Projet Lifap7 : Gyromite

Haowei Wang
Fatmana Altay

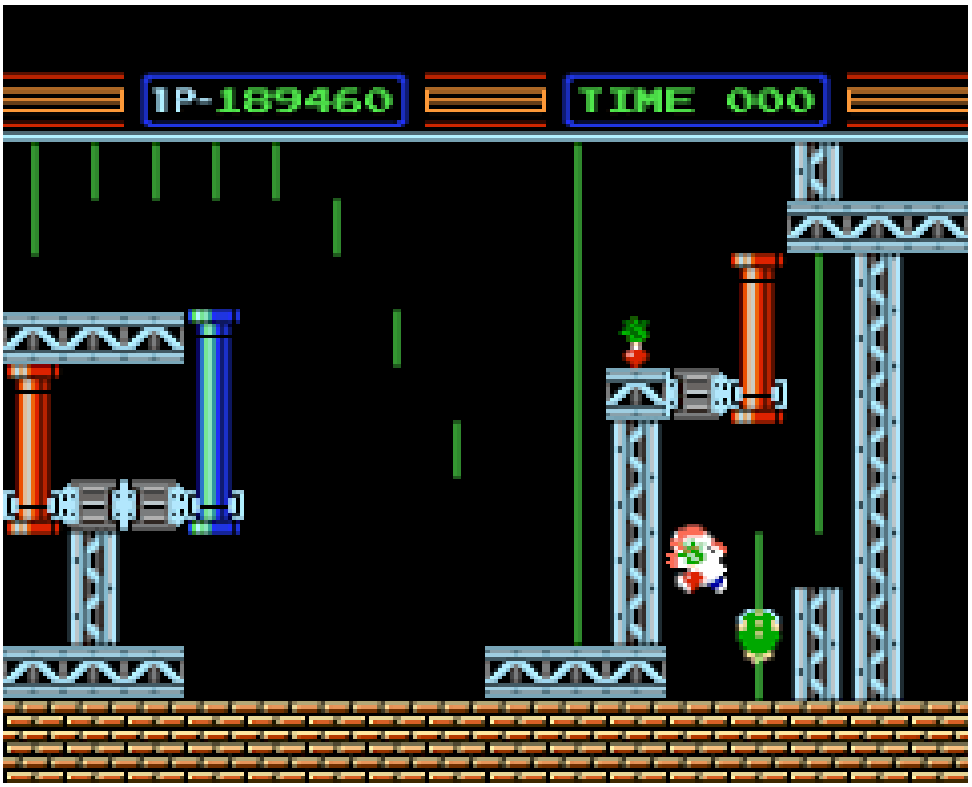


Table des matières

Fonctionnalité

Gestion du Jeu

- Déplacer le professeur grâce au code fournit.

- Ajout du déplacement avec la corde, maintenant le professeur est capable de monter et descendre avec la corde. Environ

-Ajout des déplacements des colonnes verticales et le professeur se déplace en même temps lorsqu'il est sur une colonne. Nous avons décidé que ce serait le joueur qui déplacerait les colonnes. Environ

-Gestion de la gravité, le professeur tombe lorsqu'il n'a rien sous les pieds. Environ

-Implémentation de l'IA qui permet au smick de se déplacer sans interventions du joueur.

-Gestion de la collision :

-Dans un premier temps, le professeur ou un smick avec le mur.

-Puis gestion des collisions entre un smick et le professeur. Le professeur retourne à sa position de début de partie lorsqu'il est touché par son ennemi.

-Le professeur avec ce qu'on appelle les « Ramassables ». Le professeur a la capacité à ramasser les bonus et les bombes. Une fois qu'il les a ramassés, ceux-ci n'apparaissent plus graphiquement dans le jeu.

-Entre les smicks ou le professeur avec les colonnes. Les entités dynamiques se font écrasée par les colonnes.

-Ajout de règles du jeu :

- 5 vies en début de partie, on perd une vie quand il y a une collision entre un smick et le professeur. Si on perd les 5 vies sont utilisées, le joueur perdus et la partie est terminée.

- Compteur pour les bombes ramassées. Si le jouer à ramasser toutes les bombes alors il a gagné et la partie est terminée.

- Compteur pour les bonus. Chaque fois que le professeur ramasse un bonus on lui attribue 10 points.

-Implémentation d'un chronomètre qui détermine le temps de jeu du joueur. Il doit gagner avant que le chronomètre ne se termine, sinon il perd et la partie est terminée.

