

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Jeu de la vie

Cour de récréation

Référence Fournisseur : CPT-1

Date : 07 février 2019

Version/Édition : 0.7

État : Finalisation

Type de diffusion : Diffusion restreinte

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	--	---

Historique des révisions

Date	Description et justification de la modification	Auteur	Pages / Chapitre	Edition / Révision
12/01/19	Création	Poujol / Bellin	toutes	0.1
13/01/19	Ajout de la modélisation	Poujol / Bellin		0.2
15/01/19	Ajout diagrammes de classe	Poujol / Bellin		0.3
18/01/19	Édition de la version 1	Poujol / Bellin		0.4
31/01/19	Finalisation de la version 1	Poujol / Bellin		0.5
04/02/19	Corrections multiples	Poujol / Bellin		0.6
07/02/19	Prise en compte des remarques	Poujol / Bellin		0.7

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Table des matières

1 Terminologie.....	5
1.1 Abréviations.....	5
1.2 Définitions des termes employés.....	5
2 Descriptoin et analyse de l'environnement.....	7
2.1 Organisation de l'espace de travail.....	7
3 Structure statique.....	8
3.1 Décomposition générale.....	8
3.2 Analyse statique.....	9
Classe Eleve extends ElementdeJeu.....	11
Défnition.....	11
Attributs.....	11
Opérations.....	11
Classe Eleve Turbulents extends Eleve.....	12
Défnition.....	12
Attributs.....	12
Opérations.....	12
Classe Professeur extends ElementdeJeu.....	12
Défnition.....	12
Attributs.....	12
Opérations.....	13
Classe Terrain extends ElementdeJeu.....	13
Défnition.....	13
Classe CharsetCR implements Tileset.....	14
Défnition.....	14
Attributs.....	14
Opérations.....	14
Classe Fenetre extends JFrame.....	14
Défnition.....	14
Classe FenetreParametres extends JFrame.....	14
Défnition.....	14
private class Panel extends Jpanel implements ActionListener.....	15
Classe ConsolePanneau extends Jpanel implements ActionListener.....	16
Défnition.....	16
Attributs.....	16
Opérations.....	16
Classe FenetrePopup extends JFrame.....	16
Défnition.....	16
Classe interne de FenetrePopup.....	16
Défnition.....	16
Classe Map extends JPanel.....	17
Défnition.....	17

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	--	---

Attributs.....	17
Opérations.....	17
Classe Tile.....	17
Défnition.....	17
Attributs.....	17
Opérations.....	17
Interface TileSet.....	17
Défnition.....	18
Classe TilesetCR implements Tileset.....	18
Défnition.....	18
Attributs.....	18
Opérations.....	18
Classe Cell.....	19
Défnition.....	19
Attributs.....	19
Classe Constant.....	19
Défnition.....	19
Attributs.....	19
Interface Drawable.....	19
Défnition.....	19
Opérations.....	19
Classe ElementdeJeu implements Drawable.....	20
Défnition.....	20
Attributs.....	20
Classe Grille.....	20
Défnition.....	20
Attributs.....	20
Classe Controller.....	20
Défnition.....	20
Attributs.....	20
Opérations.....	21
Défnition.....	21
Opérations.....	22
4Aspects dynamiques.....	24
4.1Echanges entre éléments.....	24
4.2Cas d'erreurs.....	25
5Interface graphique.....	26
5.1La fenêtre de paramètres.....	26
5.2Les acteurs.....	32
5.3La fenêtre de jeu	34
5.4Fin du jeu.....	36

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	--	---

1 Terminologie

1.1 Abréviations

UML Unified Modeling Language

1.2 Définitions des termes employés

attribut	Un attribut est une information caractéristique mémorisée par un objet.
Diagramme de séquence	Représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML
Cas d'utilisation	Cas d'utilisation du système, par extension il représente également la technique de modélisation mise en œuvre dans UML (use case).
Diagramme d'activité	Permet de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables.
Catégorie	Une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe, associée au concept UML de package. Ce concept permet une présentation plus synthétique du diagramme des classes d'un système réel.
Classe	Une classe définit un ensemble d'objets similaires potentiels. Elle fournit le modèle de la structure et les possibilités de chaque objet.
Objet	Un objet est une instance d'une classe, c'est une entité informatique unique possédant ses propres attributs et opérations
Opération ou méthode	Une opération est un traitement spécifique qu'un objet est en charge de fournir.
Tâche	Une tâche représente un élément manipulé par le système et ordonnançable de manière individuelle. Cela peut représenter un process d'un système Unix, une tâche d'un moniteur temps-réel, un thread d'une application.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	--	---

Tile	Zone spécifique du tileset représentant un élément visuel du jeu (par exemple un élève ou un professeur)
Tileset	Image composée des éléments graphiques statiques du jeu (la cour de récré, les bâtiments...)
Charset	Image composée des éléments graphiques dynamiques du jeu (les élèves)

