Hello everyone!



#### Université Sidi Mohamed Ben Abdellah

#### Faculté des Sciences DHAR EL-MEHRAZ













### Projet de Fin d'Études

Conception et réalisation du Jeu Tetris



Réalisé par : Mohammed EL KHOU & Mohammed KASMI

Encadré par : M. Mohammed EL MOUHAJIR

# PLAN

- 1 Introduction
- 2 Présentation Général
- 3 Concéption & Modélisation du Système
- 4 Réalisation de l'aplication
- 5 Conclusion & Perspectives



2018

- 53 % Evaluation de nombre des joueurs 20 %
- 34 ans Evaluation de l'âge moyen des joueurs 21 ans
- 4,3 milliards Evalua<mark>tion de chiffre d'a</mark>ffaires 862 millions

## L'évolution des jeux videos

Depuis: fr.statista.com & www.huffingtonpost.fr



#### 2. Présentation Général:

**Objectifs** 



9

#### Présentation Général

les objectifs



# Une version graphiques du jeu

c'est la réalisation de L'interface graphique, qui permettra à l'utilisateur d'interagir avec notre programme de manière plus conviviale.

# Contrôler le déplacement des pieces

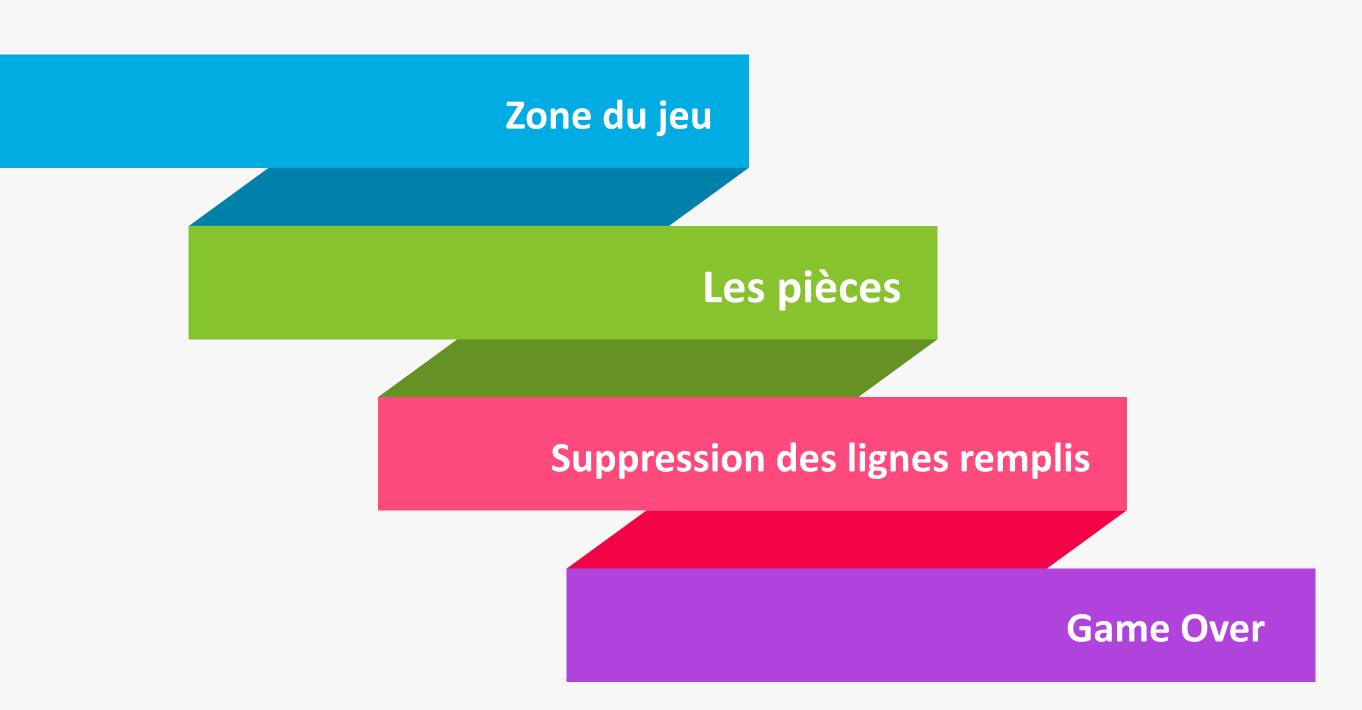
déplacer les pièces

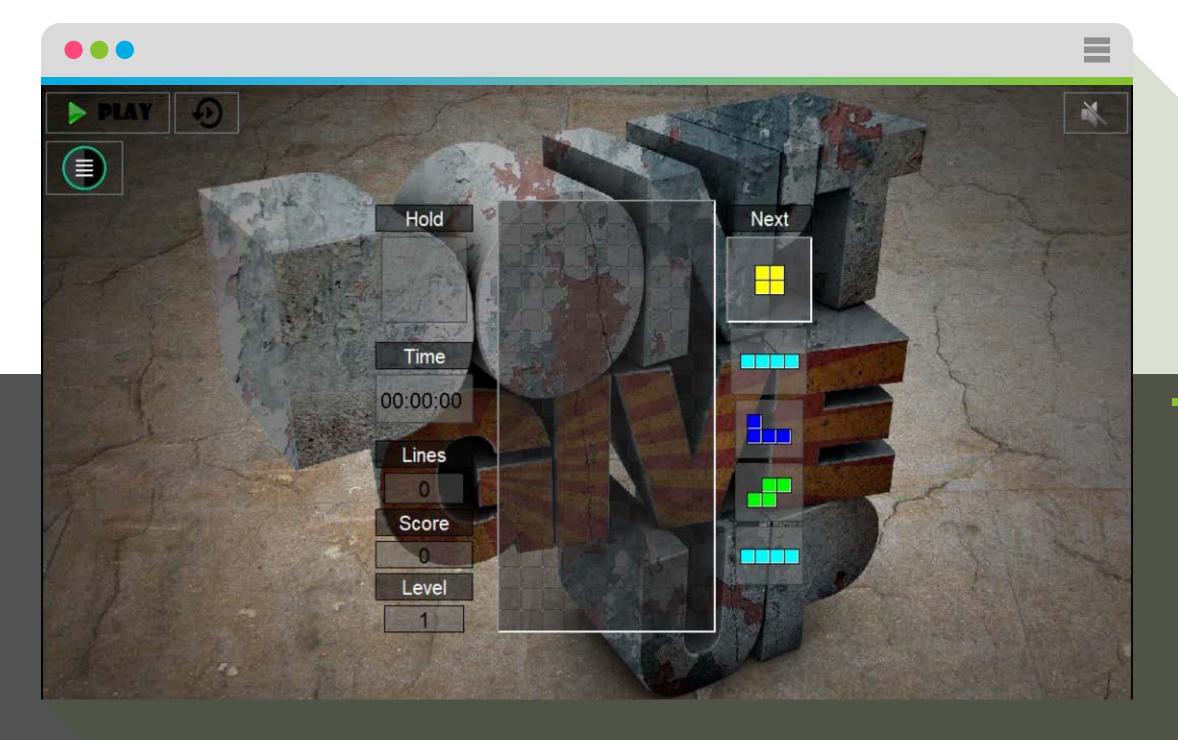
Rotation les pièces



### Présentation Général

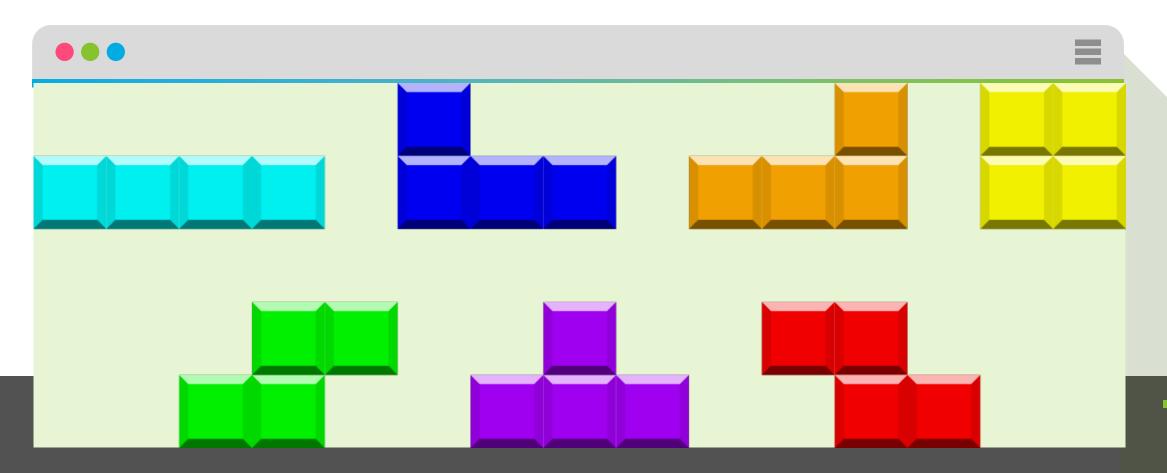
**Déscription** 





### Zone du jeu

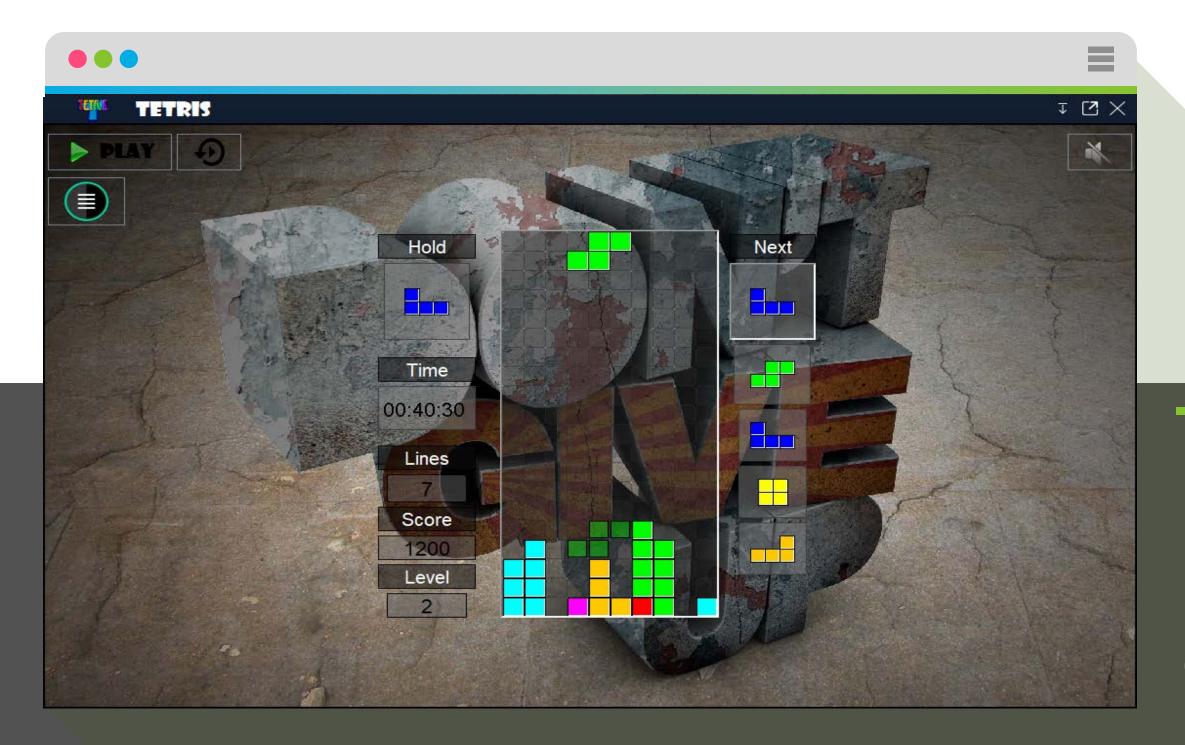
Le jeu est principalement composé d'un champ de jeu à largeur fixe(20 lignes\*10 colonnes), où des pièces descendent du haut de l'écran vers le bas



