

Télécom INT
Année scolaire 2024/2025

Projet Informatique 1re Année PRO 3600

Ducournau Ethan
Prugnat Aurélien
Deleporte Hugo
Benois Loup
Jean-Marie Matisse

Enseignant responsable : Lallet Éric

10/02/2025

I. Introduction.....	2
II. Cahier des charges.....	3
III. Gestion de projet.....	4
A. Diagramme de Gantt.....	4
B. Plan de charge.....	5
C. Suivi d'activité.....	6
IV. Développement.....	7
A. Analyse du problème et spécification fonctionnelle.....	7
B. Conception préliminaire.....	7
1. Diagramme de classe.....	7
2. Fonctionnement.....	7
C. Conception détaillée.....	7
D. Codage.....	8
E. Tests.....	8
V. Manuel utilisateur.....	9
VI. Conclusion.....	10
VII. Bibliographie.....	11
VIII. Annexes.....	12
A. Annexe 1 : Méthode de “...”.....	12

I. Introduction

Le but est de réaliser un jeu sur ordinateur en Java. Pour le jeu, Il s'agit d'un Tetris où les blocs se transforment en sable, soumis à une physique réaliste lorsqu'ils touchent le sol ou d'autres blocs. Pour faire disparaître du sable, du sable d'une même couleur doit toucher simultanément les côtés droit et gauche du niveau.

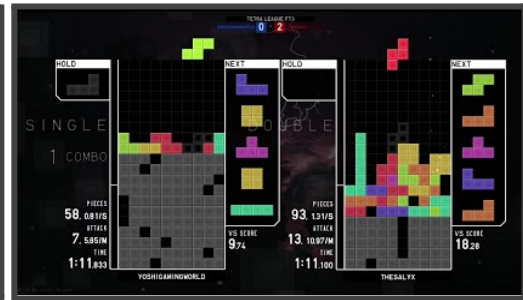
Dans le cas de la présence d'un mode multijoueur, nous nous inspirerons du jeu web Tetr.io qui implémente des mécaniques de jeu adaptées à un affrontement sous forme de Battle royal (Il s'agit d'un jeu où le but est de rester en vie le plus longtemps).



Sandtris



Tetr.io



II. Cahier des charges

Fonctionnalité	Tests
Implémentation de la physique du sable. Remarque : Chaque grain de sable doit tomber à nouveau lorsque du sable disparaît en dessous.	<ul style="list-style-type: none">- Le sable ne doit pas sortir de la fenêtre de jeu- Il doit y avoir des collisions entre les grains de sable- Le bloc doit se transformer en sable
Détection des différentes composantes connexes formées par les différentes couleurs de sable.	<ul style="list-style-type: none">- La composante doit savoir si elle touche les bords du terrain- Les diagonales ne doivent pas être considérées
Gestion du score qui prend en compte la taille de la composante qui vient d'être supprimée.	<ul style="list-style-type: none">- On marque autant de points que le nombre de grains de sable dans une composante connexe
Implémentation d'une interface de jeu avec : un tableau des "high scores", un menu, un choix de difficulté et le niveau qui est en train d'être joué.	<ul style="list-style-type: none">- Le tableau doit être trié par score décroissant- On doit pouvoir changer de menu en cliquant sur les différents boutons.
Un mode multijoueur au format 1vs1 (ou plus) inspiré de "Tetr.io".	<ul style="list-style-type: none">- Il faut que les points que l'on obtient viennent provoquer des actions qui dérangent l'adversaire.

III. Gestion de projet

A. Diagramme de Gantt

catégorie	Tache	Semai ne 1	Semai ne 2	Semai ne 3	Semai ne 4	Semai ne 5	Semai ne 6	Semai ne 7	Semai ne 8	Semai ne 9	Semai ne 10	Semai ne 11	Semai ne 12	Semai ne 13	Semai ne 14	Semai ne 15	Semai ne 16	Souten ance		
Générale	Squelette du code																			
	Tombé et réparation des pièces																			
Menu	Menu de démarrage																			
	Menu de défaite																			
	Menu de score, Base de donné																			
	Menu de paramètre																			
Partie principale à redécoupé	physique du sable																		Respon sable	
	cassage de la pièce																			
	détection des composants connexes																			
	Disparition des composants connexes																			
	Condition de défaite																			
Gestion des pièce	Génération de la forme																			
	vitesse des pièces																			
	Prévisualisation forme + couleur																			
Features	Choix difficulté																			
	Multijoueur locale																			