Gestion des fenêtres

- Implémentation :

- Une fenêtre de début,
  - Pour déterminer le nombre de joueurs.
  - Affichage les contrôleurs du jeu.
  - Sélectionner niveau

  (Environ 3h)

- La fenêtre de jeu à laquelle nous avons ajoutés au haut dessus de la grille de jeu :
  - Affichage du chronomètre.
  - Affichage du nombre de vies.
  - Affichage des points gagnés.
  - Affichage du nombre de smick restants.
  - Affichage du nombre de bonus restants.

  (Environ 3h)

- Une fenêtre de fin de jeu
  - Affichage lorsque le joueur a perdu ou gagné.
  - Affichage du meilleur score. (Environ 2h)
  - Un bouton pour recommencer le jeu. (Environ 1h)

Liste des extensions

- Ajout d'un deuxième joueur. (Environ 3h)

- Une variante du niveau scrollable. C'est à dire que le joueur peut agrandir rapetisser la fenêtre lui-même. (Environ 1h)

- Ajout de colonnes avec déplacement horizontal. (Environ 2h)

- Implémentation d'un chronomètre. (Environ 2h)

- Fonction pour redémarrer le jeu. (Environ 1h)

- Plusieurs niveaux, avec la fonction switch case. Faire la choisir sur la FenetreDebut. (Environ 2h)

- Le meilleur score. Afficher le meilleur score et enregistrer le meilleur score. (Environ 2h)

- Emballez tous les fichiers requis dans jar et ne lisez et n'écrivez que via le package jar. (Environ 3h)

Documentation UML

Choix de Conception

Pour le projet, un code de départ nous a été fourni avec la plupart des fonctionnalités de base (représentation des données avec des tableaux deux dimensions, les Classes Jeu et VueControleurGyromite qui sont le cœur du projet, et le déplacement des entités dynamiques).

Le projet est structuré selon le modèle MVC (Modèle Vue Contrôleur) que nous avons gardé.

Les classes qui implémentent l'interface « Runnable » sont des classes qui représentent des threads particuliers pour le jeu.

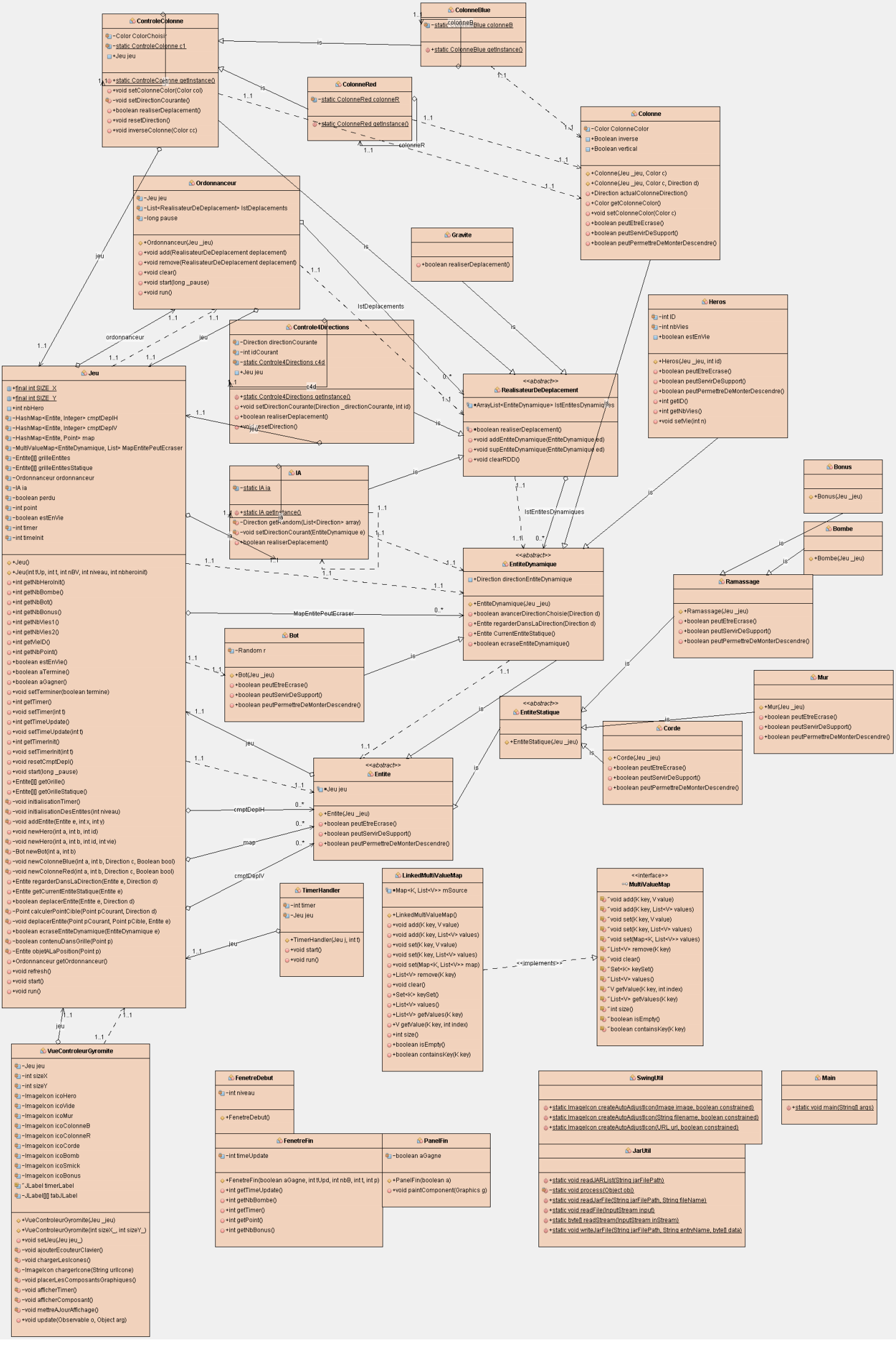
La classe Jeu a été modifiée pour implémenter l'interface « Runnable ». Elle représente le thread principal qui gère les données (notamment les données pour les règles du jeu) et c'est la classe VueControleurGyromite qui se charge du rafraîchissement de l'affichage.