# Les pièces:

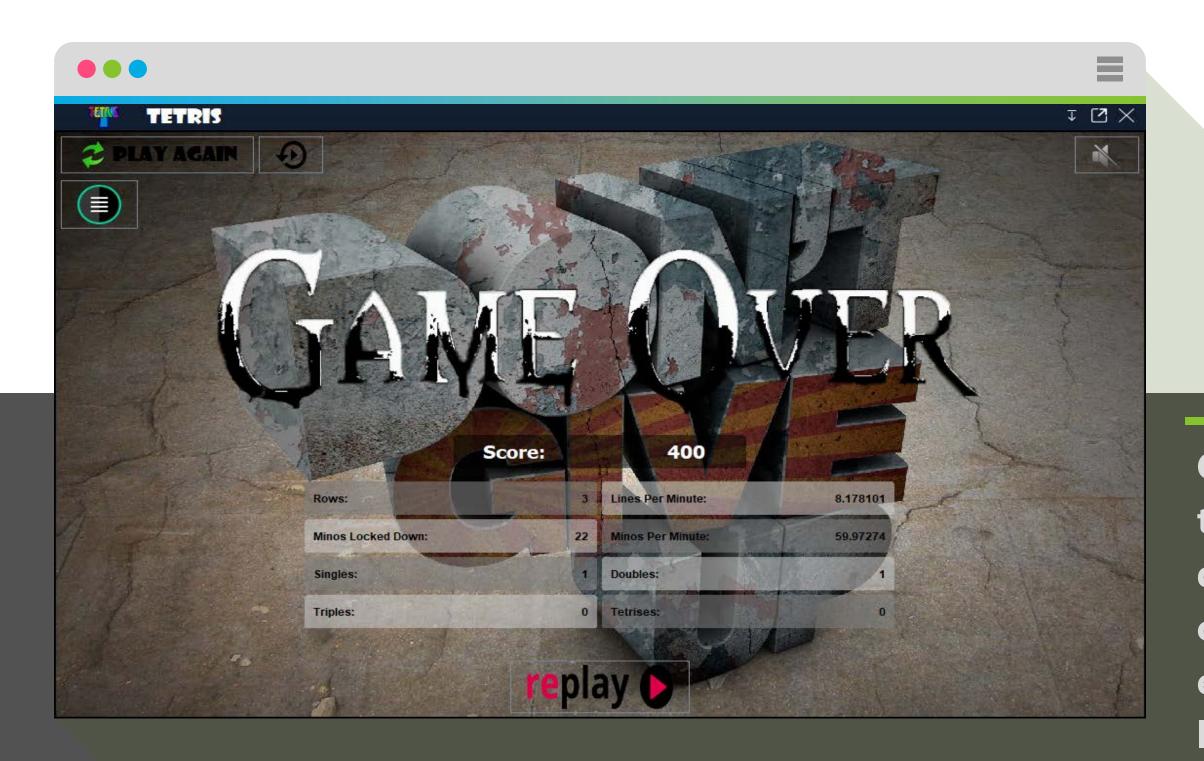
Les pièces de Tetris, sur lesquelles repose entièrement le jeu sont :

- →de 7 formes différentes.
- → chaqune d'elles est formée de 4 carreaux.



# Suppression des lignes remplis

Les lignes complétées disparaissent tout en rapportant des points et le joueur peut de nouveau remplir les cases libérées.



