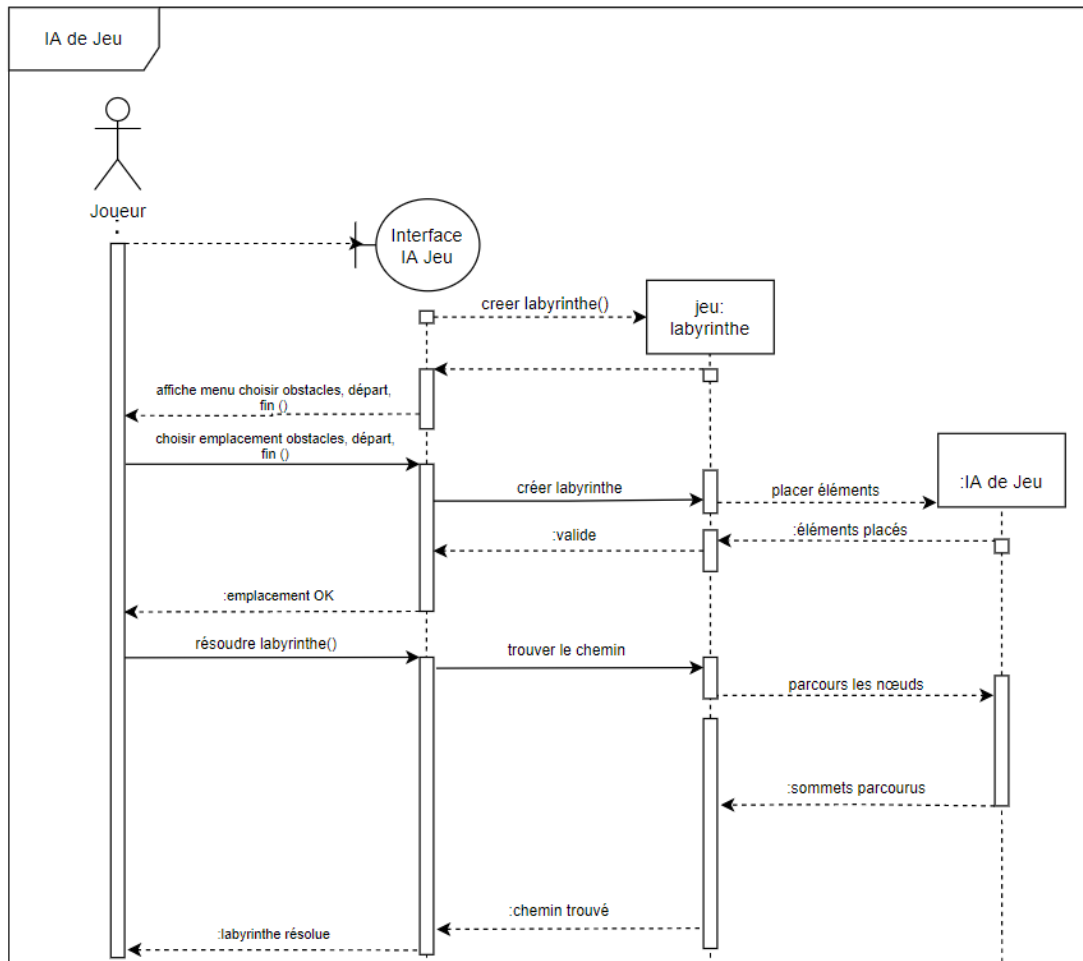


Figure 7: Génère un labyrinthe

Captures écran :

Dans cette section vous pouvez voir des captures d'écran des différents éléments disponible dans ce projet.

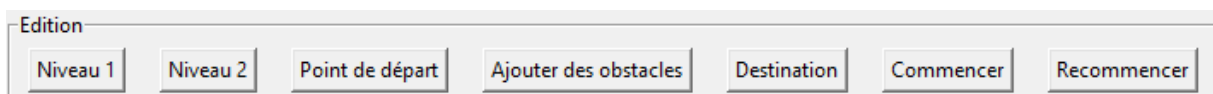


Figure 8: Menu

Labyrinthe																			
0	1	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	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379
380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399

Figure 9: Premier niveau du jeu

Labyrinthe																			
0	1	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	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379
380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399

Figure 10: Deuxième niveau du jeu

Labyrinthe																			
0	1	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	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379
380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399