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	--	---

2 Description et analyse de l'environnement

2.1 Organisation de l'espace de travail

Le logiciel sera réalisé sous Eclipse Oxygen for java Developpers. Le projet sera séparé en 2 grandes parties. D'un côté la création de la documentation, de l'autre côté le développement :

Projet Cour de Récréation :

→ Developpement

- Programmation Projet
- Tests

→ Documentation

→ Conception :

- Diagramme UML
- Javadoc

→ Tests :

- Fiche de Tests
- Compte rendu de test

3.2 Analyse statique

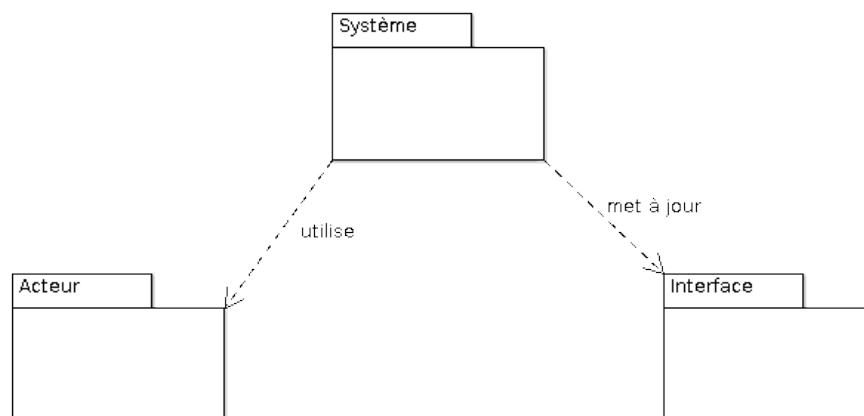


Illustration 2: Diagramme package

La structure du projet suit une architecture MVC :

- Les modèles étant les acteurs (package Acteur)
- Les vues étant de façon minoritaire l'affichage console (au final relayé au simple rôle de debugger) et surtout l'IHM (package Interface)
- Le contrôleur étant la classe Controller (package Systeme)

Package Acteur

Classe Eleve extends ElementdeJeu

Définition

Il s'agit de la classe représentant l'objet élève dans la cour de récréation.

Attributs

```
public static enum Direction {  
    UP,  
    DOWN,  
    LEFT,  
    RIGHT  
}
```

Il s'agit de la liste des directions possible d'un élève.

int actionEnCours → action de l'élève en cours

int punition → total des punitions de l'élève

boolean sexe → sexe de l'élève, fille ou garçon

int portee → portée de l'élève lorsqu'il se déplace

String nom → nom de l'élève

int visibilite → visibilité de l'élève lorsqu'il cherche un autre élève

Direction vers → direction vers laquelle se dirige l'élève pour pouvoir le dessiner dans le bon sens.

boolean mouvement → permet de savoir si l'élève est en mouvement ou non

Opérations

Int choisirAction() → renvoi un indice correspondant à l'action choisie par l'élève,

0 = bagarre

1 = bisous

2 = jouer

boolean verifPunition() → vérifie si l'élève à déplacer le maximum de punitions autorisées.

void majPunition() → incrémente le nombre total de punition de 1

Cell déplacement() → renvoi la case choisie par l'élève pour se déplacer, l'élève doit

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

se déplacer d'autant de case que sa portée lui permet, ou moins si les cases sont déjà occupées. La stratégie de déplacement est aléatoire pour un élève normal.

void initAction() → change l'indice de l'action en cours à -1, ce qui signifie qu'il n'y a aucune action en cours

void draw(Graphics g, int x, int y) → dessine l'élève en fonction de la case sur laquelle il se situe, de son action en cours ou alors de sa direction choisie.

Classe Eleve Turbulents extends Eleve

Définition

Il s'agit de la classe représentant un objet élève turbulent dans la cour de récréation. Un élève turbulent possède une stratégie de déplacement différente d'un élève normal. Un élève turbulent peut être romantique ou bagarreur.

Attributs

boolean estBagarreur → indique si l'élève est un bagarreur ou un romantique

Opérations

int choisirAction() → un élève bagarreur choisi systématiquement une bagarre, un élève romantique choisi systématiquement de faire un bisous.

Cell déplacement() → Un élève turbulent cherche autour de lui un autre élève, en fonction de sa visibilité. S'il trouve un élève il se dirige vers lui, sinon il se dirige aléatoirement

void draw() → Cette méthode à la même fonction que celle d'élève, seulement les élèves turbulents ont un aspect différent.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe Professeur extends ElementdeJeu

Définition

Il s'agit de la classe permettant de représenter un professeur dans la cour de récréation, c'est un élément de jeu.