#### Game Over

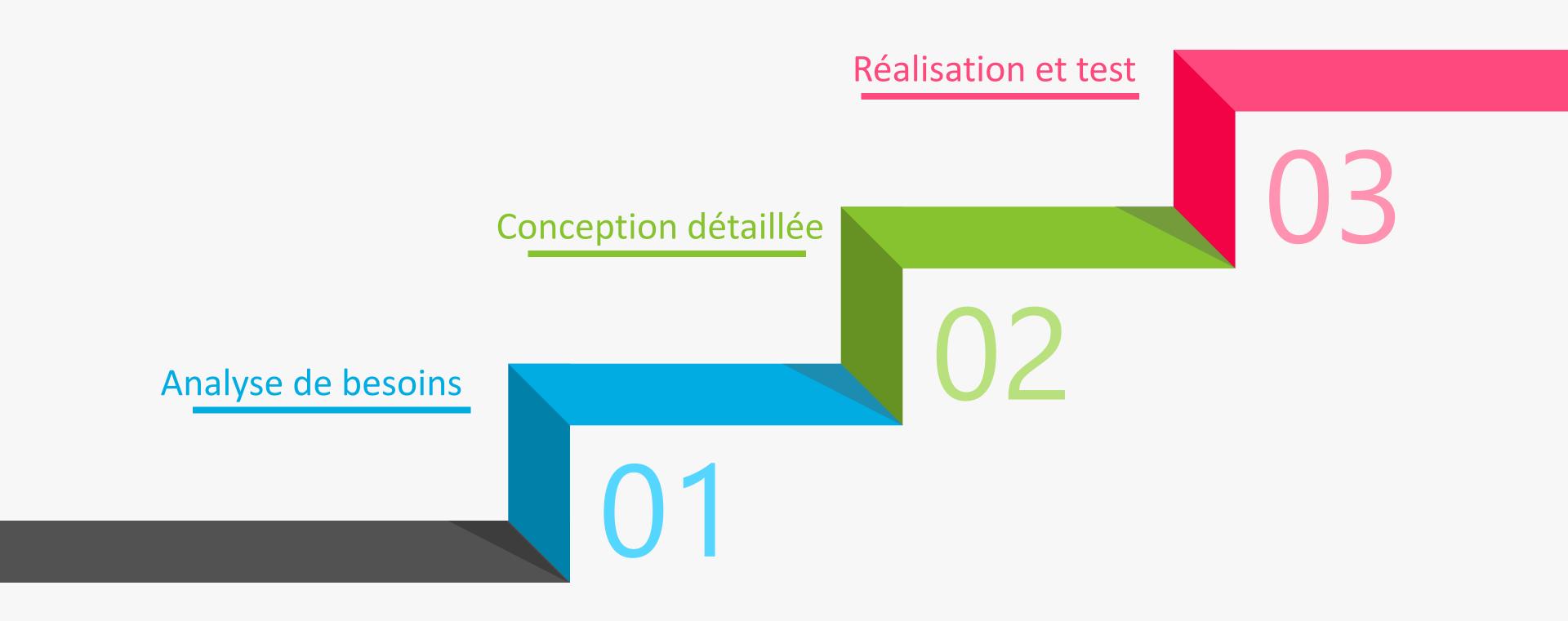
Chaque fois qu'une partie se termine, notre interface graphique devra imprimer dans le champ de jeu combien de lignes l'utilisateur a effacées au total pendant le match et lui donner aussi la possibilité de démarrer une nouvelle partie ou de quitter la plateforme de jeu.