Nous avons ensuite ajouté un thread pour la classe TimerHandler qui est le timer du jeu.

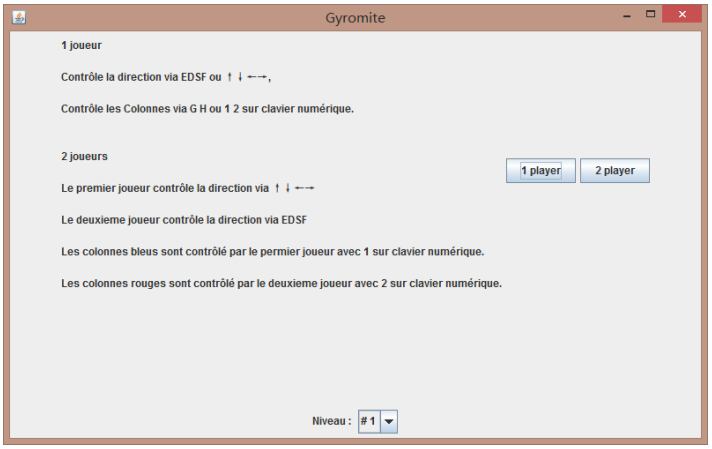
Les classes qui ont été données au départ ont été modifiée, notamment la structure pour stocker et retrouver les entités dynamiques. Nous avons modifié les grilles deux dimension et le système des HashMap avec des structures ressemblantes modélisées par les classes MultiValueMap et LinkedMultiValueMap. Ce changement a été fait pour mieux gérer les suppressions d'entité lors de collisions.

Ensuite, nous avons bien évidemment rajouté des attributs/fonctions/procédures membres à certaines classes pour faire évoluer le jeu. Nous avons aussi ajouté d'autres classes (Bot, Rammassages, FenetreFin, FenetreDebut...)