Attributs

int patience → indice qui représente le taux de patience du professeur

int anciennete → indice qui représente l'ancienneté du professeur

String nom → nom du professeur

Opérations

boolean verifPatience() → Vérifie si le professeur à une patience au dessus de 0 ou non

boolean verifAnciennete() → Vérifie si l'ancienneté du professeur est en dessous de 20 ou non

void majPatience(int action) → Met à jour la patience du professeur selon le type de l'action choisie par l'élève dans son champ de vision

bagarre = -10

bisous = +1

void addAnciennete() → incrémente l'ancienneté du professeur de 1

void draw(Graphics g, int x, int y) → dessine le professeur à sa position

Classe Terrain extends ElementdeJeu

Définition

Le terrain correspond à une case vide dans la cour, c'est un élément du jeu.

Package Interface

Classe CharsetCR implements Tileset

Définition

Il s'agit de la classe permettant de gérer les charset. La classe CharsetCR implémente le design pattern Singleton.

Attributs

Static Image tilesetImg → correspond à l'image contenant tous les éléments à dessiner individuellement.

int li → Correspond au nombre de lignes du charset (6)

int co → Correspond au nombre de colonnes du charset (9)

int tileW → Correspond à la largeur d'un tile (60px)

int tileH → Correspond à la hauteur d'un tile (60px)

static Tile[][] → Correspond au tableau de tile

Opérations

Static CharsetCR getInstance() → Permet de parcourir le charset en fonction des lignes et des colonnes définies, et de sélectionner tous les tiles de largeur tileW et de hauteur tileH, afin de les ajouter au tableau Tile.

Tile getTelPersonnage() → Retourne le tile voulu présent dans la tableau Tile.

Classe Fenetre extends JFrame

Définition

Il s'agit de la classe permettant de créer une fenêtre de jeu.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe FenetreParametres extends JFrame

Définition

Il s'agit de la classe permettant de créer la fenêtre de sélection de paramètre. Dans cette fenêtre l'utilisateur choisi le nombre d'élèves normaux, le nombre d'élèves turbulents et le nombre de professeurs.

Classes internes de FenetreParametres

private class Panel extends JPanel implements ActionListener

Définition :

Permet d'afficher les limites des paramètres en haut de la fenêtre paramètre
Attributs :

Attributs :

Image profimg → permet d'avoir un tile professeur au-dessus du TextField qui permet d'entrer le nombre de professeurs désiré

Image enfantimg1 → permet d'avoir un tile élève au-dessus du TextField qui permet d'entrer le nombre d'élèves désirés

Image enfantimg2 → permet d'avoir un tile élève bagarreur au-dessus du TextField qui permet d'entrer le nombre de d'élèves turbulents désiré

Opérations

public void paintComponent(Graphics g) → dessine sur la fenêtre paramètre un message concernant les limites des paramètres à l'intention de l'utilisateur et dessine 3 tiles aux dessus des champs paramètres

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

private class Panel_Parameters extends Jpanel

Définition:

permet d'afficher les JTextField dans la fenêtre paramètres, afin que l'utilisateur puisse entrer les nombres souhaités.

Attributs:

JTextField nb_eleves → Champ permettant de récupérer le nombre d'élèves entré par l'utilisateur.

JTextField nb_profs → Champ permettant de récupérer le nombre de professeurs entré par l'utilisateur.

JTextField nb_elevesturbu → Champ permettant de récupérer le nombre d'élèves turbulents entré par l'utilisateur.

JLabel eleve_label → affiche l'information « nombre d'élèves » à coté du JTextField nb_eleves

JLabel prof_label → affiche l'information « nombre de professeurs » à coté du JTextField nb_profs

JLabel eleveturbu_label → affiche l'information « nombre d'élèves turbulents » à coté du JTextField nb_elevesturbu

JButton lancer → Au clique de ce bouton, la partie se lance en gérant actionPerformed

Opérations:

public void actionPerformed(ActionEvent arg0) → Lors du clique du bouton lancer, on récupère la source ici, et on récupère dans des int les paramètres entrés, on modifie les variables globales et enfin on utilise setInit() et setGameStarted() pour lancer la partie. De plus, on vérifie que les paramètres ne sont pas négatifs, que le nombre d'élèves soit inférieur ou égal à 60 et le nombre de nombre à 20.

En cas d'erreur on affiche une fenêtre popup.

Fin des classes internes

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe ConsolePanneau extends JPanel implements ActionListener

Définition

Il s'agit de la classe permettant de d'afficher deux boutons, sur la fenêtre de la map, qui permettent de gérer la vitesse du jeu en cours.

Attributs

Jbutton accélérer → bouton qui accélère la vitesse du jeu lors d'un clique. Chaque clique enlève 100 à la variable vitesse,

Jbutton ralentir → bouton qui ralentis la vitesse du jeu lors d'un clique. Chaque clique ajoute 100 à la variable vitesse

Opérations

Public void actionPerformed(ActionEvent e) → permet de accélérer/ralentir le jeu en gérant la variable «vitesse» définit dans la classe controller au clique de l'un des boutons.

