## Document de conception v4.1 ■ GameController (A) Entite getCommand() Elements (A) Case Modele C Hero **C** Monstre Modele C GameEngineGraph Modele C TextureFactory C Labyrinthe int x C LabyrinthController int y int x; int x; int xcamera Bufferedimage textures.. int y; int y; int pv; void main() int[n][n] labyrinthe (0 ok, 1 mur) boolean deplacementPossible(int x, int y) int ycamera Bufferedlmage image int pv; kevListener() actionListener() void startConsole() void genererCombinaison() void startGraphique() boolean isDead() void seRapprocher() void handleSpecialEffect(Hero h) Modele C Niveau 1..n Engine Modele (I) Game **C** LabyrintheGenerator int niveau C GraphInterface C Arrive **C** Magique C Teleporteur C Piege **C** Brique C Sol int getPlayerX() int getPlayerY() void chargerNiveau(File file) void deplacerHero(x,y) String toString() bool isFinished() int[][] genererLabyrinthe() void paint() void evolve(Event e) 1..n **C** LabyrinthGame C DrawingPanel void genLabyrinth() String toString() void paint() I GamePainter Engine int getWidth() int getHeight() void draw() C LabyrinthPainter static int width static int height

## Répartition du travail :

	Julien	Noah	Chouaid	Elliot
Cases spéciales (Pièges, magiques etc.)	Χ			X
Système d'attaque		X		
Barre de vie			Х	
Gestion d'aggro des monstres			X	
Animation du héro	X			
Gestion de la victoire/défaite		X	X	X
Création des différents fichiers de labyrinthe				X
Fond d'écran du jeu	Χ			
Menu de sélection de niveaux	Х			
Tests unitaires				X