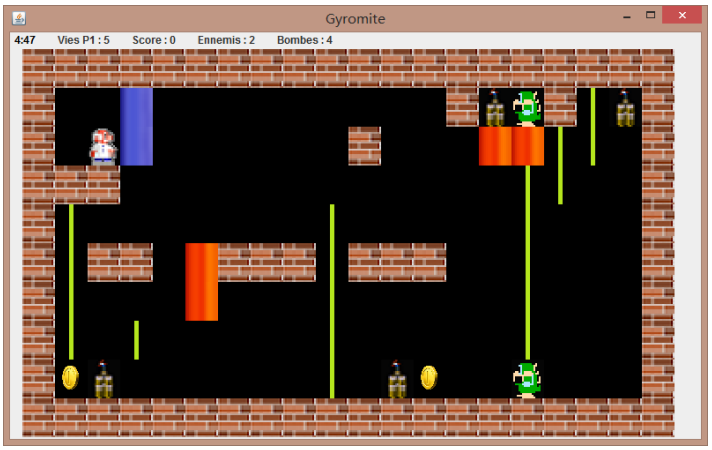
Diagramme UML

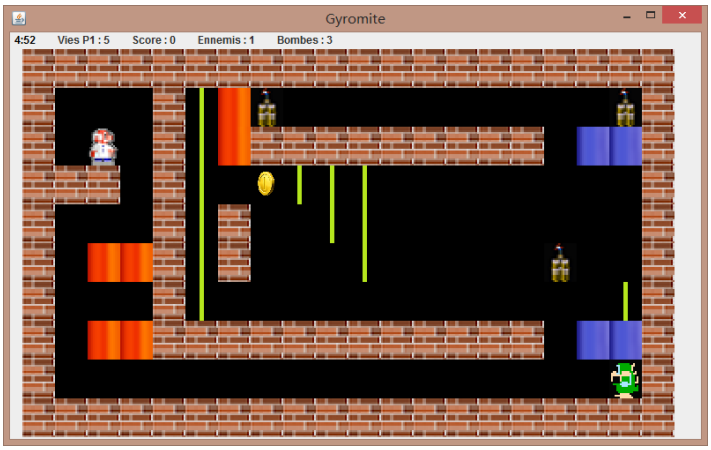

ControleColonne
-Color ColorChoisi
-static ControleColonne c1
+Jeu jeu
+static ControleColonne getInstance()
+void setColonneColor(Color col)
-void setDirectionCourante()
+boolean realiserDeplacement()
+void resetDirection()
+void inverseColonne(Color cc)
ColonneBlue
-static ColonneBlue colonneB
+static ColonneBlue getInstance()
ColonneRed
-static ColonneRed colonneR
+static ColonneRed getInstance()
Colonne
-Color ColonneColor
+Boolean inverse
+Boolean vertical
+Colonne(Jeu _jeu, Color c)
+Colonne(Jeu _jeu, Color c, Direction d)
+Direction actualColonneDirection()
+Color getColonneColor()
+void setColonneColor(Color c)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
Ordonnanceur
-Jeu jeu
-List<RealisateurDeDeplacement> lstDeplacements
-long pause
+Ordonnanceur(Jeu _jeu)
+void add(RealisateurDeDeplacement deplacement)
+void remove(RealisateurDeDeplacement deplacement)
+void clear()
+void start(long _pause)
+void run()
Gravite
+boolean realiserDeplacement()
Heros
-int ID
-int nbVies
+boolean estEnVie
+Heros(Jeu _jeu, int id)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
+int getID()
+int getNbVies()
+void setVie(int n)
Controle4Directions
-Direction directionCourante
-int idCourant
-static Controle4Directions c4d
+Jeu jeu
+static Controle4Directions getInstance()
+void setDirectionCourante(Direction _directionCourante, int id)
+boolean realiserDeplacement()
+void resetDirection()
<<abstract>>
RealisateurDeDeplacement
#ArrayList<EntiteDynamique> lstEntitesDynamiques
#boolean realiserDeplacement()
+void addEntiteDynamique(EntiteDynamique ed)
+void supEntiteDynamique(EntiteDynamique ed)
+void clearRDD()
Jeu
+final int SIZE_X
+final int SIZE_Y
+int nbHero
-HashMap<Entite, Integer> cmptDeplH
-HashMap<Entite, Integer> cmptDeplV
-HashMap<Entite, Point> map
-MultiValueMap<EntiteDynamique, List> MapEntitePeutEcraser
-Entite[][] grilleEntites
-Entite[][] grilleEntitesStatique
-Ordonnanceur ordonnanceur
-IA ia
-boolean perdu
-int point
-boolean estEnVie
-int timer
-int timeInit
+Jeu()
+Jeu(int tUp, int t, int nBV, int niveau, int nbheroInit)
+int getNbHeroInit()
+int getNbBombe()
+int getNbBot()
+int getNbBonus()
+int getNbVies1()
+int getNbVies2()
+int getVielD()
+int getNbPoint()
+boolean estEnVie()
+boolean aTermine()
+boolean aGagner()
+void setTerminer(boolean termine)
+int getTimer()
+void setTimer(int t)
+int getTimeUpdate()
+void setTimeUpdate(int t)
+int getTimerInit()
+void setTimerInit(int t)
+void resetCmptDepl()
+void start(long _pause)
+Entite[][] getGrille()
+Entite[][] getGrilleStatique()
-void initialisationTimer()
-void initialisationDesEntites(int niveau)
-void addEntite(Entite e, int x, int y)
+void newHero(int a, int b, int id)
-void newHero(int a, int b, int id, int vie)
-Bot newBot(int a, int b)
-void newColonneBlue(int a, int b, Direction c, Boolean bool)
-void newColonneRed(int a, int b, Direction c, Boolean bool)
+Entite regarderDansLaDirection(Entite e, Direction d)
+Entite getCurrentEntiteStatique(Entite e)
+boolean deplacerEntite(Entite e, Direction d)
-Point calculerPointCible(Point pCourant, Direction d)
-void deplacerEntite(Point pCourant, Point pCible, Entite e)
+boolean ecraseEntiteDynamique(EntiteDynamique e)
-boolean contenuDansGrille(Point p)
-Entite objetALaPosition(Point p)
+Ordonnanceur getOrdonnanceur()
+void refresh()
+void start()
+void run()
IA
-static IA ia