Classe FenetrePopup extends JFrame

Définition

Il s'agit de la classe permettant de créer une fenêtre qui va informer l'utilisateur que les paramètres entrés ne respectent pas les limites.

Classe interne de FenetrePopup

private class Panel_popup extends JPanel

Définition

Il s'agit de la classe permettant de créer un panneau qui va s'afficher sur la fenêtrePopup

Opérations :

public void paintComponent(Graphics g) → affiche une image et un message d'erreur sur le panneau

Fin des classes internes

Ref : CPT-1 Emetteur : POIJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe Map extends JPanel

Définition

Il s'agit de la classe permettant de dessiner la cour de récréation ainsi que tous les élèves et les professeurs à chaque tour.

Attributs

Controller ctrl → Il s'agit d'une instance du controller

Opérations

void paintComponent(Graphics g) → dessine sur chaque case de la cour l'élément qui correspond au sol de la cour. Dessine aussi chaque élève et professeur à leur position en faisant appel à la fonction « draw » interne aux classes correspondantes. Les listes d'élèves et de professeurs sont récupérées par l'instance du controller.

Classe Tile

Définition

Il s'agit de la classe correspondant à l'objet tile d'un tileset.

Attributs

Tileset ts → Le tileset dont est issu le Tile

int numPosX → numéro de position de la ligne du tile dans le tileset

int numPosY → numéro de position de la colonne du tile dans le tileset

int width → largeur d'un tile (60px)

int height → hauteur d'un tile (60px)

Opérations

void drawTile(Graphics g, int x, int y, Object o) → dessine le tile à la position donnée

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Interface TileSet

Définition

Interface des tiles permettant de récupérer les paramètres d'un tile

Classe TilesetCR implements Tileset

Définition

Il s'agit de la classe permettant de gérer un tileset. Ici il s'agira du sol de la cour de récréation. La classe TilesetCR implémente le design pattern Singleton.

Attributs

Image tilesetImg → L'image contenant le tile
int li → nombre de lignes du tileset
int co → nombre de colonnes du tileset
int tileW → largeur du tile (60px)
int tileH → hauteur du tile (60px)
static Tile[][] → tableau de tileset

Opérations

Static TilesetCR getInstance() → retourne une instance d'un tileset qui a été lu et découpé en fonction du nombre de lignes et de colonnes.
Tile getTelTile() → retourne tel tile.

Package Systeme

Classe Cell

Définition

Il s'agit de la classe qui représente une cellule de la grille de jeu et son contenu.

Attributs

boolean empty → indique si la case est vide ou non, une case est pleine si elle est occupée par un élève ou un professeur.

ElementdeJeu content → Enregistre le contenu de la case, il peut s'agir d'un Terrain, d'un Eleve, d'un EleveTurbulent ou d'un Professeur.

Classe Constant

Définition

Il s'agit de la classe qui enregistre toutes les constantes utiles et importantes dans le projet.

Attributs

int mapWidth → la largeur de la map

int mapHeight → la hauteur de la map

String pathTelChemin → le chemin pour obtenir telle ou telle image, d'un tileset ou charset par exemple.

Interface Drawable

Définition

Il s'agit de l'interface qui indique si élément est dessinnable ou non

Opérations

Void draw(Graphics g, int x, int y) → fonction abstraite définie dans des éléments dessinnable (Par exemple un élèves ou un professeur est dessinnable)

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe ElementdeJeu implements Drawable

Définition

Il s'agit de la classe qui représente une élément du jeu.

Attributs

int pos_X → Position en abscisse de l'élément de jeu.
int pos_Y → Position en ordonnée de l'élément de jeu.

Classe Grille

Définition

Il s'agit de la classe permettant de définir la grille du jeu

Attributs

int li → nombre de lignes de la grille
int co → nombre de colonnes de la grille
Cell cells[][] → tableau comportant chaque cellule de la grille

Classe Controller

Définition

Le controller est l'élément principal du jeu, c'est lui qui initialise la cour et ses éléments, il permet l'interaction entre les différents objets du projet. La classe Controller implémente le design pattern Singleton car il n'y a besoin que d'une seule instance du contrôleur pour gérer le déroulement du jeu.

