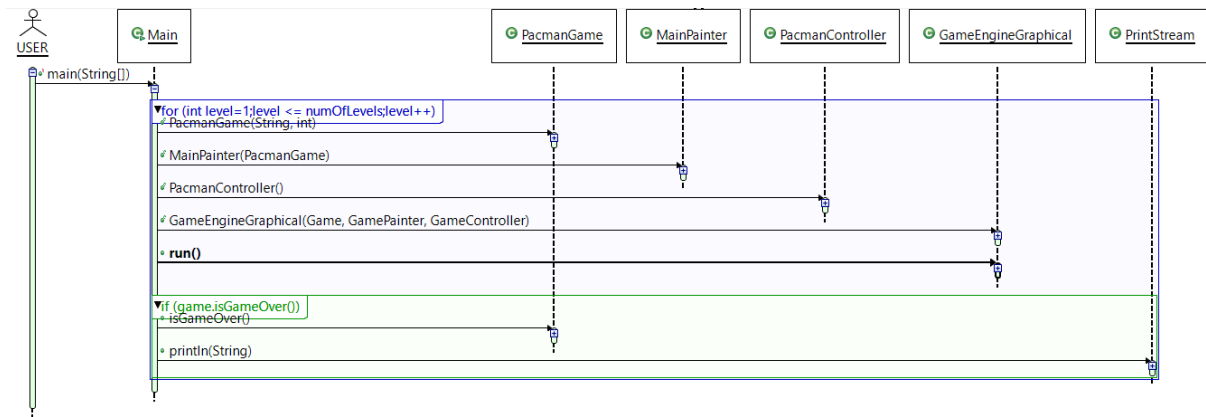
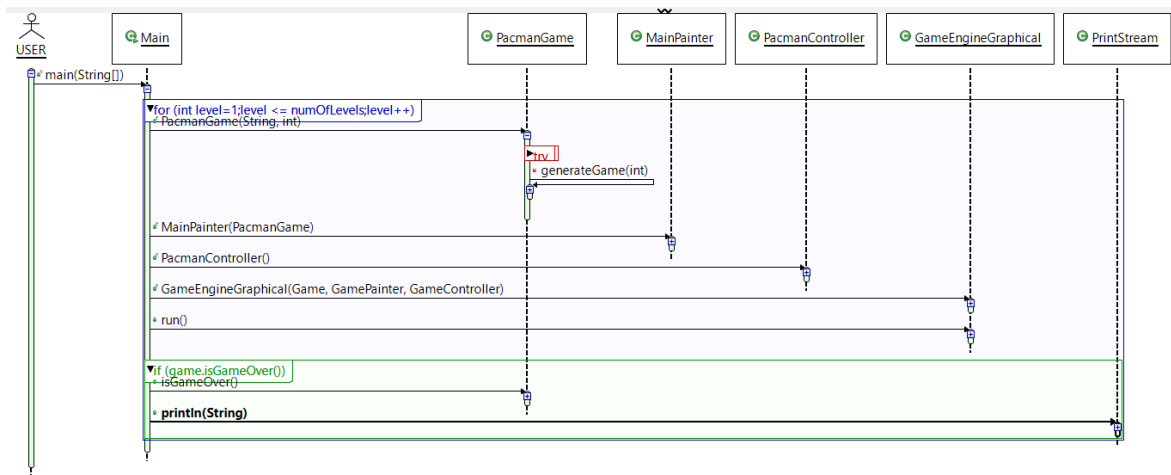


Le diagramme séquence général:

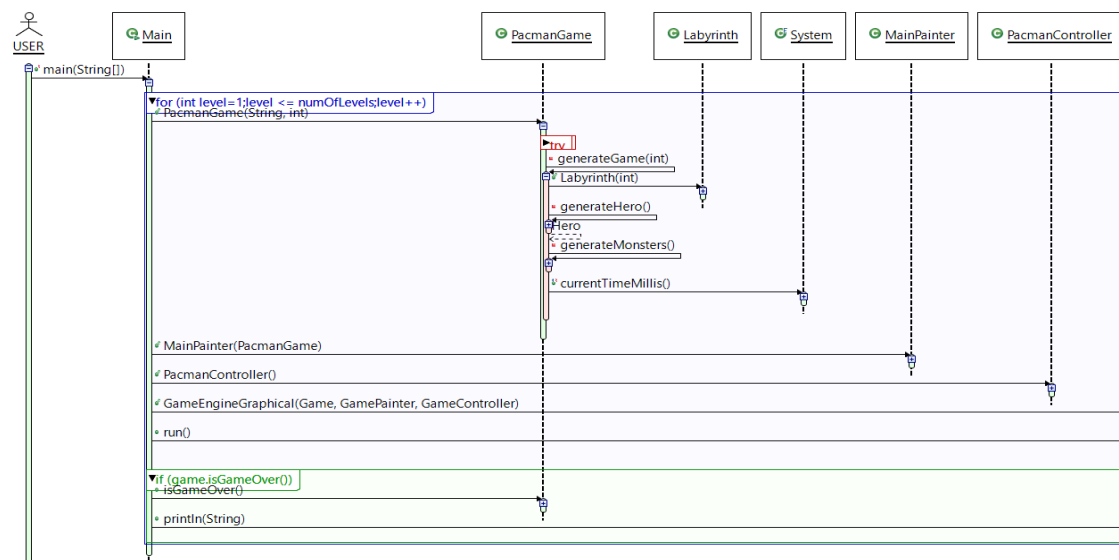


Le diagramme séquence par morceaux :

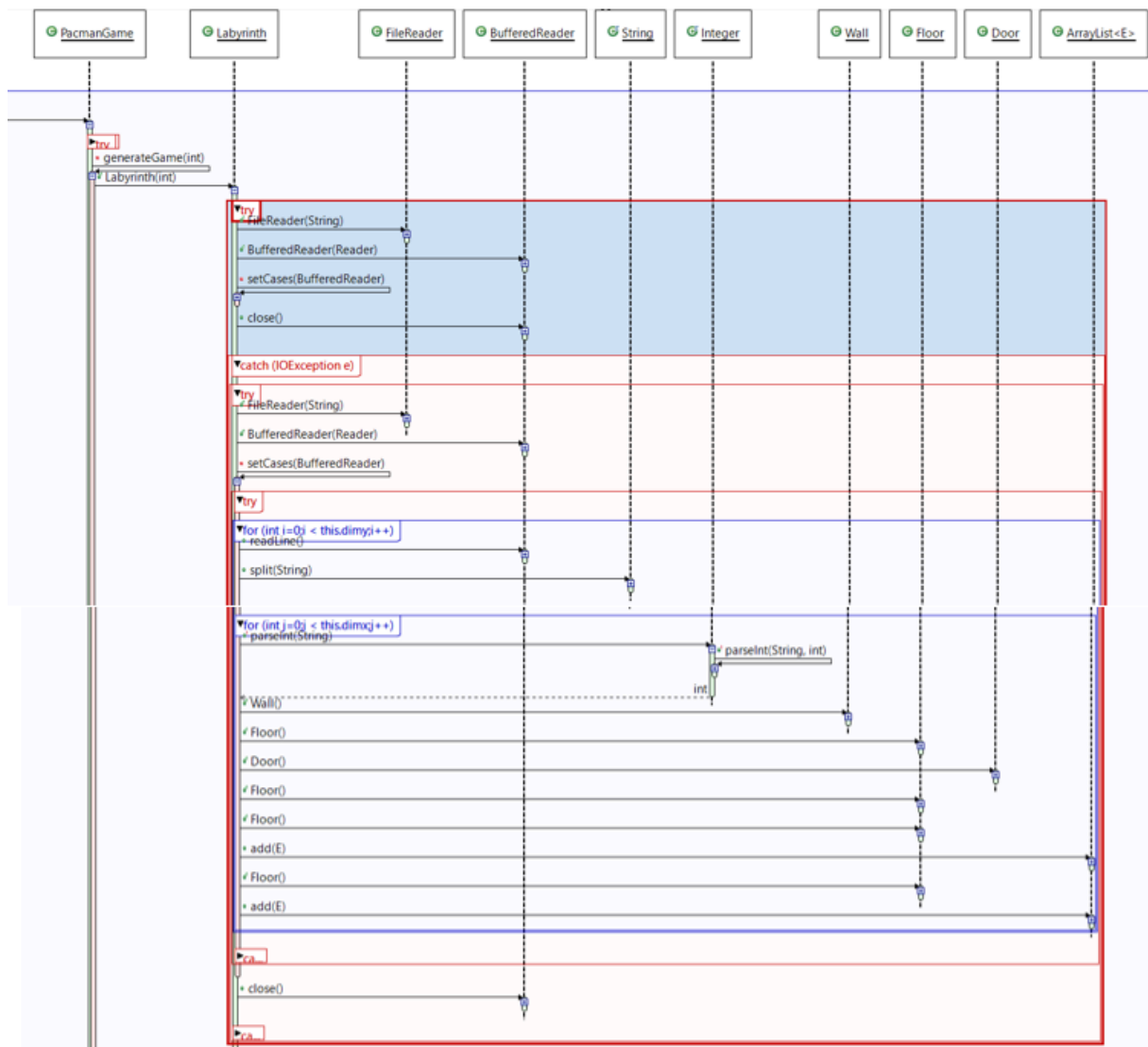
PacmanGame(String, int) :