B. Plan de charge

	PLAN DE CHARGES PRÉVISIONNEL							SUIVI D'ACTIVITÉS (Charge Consommée)						
	Charge en H / Participant							Charge en H / Participant						
Description de l'activité	Charge en %	Charge en H	Deleport Hugo	Prugnat Aurelien	Jean-Marie Matisse	Ducourneau Ethan	Benois Loup	Charge en %	Charge en H	Deleport Hugo	Prugnat Aurelien	Jean-Marie Matisse	Ducourneau Ethan	Benois Loup
Total	100%	250	50	50	50	50	50	100%	5	1	1	1	1	1
Gestion de projets	11%	28	7	8	4	5	4	100%	5	1	1	1	1	1
Réunion de lancement	2%	5	1	1	1	1	1	100%	5	1	1	1	1	1
Planning prévisionnel et Suivi d'activités	3%	8	0	6	1	1	0	0%	0	0	0	0	0	0
Réunions de suivi	2%	5	1	1	1	1	1	0%	0	0	0	0	0	0
Rédaction	2%	4	0	0	1	1	2	0%	0	0	0	0	0	0
Outils collaboratifs (svn, etc.)	2%	6	5	0	0	1	0	0%	0	0	0	0	0	0
Spécification	4%	10	3	2	0	5	0	0%	0	0	0	0	0	0
Définition des fonctionnalités	4%	10	3	2	0	5	0	0%	0	0	0	0	0	0
Conception préliminaire	24%	59	10	10	14	11	14	0%	0	0	0	0	0	0
Définition d'un modèle de données	10%	25	5	5	5	5	5	0%	0	0	0	0	0	0
Définition d'un format de fichiers associé au modèle de données	4%	11	0	2	4	1	4	0%	0	0	0	0	0	0
Définition des fonctionnalités	4%	11	3	3	0	5	0	0%	0	0	0	0	0	0
Définition des modules	5%	12	2	0	5	0	5	0%	0	0	0	0	0	0
Conception détaillée	27%	67	11	15	14	13	14	0%	0	0	0	0	0	0
Définition des classes	2%	4	2	0	0	2	0	0%	0	0	0	0	0	0
Définition des méthodes	5%	12	3	2	2	3	2	0%	0	0	0	0	0	0
Définition des tests unitaires	4%	10	4	2	1	2	1	0%	0	0	0	0	0	0
Auto-formation	14%	36	1	10	10	5	10	0%	0	0	0	0	0	0
Maquettage des interfaces	2%	5	1	1	1	1	1	0%	0	0	0	0	0	0
Codage	22%	56	13	9	12	10	12	0%	0	0	0	0	0	0
Codage des classes	3%	7	3	2	0	2	0	0%	0	0	0	0	0	0
Codage des méthodes	10%	25	10	5	2	6	2	0%	0	0	0	0	0	0
Codage des tests unitaires	10%	24	0	2	10	2	10	0%	0	0	0	0	0	0
Intégration	0%	0	0	0	0	0	0	0%	0	0	0	0	0	0
Intégration des modules	0%	0	0	0	0	0	0	0%	0	0	0	0	0	0
Tests d'intégration	0%	0	0	0	0	0	0	0%	0	0	0	0	0	0
Soutenance	12%	30	6	6	6	6	6	0%	0	0	0	0	0	0
Préparation de la soutenance	10%	25	5	5	5	5	5	0%	0	0	0	0	0	0
Soutenance	2%	5	1	1	1	1	1	0%	0	0	0	0	0	0

C. Suivi d'activité

Le suivi des activités se fait via les "issues" Github et pourra être reporté ici ultérieurement.

IV. Développement

A. Analyse du problème et spécification fonctionnelle

Pour la physique du sable : Pour l'effet tas de sable on a décidé que la meilleure façon de le faire est la suivante. Si un grain de sable n'est pas au sol, n'a personne au-dessus de lui et qu'il n'y a personne en bas à gauche de lui alors il tombe sur la gauche (de même, dans les mêmes conditions mais où il n'y a personne en bas à droite alors il tombe à droite). Il faut donc savoir comment les différents grains de sable sont situés les uns par rapport aux autres. De plus, lorsqu'un grain de sable disparaît il faut faire en sorte que celui au-dessus commence sa chute libre.

B. Conception préliminaire

1. Diagramme de classe

Le projet repart de zéro et n'a donc pas de diagramme des classes intéressantes pour le moment.

2. Fonctionnement

Pour la physique du sable : Deux options s'offrent à nous : une matrice géante où chaque emplacement correspond à un grain de sable, ou une classe grain de sable qui aurait comme attribut des références vers les grains de sable autour de lui.

Après un brainstorming entre Hugo et Ethan la seconde s'est révélée plus intéressante. En effet, cette seconde option permet d'utiliser moins d'espace en mémoire qu'une matrice gigantesque.

C. Conception détaillée

Pour l'instant, il n'y a qu'une fenêtre avec un carré que l'on peut déplacer de droite à gauche.

D. Codage

1. Choix des outils

On utilise JavaFX à la place de swing, puisque swing n'est plus mis à jour depuis 15 ans. De plus, on veut se contenter d'un outil simpliste afin de réaliser le maximum de chose nous-mêmes. On utilise également Maven pour faciliter la gestion des packages. Pour la gestion des scores, nous allons utiliser une base de données gérée via MySQL puisqu'il s'agit d'un outil que tout le monde a déjà utilisé dans le groupe.

E. Tests

Pour l'instant, aucun test n'a été réalisé. C'est pourquoi cette partie est vide pour le moment.

V. Manuel utilisateur

Ce manuel sera rempli dès lors que le projet aura un avancement plus important.

VI. Conclusion

C'est ici que nous noterons si le projet a été un échec ou une réussite que ce soit du point de vue de la gestion de projet ou du développement ainsi que les raisons justifiant le résultat final.

VII. Bibliographie

[1] : API java (<https://docs.oracle.com/javase/8/docs/api/>)

VIII. Annexes

A. Annexe 1 : Méthode de “...”