Projet CPS :

Lemmings

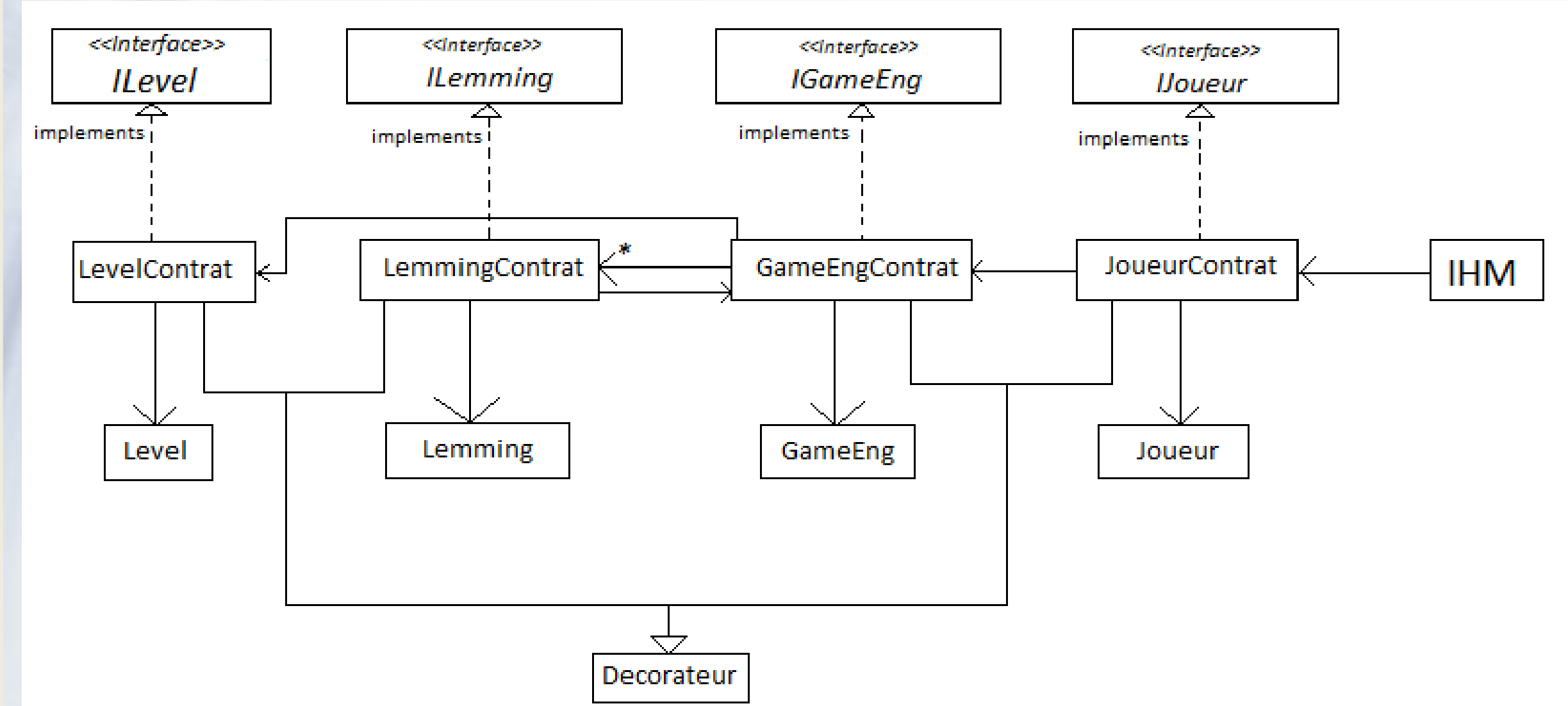
par Margarido Julien
et Lima Gorito Félix

(M1 STL, UPMC)
2016

Sommaire

- **Architecture**
- **Contrats**
- **Tests MBT et bugs**
- **Démo**
- **Bilan et conclusion**

Architecture



Contrats

Service: ILemming
Types: int, bool, double, IGameEng, EtatLemming

Observators:

```
getWidth: [ILemming]-> int
getHeight : [ILemming]-> int
getId : [ILemming]-> int
isDroitier : [ILemming]-> bool
getEtats : [ILemming]-> EtatLemming[]
getGameEng : [ILemming]-> IGameEng
getNbCasesFalling : [ILemming] -> int
getNbToursBomber : [ILemming] -> int
getNbToursBuilder : [ILemming] -> int
getNbDallePose : [ILemming] -> int
getNbCreuse : [ILemming] -> int
```