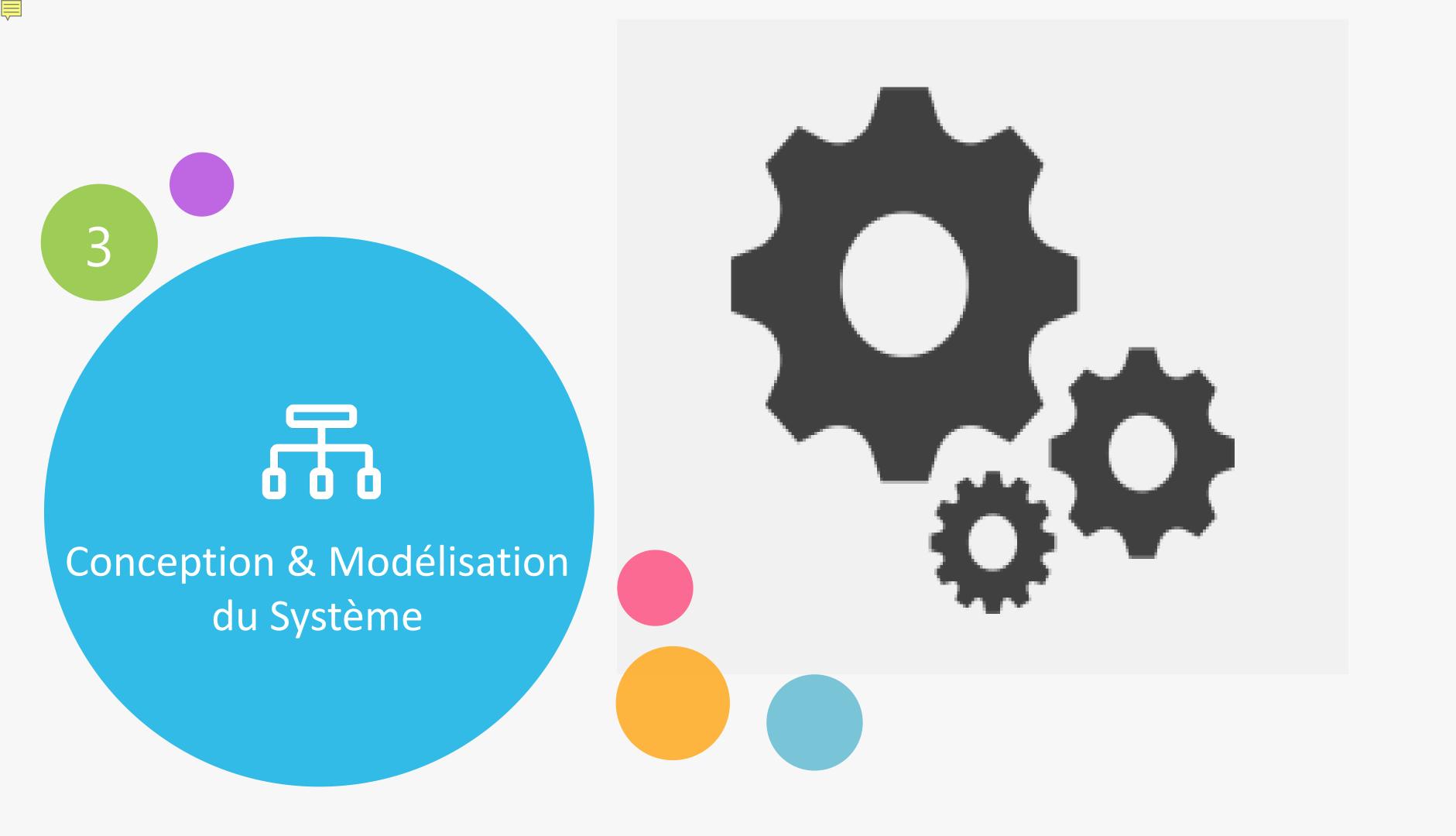
14



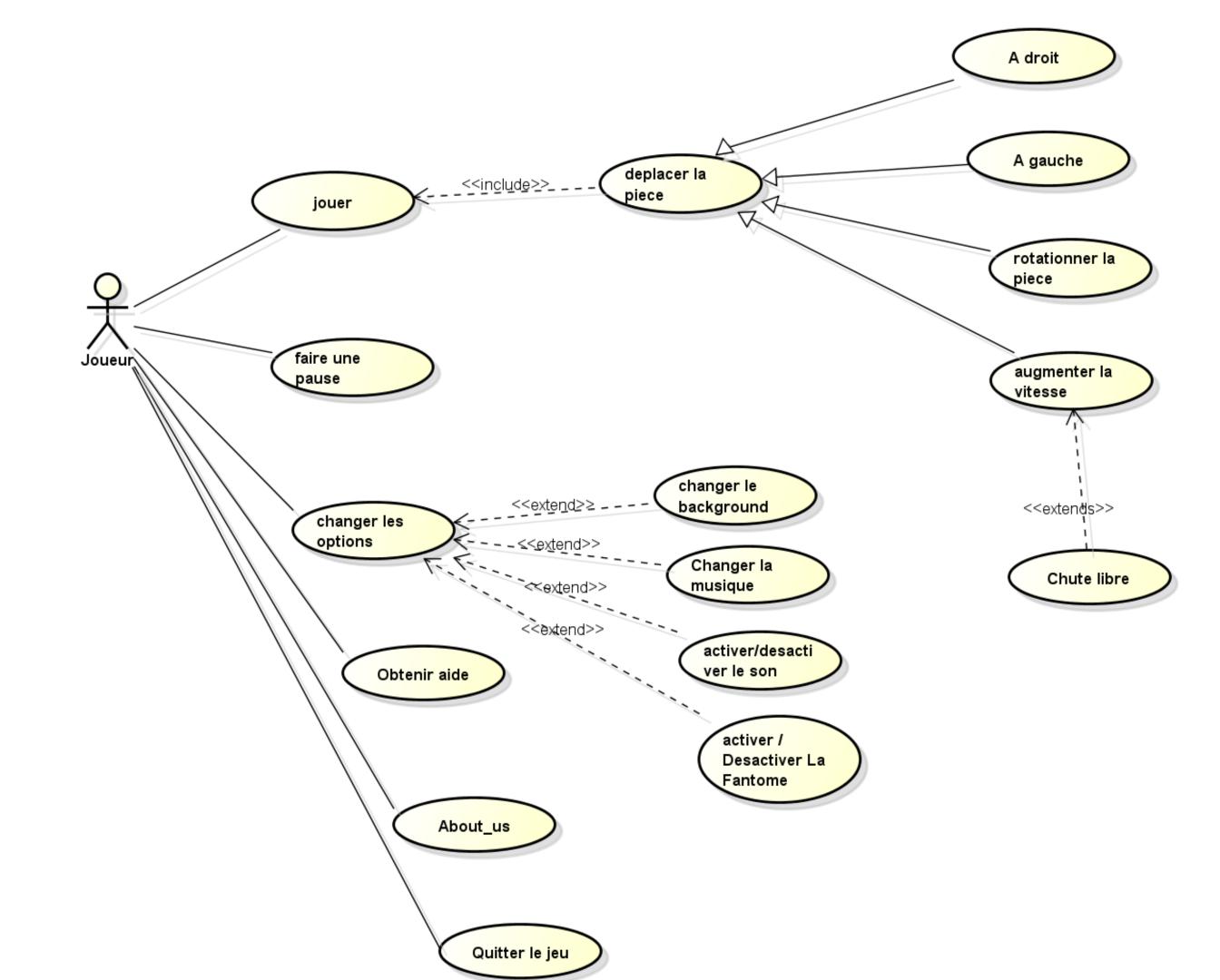
### 2. Présentation Général

Les phases de notre project :