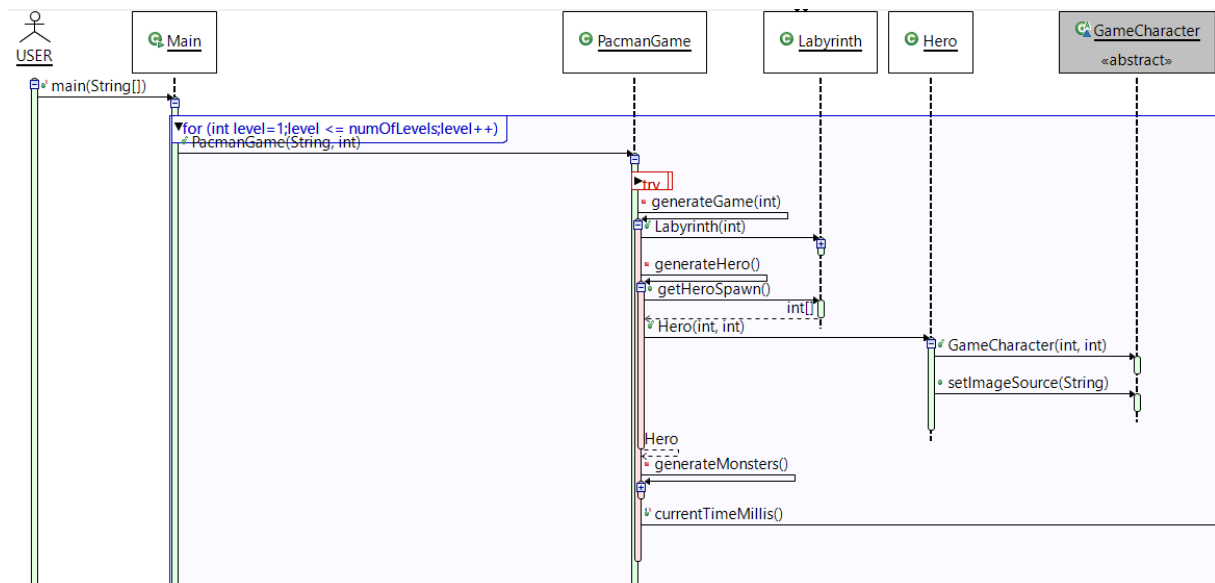
-generateGame(int) :



-Labyrinth(int) :