Constructors:

```
init: [GameEng] -> [ILemming]
```

Operators :

```
setHeight : [ILemming] * int -> [ILemming]
  pre setHeight(Le, height) require 1 <= height < getGameEng().getLevel().getHeight()

setWidth : [ILemming] * int ->  [ILemming]
  pre setWdith(Le, width) require 1 <= width < getGameEng().getLevel().getWidth()

setDirection: [ILemming] -> [ILemming]
setEtat : [ILemming] * EtatLemming -> [ILemming]
step : [ILemming] * int -> [ILemming]
```

Observations :

[invariants]

```
getHeight(Le) >= 0
getWidth(Lem) >= 0
getHeight(Lem) < getGameEng().getLevel().getHeight()
getWidth(Lem) < getGameEng().getLevel().getWidth()
getId(Lem) <= getGameEng().getSizeColony()
getNbCasesFalling(Lem) <= getGameEng().getLevel().getHeight()
```

[init]

```
getGameEng(init(gameEng)) == gameEng
getWidth(init(gameEng)) == gameEng.getLevel().getEntranceWidth()
getHeight(init(gameEng)) == gameEng.getLevel().getEntranceHeight()
getId(init(gameEng)) == gameEng.getSpawned()
isDroitier(init(gameEng))==true
getEtats(init(gameEng)).contains(EtatLemming.FALLER)
getNbCasesFalling(init(gameEng)) == 0
getNbToursBomber(init(gameEng)) == 0
getNbToursBuilder(init(gameEng)() == 0
getNbDallePose(init(gameEng)) == 0
getNbCreuse(init(gameEng)) == 0
```

[setHeight]

```
getHeight(setHeight(Lem, height))==height
```

[setWidth]

```
getWidth(setWidth(Lem, width))==width
```

[setDirection]

```
isDroitier(setDirection(Lem)) != isDroitier(Lem)@pre
```

[setEtat]

```
getEtat(setEtat(Lem,etat).contains(etat)
if(etat==EtatLemming.STOPPER){
      gameEng.getLevel().getNature(height, width)==Nature.STOPPER;
      gameEng.getLevel().getNature(height-1, width)==Nature.STOPPER;
 }
 if(getEtat()@pre.contains(EtatLemming.FLOATER)){
      getEtat().contains(EtatLemming.FLOATER)
 }
 if(getEtat()@pre.contains(EtatLemming.BOMBER)){
      getEtat().contains(EtatLemming.BOMBER)
 }
 if(getEtat()@pre.contains(EtatLemming.CLIMBER)){
      getEtat().contains(EtatLemming.CLIMBER)
 }
```

[step]