Attributs

Static Controller INSTANCE → instance unique du controller
int indiceListeEleves → indice permettant de parcourir les listes d'élèves
int vitesse → permet de gérer la vitesse du jeu
List<Eleve> eleves → liste de tous les élèves normaux en jeu
List<EleveTurbulent> elevesT → liste de tous les élèves turbulents en jeu.
List<Professeur> professeurs → liste de tous les professeurs en jeu.
Fenetre fenetre → fenetre de jeu
Map map → la map du jeu

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Grille grille → la grille du jeu

int nb_Professeurs → le nombre de professeurs en jeu

int nb_Eleves → le nombre d'élèves en jeu

int nb_ElevesTurbulent → le nombre d'élèves turbulents en jeu

static int nbMaxEleve → le nombre maximum d'élèves possible

static int nbMaxEleveT → le nombre maximum d'élèves possible

static nbMaxProf → le nombre maximum de professeurs possible

Opérations

Static Controller getInstance() → retourne l'instance du controller, permet de créer une instance unique du controller.

void placerEleve(int li, int co) → place un élève aléatoirement sur la grille

void placerEleveT(int li, int co) → place un élève turbulent aléatoirement sur la grille

void placerProfesseur(int li, int co) → place un professeur aléatoirement sur la grille

void exclude(Eleve eleve) → supprime l'élève en paramètre du jeu

void burnout(Professeur prof) → supprime le professeur en paramètre du jeu

void init() → initialise le jeu en demandant le nombre d'élèves et de professeur, en les plaçant tous sur la grille et en lançant la fenêtre de jeu.

ArrayList<Professeur> recherProf(Eleve e) → recherche tous les professeurs autour de l'élève en paramètre et renvoi la liste de ces professeurs.

void traitementAction(Eleve e, int action) → traite l'action choisie par l'élève en vérifiant la présence de professeur autour de lui. S'il y avait un professeur la patience de celui est mise à jour ainsi que le nombre de punitions de l'élève. Si la patience du professeur est en dessous de 0 il part en burnout.

void tourSuivant() → Sélectionne l'élève suivant dans la liste de tous les élèves. Le fait se déplacer puis choisir une action et traite l'action choisie. Si l'élève choisi à plus de 3 punitions il se fait exclure.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Package Lancement

Classe Client

Définition

C'est ici que le programme se lance

Attribut

Launcher *launcher* → permet de lancer le programme

Opérations

public static void main(String[] args) throws InterruptedException, IOException
 → exécute le code en initialisant l'attribut launcher en un objet Graphique et en lui appliquant la méthode launch()

Classe abstraite Launcher

Opérations

protected abstract void startGame() throws InterruptedException, IOException
 → Lance la partie en utilisant la méthode tourSuivant() si il n'y a pas de « gameover »

public abstract void launch() throws InterruptedException, IOException
 → créer une fenêtreParamètre et lance le jeu.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Classe Graphique extends Launcher

Attribut

Graphique *INSTANCE* → l'instance de graphique

Opérations

public static Graphique getInstance() → permet d'avoir une seule instance de graphique.

protected void startGame() throws InterruptedException, IOException → Lance la partie en utilisant la méthode tourSuivant() si il n'y a pas de « gameover »

public void launch() throws InterruptedException, IOException → créer une fenêtreParamètre et lance le jeu.

! pour toutes les classes, les opérations de type constructeur, setter et getter n'ont pas été précisées mais sont évidemment nécessaires.