+static IA getInstance()
-Direction getHasard(List<Direction> array)
-void setDirectionCourant(EntiteDynamique e)
+boolean realiserDeplacement()
<<abstract>>
EntiteDynamique
+Direction directionEntiteDynamique
+EntiteDynamique(Jeu _jeu)
+boolean avancerDirectionChoisie(Direction d)
+Entite regarderDansLaDirection(Direction d)
+Entite CurrentEntiteStatique()
+boolean ecraseEntiteDynamique()
Bonus
+Bonus(Jeu _jeu)
Bombe
+Bombe(Jeu _jeu)
Ramassage
+Ramassage(Jeu _jeu)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
Bot
-Random r
+Bot(Jeu _jeu)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
<<abstract>>
EntiteStatique
+EntiteStatique(Jeu _jeu)
Mur
+Mur(Jeu _jeu)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
Corde
+Corde(Jeu _jeu)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
<<abstract>>
Entite
#Jeu jeu
+Entite(Jeu _jeu)
+boolean peutEtreEcrase()
+boolean peutServirDeSupport()
+boolean peutPermettreDeMonterDescendre()
LinkedMultiValueMap
#Map<K, List<V>> mSource
+LinkedMultiValueMap()
+void add(K key, V value)
+void add(K key, List<V> values)
+void set(K key, V value)
+void set(K key, List<V> values)
+void set(Map<K, List<V>> map)
+List<V> remove(K key)
+void clear()
+Set<K> keySet()
+List<V> values()
+V getValue(K key, int index)
+List<V> getValues(K key)
+int size()
+boolean isEmpty()
+boolean containsKey(K key)
<<implements>>
<<interface>>
MultiValueMap
~void add(K key, V value)
~void add(K key, List<V> values)
~void set(K key, V value)
~void set(K key, List<V> values)
~void set(Map<K, List<V>> values)
~List<V> remove(K key)
~void clear()
~Set<K> keySet()
~List<V> values()
~V getValue(K key, int index)
~List<V> getValues(K key)
~int size()
~boolean isEmpty()
~boolean containsKey(K key)
TimerHandler
-int timer
-Jeu jeu
+TimerHandler(Jeu j, int t)
+void start()
+void run()
VueControleurGyromite
-Jeu jeu
-int sizeX
-int sizeY
-ImageIcon icoHero
-ImageIcon icoVide
-ImageIcon icoMur
-ImageIcon icoColonneB
-ImageIcon icoColonneR
-ImageIcon icoCorde
-ImageIcon icoBomb
-ImageIcon icoSmick
-ImageIcon icoBonus
~JLabel timerLabel
-JLabel[][] tabJLabel
+VueControleurGyromite(Jeu _jeu)
+VueControleurGyromite(int sizeX_, int sizeY_)
+void setJeu(Jeu jeu_)
-void ajouterEcouteurClavier()
-void chargerLesIcones()
-ImageIcon chargerIcone(String urlIcone)
-void placerLesComposantsGraphiques()
-void afficherTimer()
-void afficherComposant()
-void mettreAJourAffichage()
+void update(Observable o, Object arg)
FenetreDebut
-int niveau
+FenetreDebut()
FenetreFin
-int timeUpdate
+FenetreFin(boolean aGagne, int tUpd, int nbB, int t, int p)
+int getTimeUpdate()
+int getNbBombe()
+int getTimer()
+int getPoint()
+int getNbBonus()
PanelFin
-boolean aGagne
+PanelFin(boolean a)
+void paintComponent(Graphics g)
SwingUtil
+static ImageIcon createAutoAdjustIcon(Image image, boolean constrained)
+static ImageIcon createAutoAdjustIcon(String filename, boolean constrained)
+static ImageIcon createAutoAdjustIcon(URL url, boolean constrained)
JarUtil
+static void readJARList(String jarFilePath)
-static void process(Object obj)
+static void readJarFile(String jarFilePath, String fileName)
+static void readFile(InputStream input)
+static byte[] readStream(InputStream inStream)
+static void writeJarFile(String jarFilePath, String entryName, byte[] data)
Main
+static void main(String[] args)
is
jeu
ordonnanceur
lstDeplacements
c4d
map
cmptDeplH
cmptDeplV
MapEntitePeutEcraser
lstEntitesDynamiques
colonneR
1..1
0..*