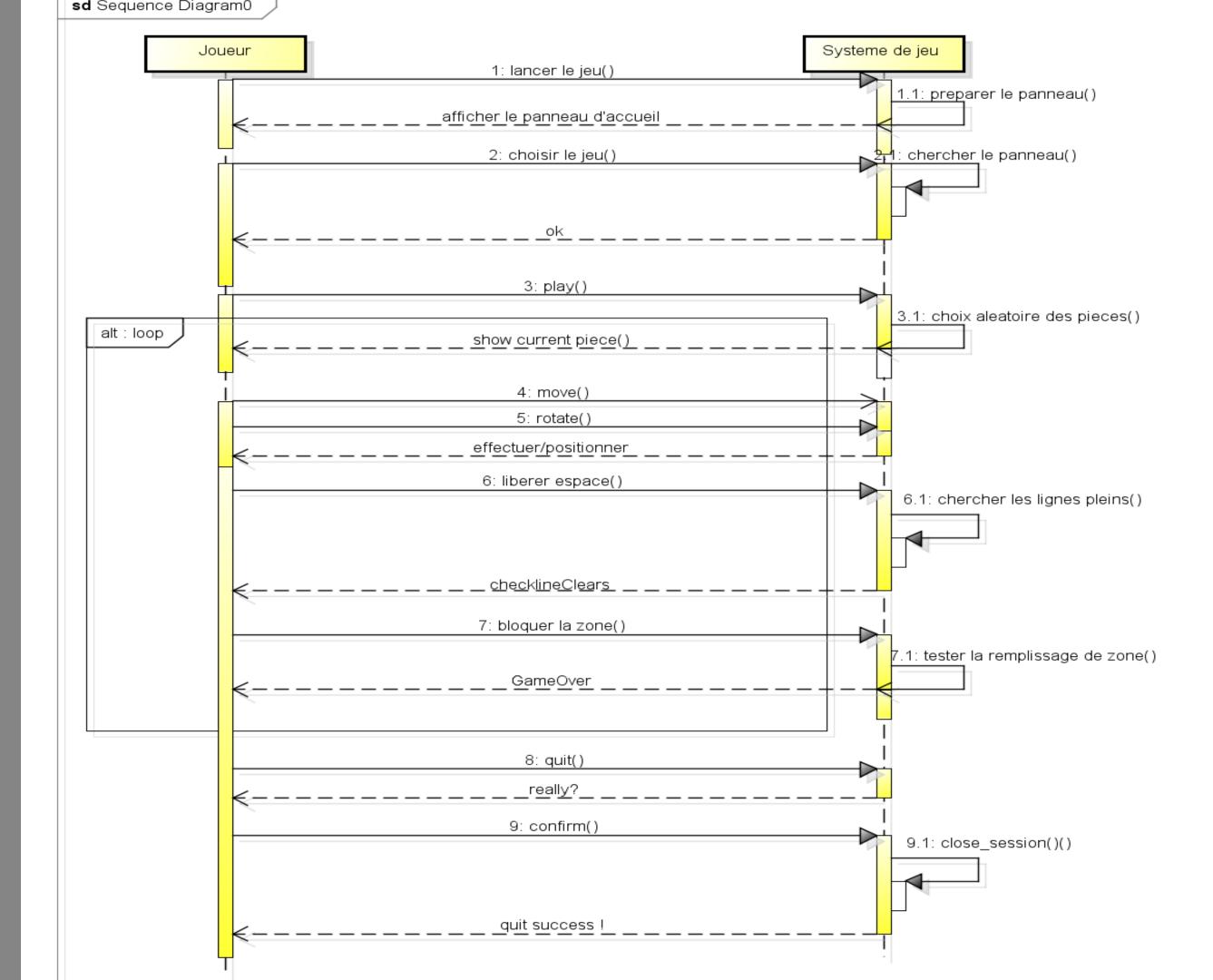




3.1.
Diagramme
de cas
d'utilisation

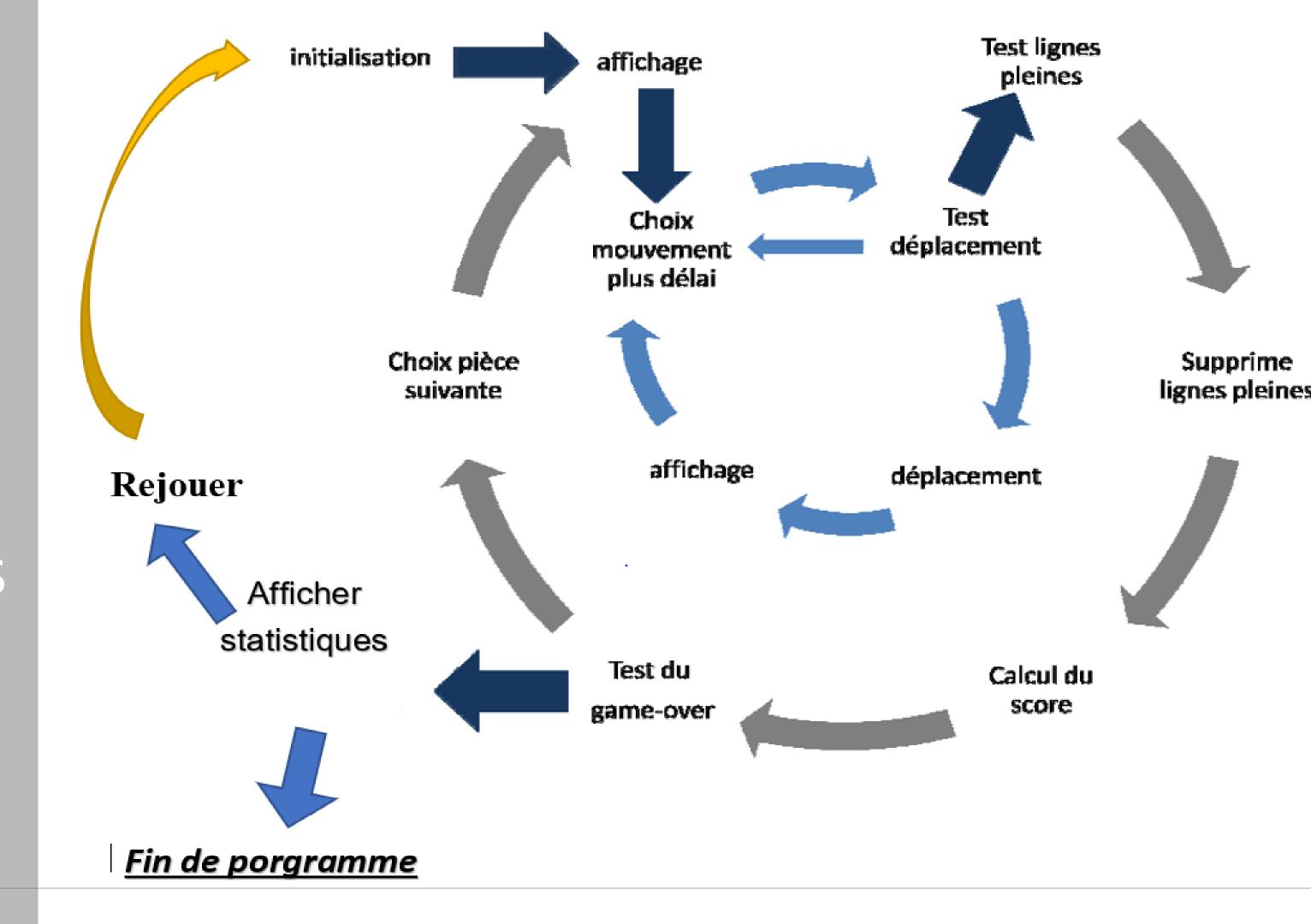


3.2.
Diagramme
de Séquence



#### **HighScores** Keyhandler Ghost ghost > - max;int : int + KeyHandler() - id : int + Ghost(Mino : int) : void + keyPressed(keyEvent : int) : void score : int + draw(Graphics : int) : void highscore + keyReleased(keyEvent : int) : void - list(): Vector<Score> + HighScore() + showTop10(): void + testHighScore(id : int) : Boolean Gamepanel + remplir(): void - movesound : AudioClip + signup(): void Mino tetrisTheme : AudioClip - level : int color : Color - score : int + pCoors(): points[] GameOver - lins : int + id(): int minocount : int - time : String 0..1 + Mino(Mino: int, boolean: int) combocount : int - score : int \_currentMino ▶ + Mino(id: int) - FPS : int ■ gamestats - lines : String + getStartCoors(): Points[] compteur : int + GameOver(id : int) + draw(Graphics : int) : void - timer : Timer + getImage(): Image + move(Point : int) : void - ghosted : boolean + getImage(String : int, Image : int) : ImageIcoon + rotate(id : int) : void musicOn : boolean + reverserotate(): void + GamePanel(JFrame : int) : void + resetGame(): void Stats Panel **⋖**Stats 0..1 0..1 + action(): void + stop(): void + StatsPanel(): void + start(): void + paintCompenent(g : Graphics) : void + restart(): void + getPerferredZize(): Dimension **⋖** matrix MinoQueue + getGhostCoors(): void + minoOutOfbounds(Mino : int) : boolean 0..1 - heldMino : Mino Minoqueue + bottonEmpty(Mino : int) : boolean - rand : int MatrixPanel + checkMoveAgainstmatrix(): void minos : ArrayList<Mino> + KeyUpdate(): void elemenets : MatrixElement[][] <sup>1</sup> MinoQueue() + paintComponent(Graphics: int): void 0..1 + shiftUp(): void + setBackImage(img : Image) : void + MatrixPanel() Timer + newMino(): Mino + checklineClears(): void + updateLevel(): void + setHeld(Mino : int) : void - min : String Matrix Element + getHeld(): Mino + removeRow(): void - sec : String - color : Color hund : String - coors : Points + MatrixElement(clr : Color, coors : Point) + MatrixElement(elem : boolean) ■ MatrixElement + draw(g : boolean, dim : int) : void