Notation diagramme d'activités

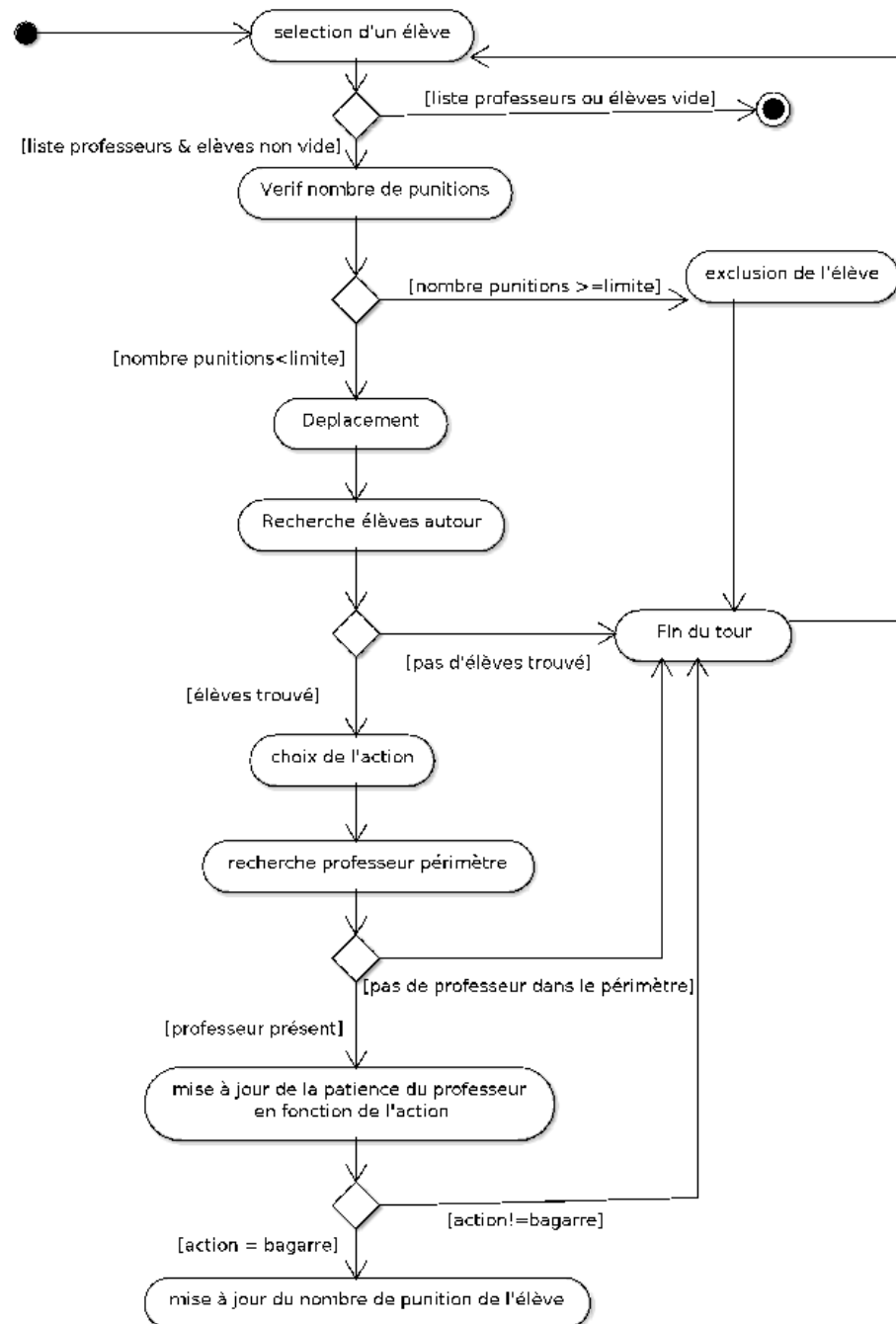


Illustration 3: Diagramme activité

4 Aspects dynamiques

4.1 Echanges entre éléments

Ci-dessous certains scénarios importants qui mettent en œuvre des échanges entre des éléments.

Scénario 1

Diagramme de séquence : Scénario « Déplacement d'un élève »