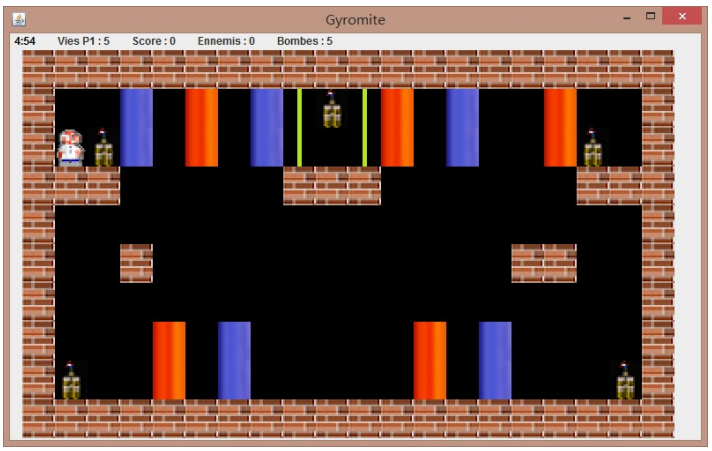
Annexe

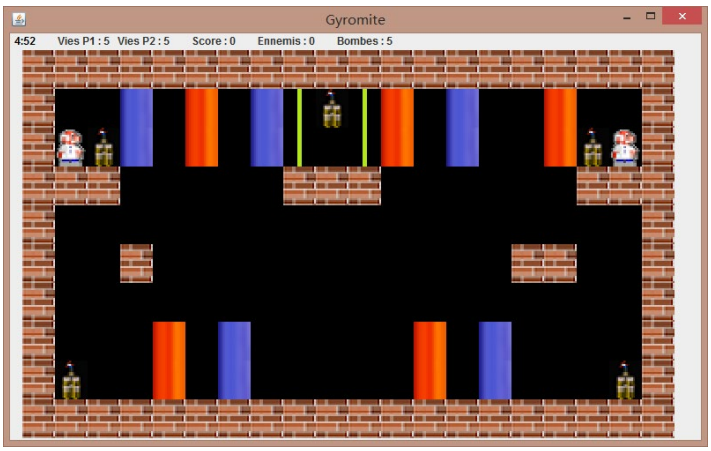

Gyromite
1 joueur
Contrôle la direction via EDSF ou ↑ ↓ ←→,
Contrôle les Colonnes via G H ou 1 2 sur clavier numérique.
2 joueurs
1 player
2 player
Le premier joueur contrôle la direction via ↑ ↓ ←→
Le deuxieme joueur contrôle la direction via EDSF
Les colonnes bleus sont contrôlé par le permier joueur avec 1 sur clavier numérique.
Les colonnes rouges sont contrôlé par le deuxieme joueur avec 2 sur clavier numérique.
Niveau : # 1


Partie Terminée !
Score : 0 Meilleur score : 20
Recommencer
Vous avez perdu!
Partie Terminée !
Score : 170 Meilleur score : 170
Recommencer
Vous avez gagné!


Gyromite
4:47 Vies P1 : 5 Score : 0 Ennemis : 2 Bombes : 4


Gyromite
4:52 Vies P1 : 5 Score : 0 Ennemis : 1 Bombes : 3


Gyromite
4:54 Vies P1 : 5 Score : 0 Ennemis : 0 Bombes : 5


Gyromite
4:52 Vies P1 : 5 Vies P2 : 5 Score : 0 Ennemis : 0 Bombes : 5