Figure 11: Obstacles placés par le joueur

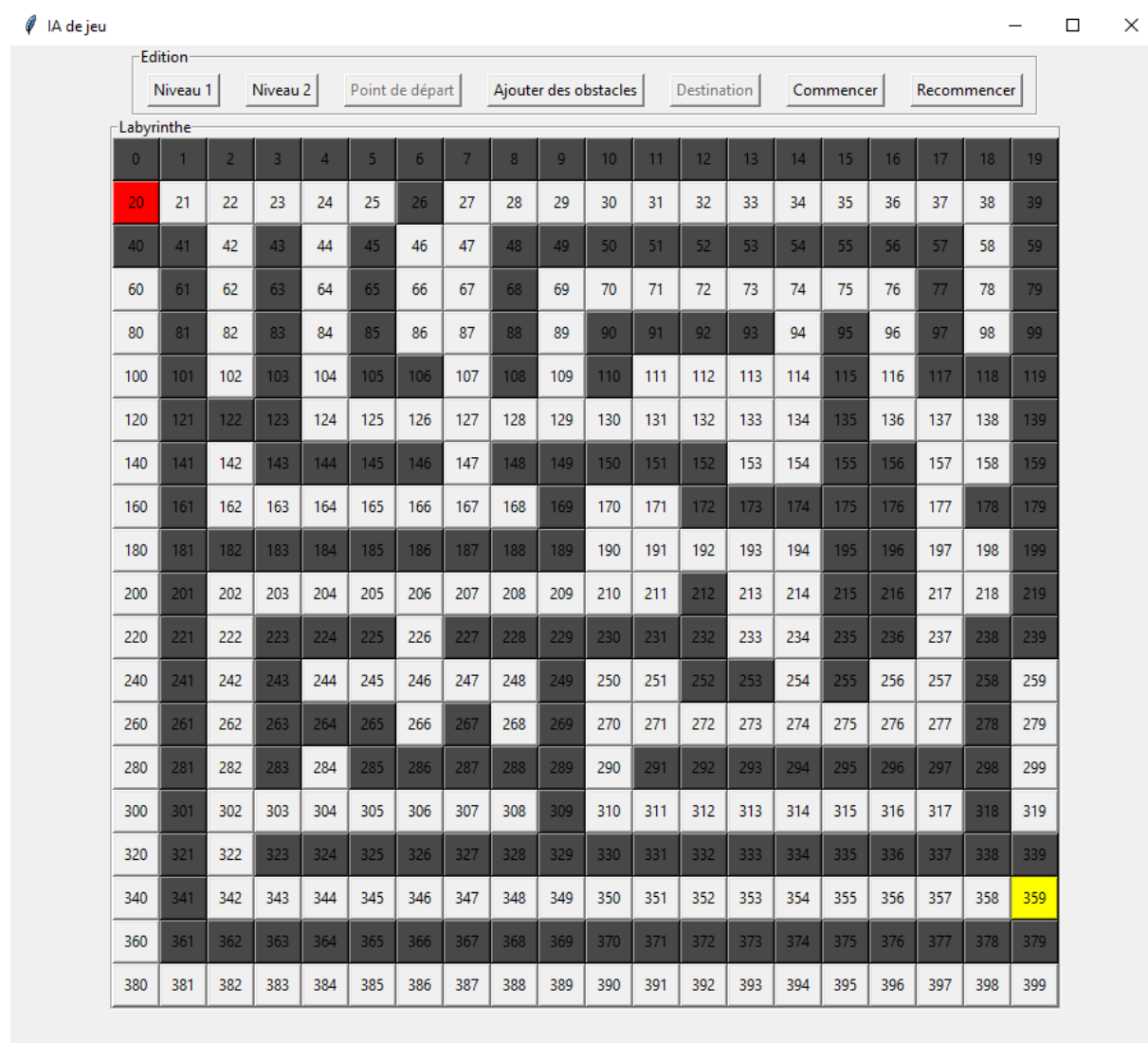


Figure 12: Niveau 1 du labyrinthe, point de départ et d'arrivée sélectionné par le joueur