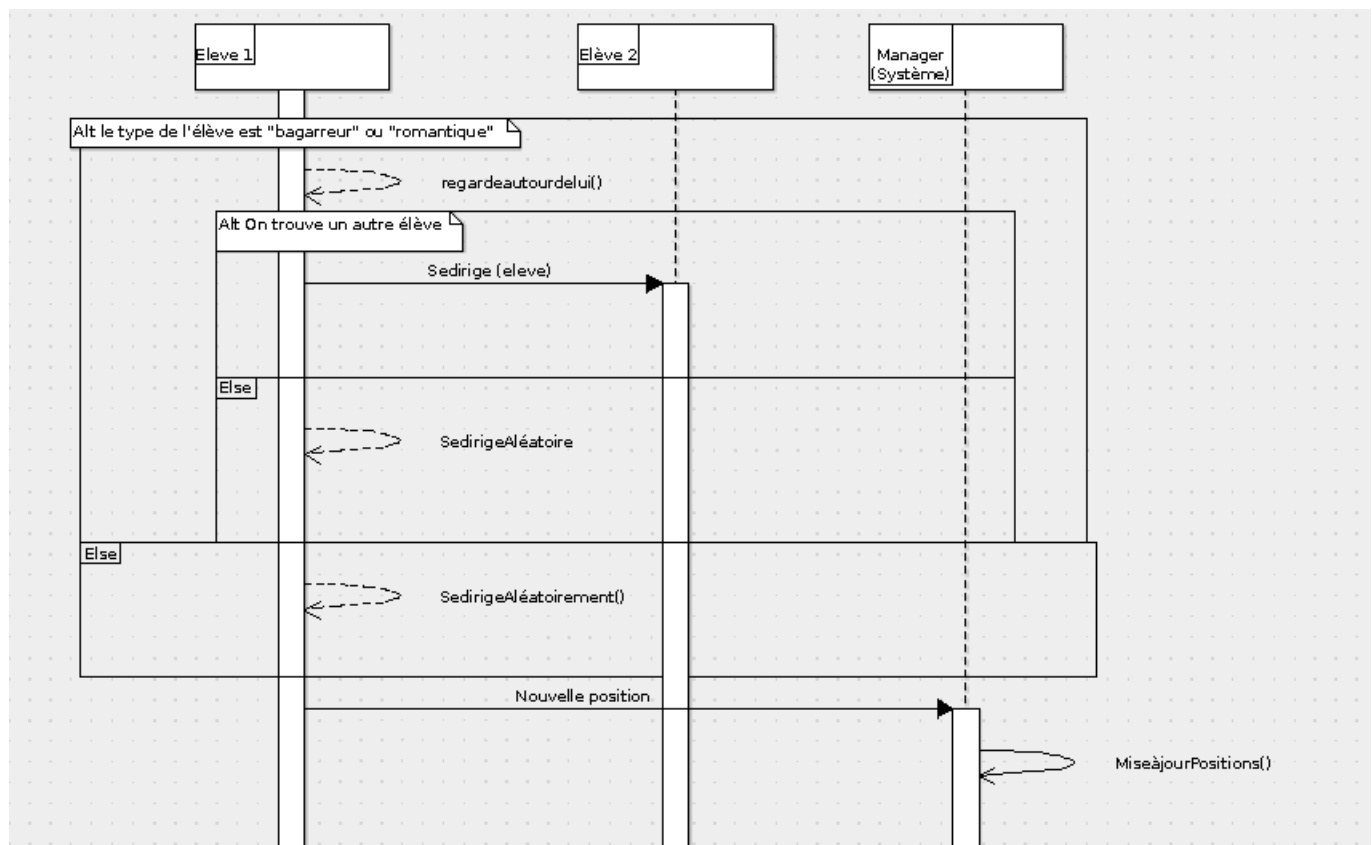


Illustration 4: Diagramme séquence

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

Scénario 2

Diagramme de séquence : Scénario « L'élève choisi une action »

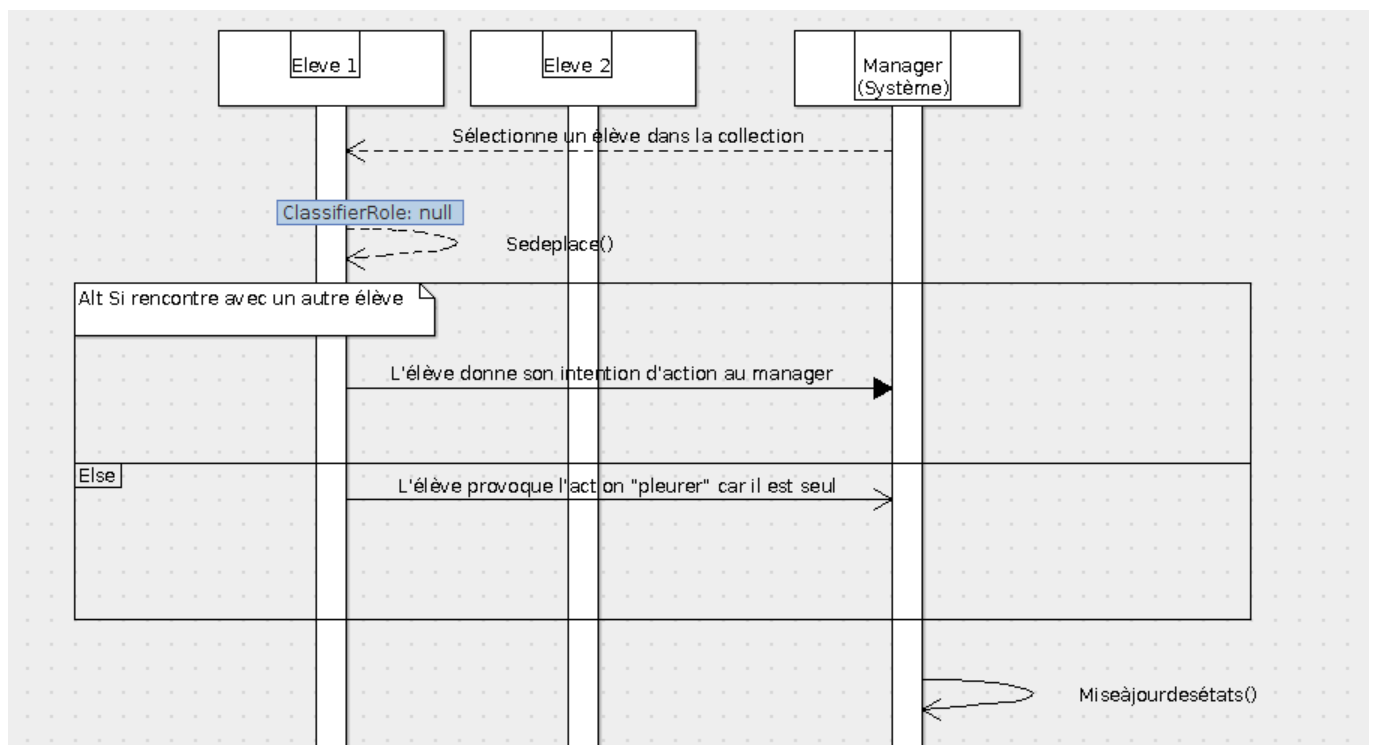


Illustration 5: Diagramme séquence 2

4.2 Cas d'erreurs

Affichage d'une popup

- Si l'utilisateur entre un nombre d'élèves et d'élèves turbulents d'on la somme est supérieur à 60.
- Si l'utilisateur donne un nombre de professeur supérieur à 20.
- Lorsque l'utilisateur ne rentre aucune donnée ou oublie de remplir l'un des 3 champs.
- Si les nombres insérés sont négatifs

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

5 Interface graphique

5.1 La fenêtre de paramètres

Ci-dessous, ce à quoi la fenêtre de paramètres doit ressembler.