-Generate(Hero) :



```

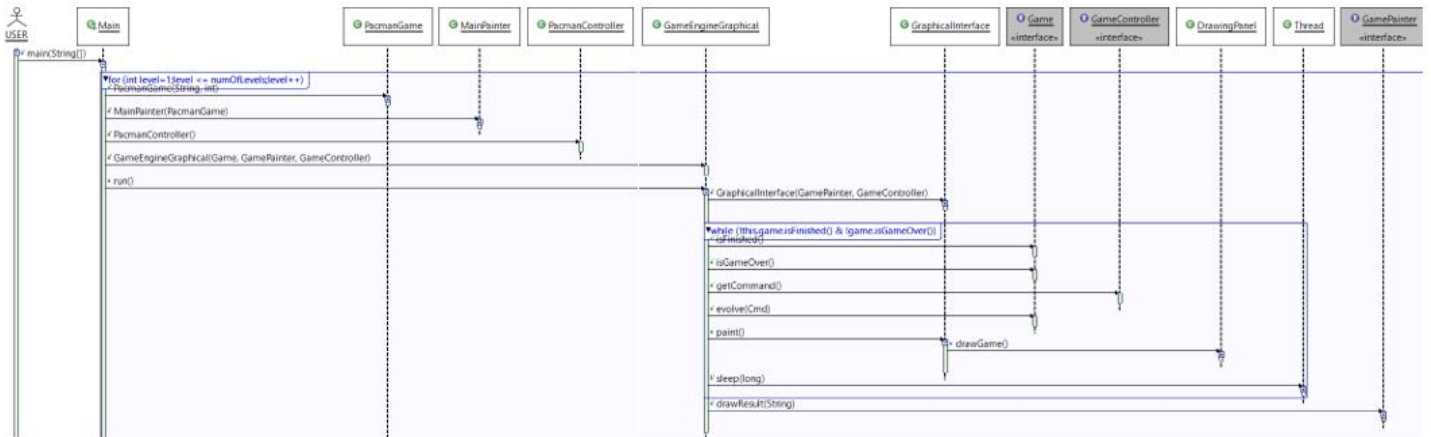
sequenceDiagram
    participant USER
    participant Main
    participant PacmanGame
    participant Labyrinth
    participant ArrayList as ArrayList<E>
    participant Monster
    participant GameCharacter as GameCharacter «abstract»
    participant MonsterSmart

    Main->>PacmanGame: generateMonsters()
    activate PacmanGame
    PacmanGame->>Labyrinth: getMonsterSpawn()
    activate Labyrinth
    Labyrinth-->>PacmanGame: ArrayList<int[]>
    deactivate Labyrinth
    PacmanGame->>ArrayList: ArrayList()
    activate ArrayList
    ArrayList-->>PacmanGame: 
    deactivate ArrayList
    PacmanGame->>PacmanGame: for (int i=0; i < monsterSpawn.size(); i++)
    activate PacmanGame
    PacmanGame->>ArrayList: get(int)
    activate ArrayList
    ArrayList-->>PacmanGame: int
    deactivate ArrayList
    PacmanGame->>PacmanGame: if (pos[2] == 1)
    activate PacmanGame
    PacmanGame->>PacmanGame: Monster(int, int)
    activate PacmanGame
    PacmanGame->>GameCharacter: GameCharacter(int, int)
    activate GameCharacter
    GameCharacter->>GameCharacter: setImageSource(String)
    deactivate GameCharacter
    PacmanGame->>ArrayList: add(E)
    activate ArrayList
    ArrayList-->>PacmanGame: 
    deactivate ArrayList
    PacmanGame->>PacmanGame: else
    activate PacmanGame
    PacmanGame->>PacmanGame: MonsterSmart(int, int)
    activate PacmanGame
    PacmanGame->>GameCharacter: GameCharacter(int, int)
    activate GameCharacter
    GameCharacter->>GameCharacter: setImageSource(String)
    deactivate GameCharacter
    PacmanGame->>MonsterSmart: Monster(int, int)
    activate MonsterSmart
    MonsterSmart->>MonsterSmart: setImageSource(String)
    deactivate MonsterSmart
    PacmanGame->>ArrayList: add(E)
    activate ArrayList
    ArrayList->>ArrayList: add(E, Object[], int)
    activate ArrayList
    ArrayList-->>PacmanGame: boolean
    deactivate ArrayList
    PacmanGame-->>Main: ArrayList<Monster>  
currentTimeMillis()
    deactivate PacmanGame
  
```

```
sequenceDiagram
    actor USER
    participant Main
    participant PacmanGame
    participant MainPainter
    participant Labyrinth
    participant PacmanController
    participant GameEngineGraphical
    participant PrintStream

    Main->>USER: * main(String[])
    activate USER
    USER->>Main: * main(String[])
    deactivate USER
    Main->>PacmanGame: for (int level=1;level <= numOfLevels;level++)
    activate PacmanGame
    PacmanGame->>MainPainter: * PacmanGame(String,int)
    activate MainPainter
    MainPainter->>PacmanGame: * MainPainter(PacmanGame)
    deactivate MainPainter
    PacmanGame->>MainPainter: * getHero()
    activate MainPainter
    MainPainter->>PacmanGame: * getMonstres()
    activate PacmanGame
    PacmanGame->>MainPainter: ArrayList<Monster>
    deactivate PacmanGame
    MainPainter->>Labyrinth: * getBuild()
    activate Labyrinth
    Labyrinth-->>MainPainter: Tile[][]
    deactivate Labyrinth
    MainPainter->>PacmanGame: * getBullets()
    activate PacmanGame
    PacmanGame->>MainPainter: ArrayList<Bullet>
    deactivate PacmanGame
    MainPainter->>PacmanController: * PacmanController()
    activate PacmanController
    PacmanController->>GameEngineGraphical: * GameEngineGraphical(Game, GamePainter, GameController)
    activate GameEngineGraphical
    GameEngineGraphical->>GameEngineGraphical: * run()
    deactivate GameEngineGraphical
    GameEngineGraphical->>PrintStream: * printn(String)
    activate PrintStream
    PrintStream-->>GameEngineGraphical: 
    deactivate PrintStream
    deactivate GameEngineGraphical
    deactivate PacmanController
    deactivate MainPainter
    deactivate PacmanGame
    deactivate Main
    deactivate USER
```

Run() :



isGameOver() :