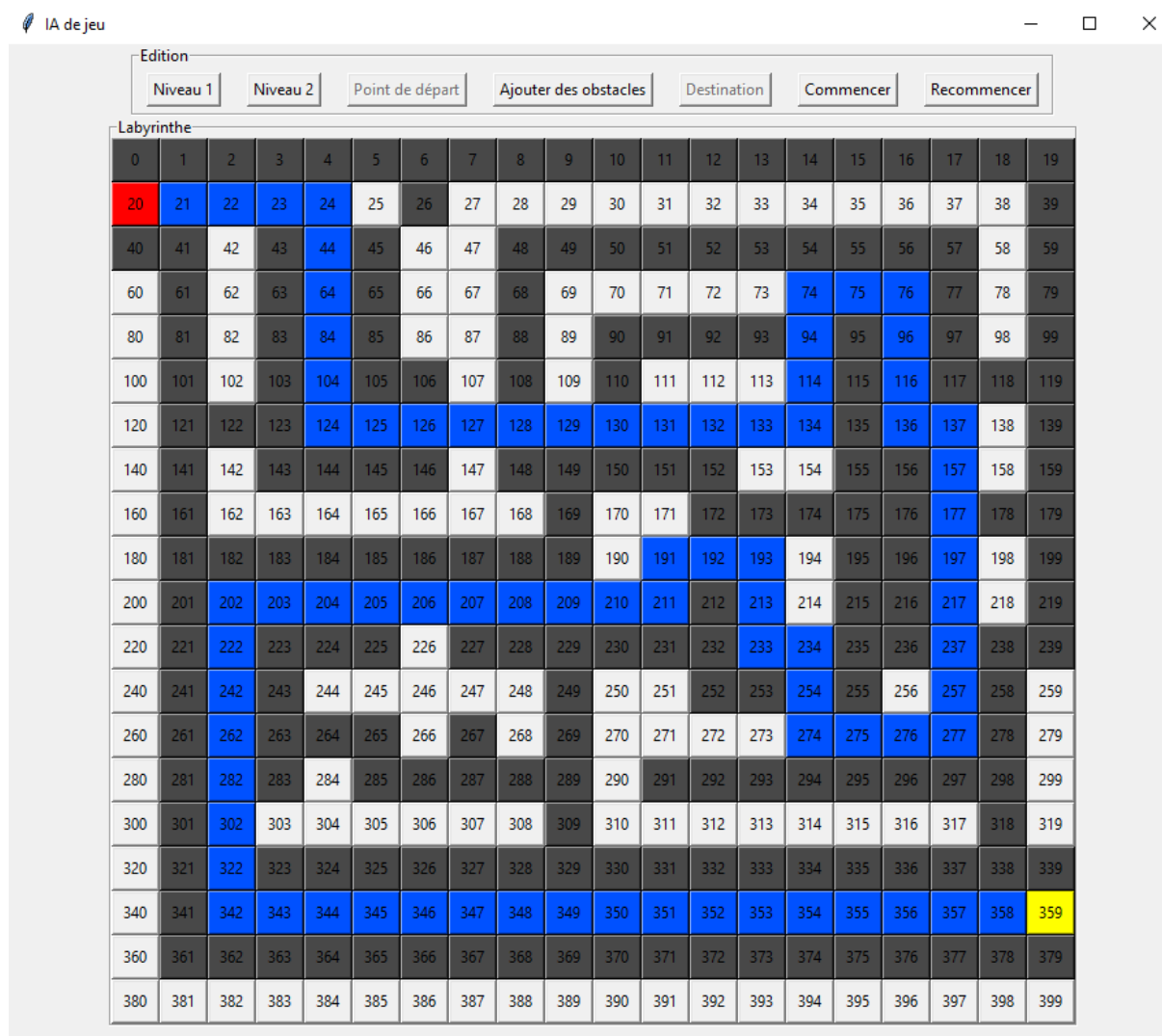


Figure 13: Chemin trouvé par l'IA

Annexe :

Table des tableaux :

Tableau 1: Planning prévisionnel	3
Tableau 2: Planning modifié après l'oral intermédiaire	3
Tableau 3: Matrice du graphe à quatre nœuds.....	5

Table des illustrations :

Figure 1: Trello du projet.....	4
Figure 2: Graphe à quatre nœuds	5
Figure 3: Maquette du projet avant l'oral intermédiaire	6
Figure 4: Maquette du projet après l'oral intermédiaire	6
Figure 5: Use case, la vue du joueur.....	7
Figure 6: Diagramme de classe, package IA_DE_JEU	8
Figure 7: Génère un labyrinthe	9
Figure 8: Menu	9

Figure 9: Premier niveau du jeu	10
Figure 10: Deuxième niveau du jeu	11
Figure 11: Obstacles placés par le joueur.....	12
Figure 12: Niveau 1 du labyrinthe, point de départ et d'arrivée sélectionné par le joueur	13
Figure 13: Chemin trouvé par l'IA.....	14