Illustration 6: Interface paramètre

Cette fenêtre permet d'entrer les paramètres du jeu, notamment le nombre d'élèves, le nombre d'élèves turbulents, le nombre de professeurs

Ces paramètres sont entrés dans 3 TextFields avec chacun un JLabel correspondant à chaque paramètre.

Les limites sont :

60 élèves maximum et 20 professeurs maximum, ces limites sont gérées en cas d'erreurs dans la section «5.2 cas d'erreurs»

Le JButton « Lancer la partie » permet de récupérer les paramètres entrés précédemment et de lancer l'affichage de la map.

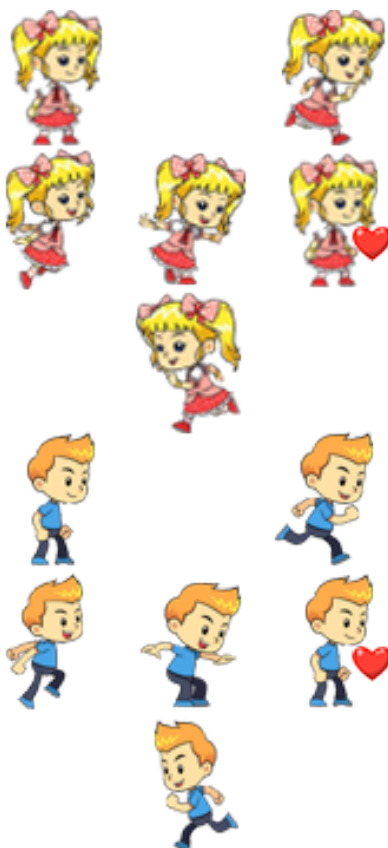
5.2 Les acteurs

Les élèves :

Les turbulents



Les romantiques



Les normaux



Illustration 7: Charset

Ci-dessus le charset représentant tous les enfants. Il s'agit d'un png à fond transparent.

Pour chaque enfant nous avons de gauche à droite : La position debout sans action ni déplacement, pour ceux qui le peuvent la bagarre, la direction à droite. Puis en dessous toujours de gauche à droite, la direction en haut, la direction en bas, pour ceux qui le peuvent le bisous. Enfin, en troisième ligne, nous avons pour ceux qui le peuvent la position bagarre et enfin la direction gauche.

Nom élève



Pour chaque élève, son nom doit être présent en bleu turquoise au-dessus de sa tête. Son quota de punition est représenté par des cœurs png à fond transparent.

Les professeurs :



Nom de prof

Tous les professeurs seront représentés par cette image png à fond transparent. Son nom doit être présent sous lui en vert clair, sa jauge de patience au dessus-de sa tête.

5.3 La fenêtre de jeu

Ci-dessous, ce à quoi la fenêtre de la grille de jeu doit ressembler



Illustration 8: Interface jeu

La map est composée de 150 cases (15 cases en largeur et 10 en hauteur)

Le lancement du jeu se fait par l'intermédiaire de la touche « entrée » du clavier.

L'utilisateur peut s'il le souhaite accélérer le jeu ou le ralentir par l'intermédiaire de 2 JButton implémentant l'interface java ActionListener et utilisant celle-ci pour modifier la vitesse (en utilisant la méthode setVitesse()) en cliquant sur l'un des boutons.

Ref : CPT-1 Emetteur : POUJOL Teddy BELLIN Clara Client : Sebastien Mavromatis Projet : Cour de récréation	Cour de récréation Conception	Date : 07 février 2019 version : 0.7 Service : École État : Finalisation
---	----------------------------------	---

5.4 Fin du jeu

Cas les professeurs ont perdu



Illustration 9: Interface professeurs perdent

Dans le cas où un professeur perd la totalité de ses points de patience, ce professeur est supprimé de la liste des professeurs et n'apparaît plus sur la map au tour suivant car la méthode `tourSuivant()` dessine à chaque tour les professeurs inclus dans la liste de professeurs. Si la liste est vide, il n'y a plus aucun professeur sur la map et le jeu s'arrête, un message apparaît « Les professeurs ont perdus ».

L'utilisateur doit maintenant fermer la fenêtre pour arrêter le jeu.