3.4.
Diagramme
d'activités

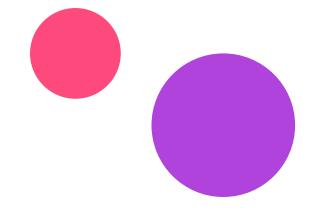














Environnement de travail

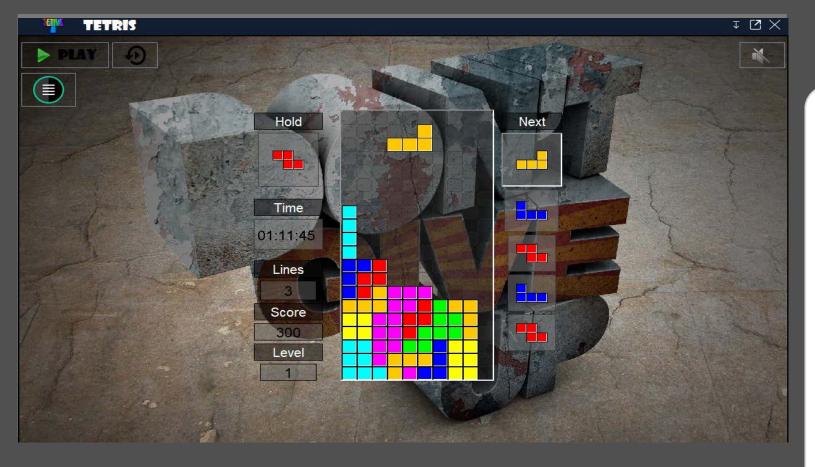


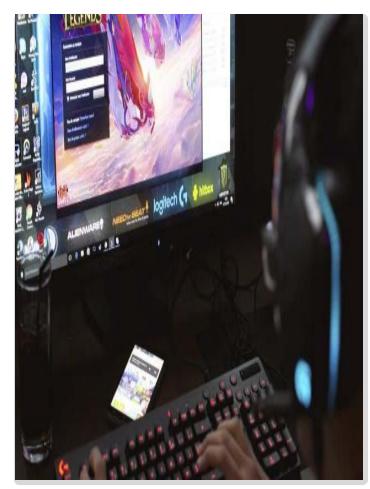




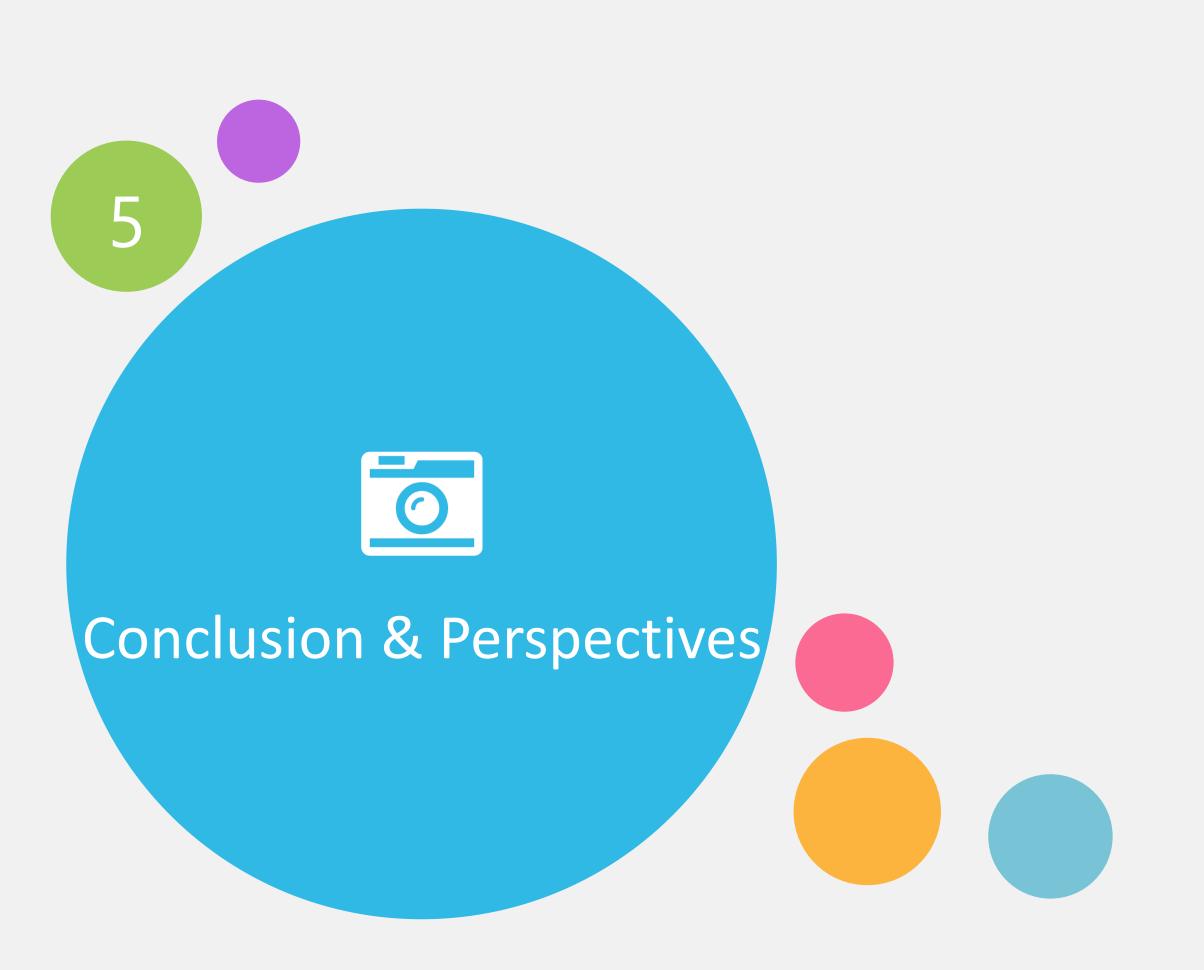








Test



### Conclusion & Perspectives

Ce projet nous à permis d'avoir une approche complète du développement de logiciel et une bonne initiation au cycle complet du développement.

# Merci pour votre Attention!



Any Questions?

# Merci pour votre Attention!

Any Questions?

Merci pour votre Attention!