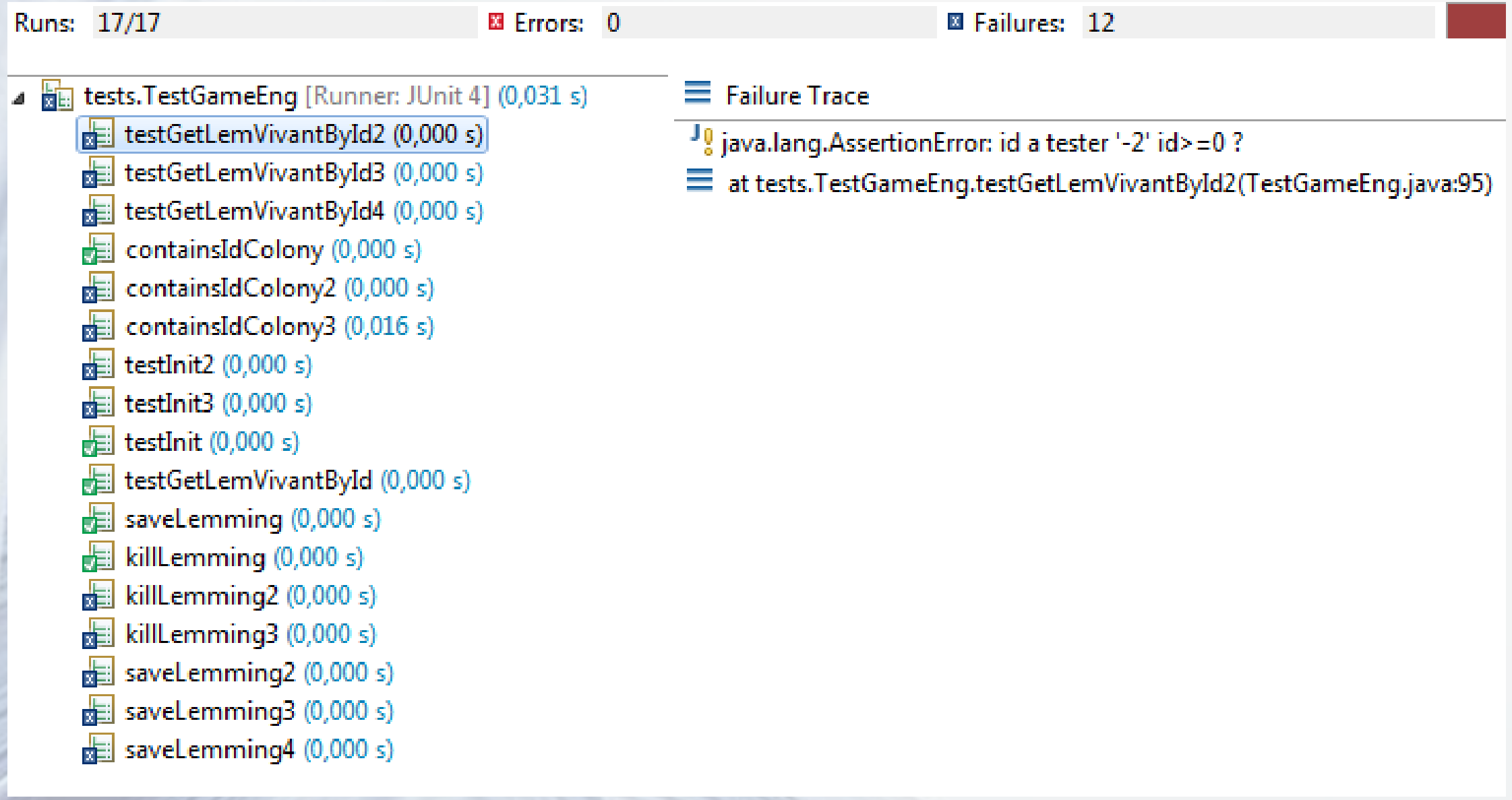
```
.... ~360 lignes ....
```

Tests MBT

```java
/**
 * Objectif de Test: getLemVivantById(int id) reussit
 *
 * Cas de Test: gameEng.getLemVivantById(int id)
 *       0 <= id <sizeColony && containsIdColony(id)
 *
 * Condition initiale:
 * Tous les lemmings ont été ajouté
 *
 * Operation:
 *  getLemVivantById(5)
 */
@Test
public void testGetLemVivantById(){
    for(int i=0; i<gameEng.getSizeColony();i++){
        //ON Ajoute manuellement tous les lemmings
        Lemming lem = new Lemming();
        lem.init(gameEng);
        gameEng.addLemming(lem);
    }
    int id=5;
    int sizeColony = gameEng.getSizeColony();
    assertTrue("id a tester '"+id+"' id>=0 ?",
            id>=0);
    try{
        assertTrue("id a tester '"+id+"', containsIdColony(id) ?"
                , gameEng.containsIdColony(id));
    }catch(PreConditionError e){
        assertFalse(e.toString(),true);
    }
    assertTrue("id a tester '"+id+"', sizeColony = "+sizeColony+ " "
            + "id<sizeColony ?",
            id < sizeColony);
}
```

```java
/**
 * Objectif de Test: getLemVivantById(int id) retourne failed
 *
 * Cas de Test: gameEng.getLemVivantById(int id)
 *       id<0
 *
 * Condition initiale:
 * Tous les lemmings ont été ajouté
 *
 * Operation:
 *  getLemVivantById(-2)
 *
 */
@Test
public void testGetLemVivantById2(){
    for(int i=0; i<gameEng.getSizeColony();i++){
        //ON Ajoute manuellement tous les lemmings
        Lemming lem = new Lemming();
        lem.init(gameEng);
        gameEng.addLemming(lem);
    }
    int id=-2;
    int sizeColony = gameEng.getSizeColony();
    assertTrue("id a tester '"+id+"' id>=0 ?",
            id>=0);
    try{
        assertTrue("id a tester '"+id+"', containsIdColony(id) ?",
                gameEng.containsIdColony(id));
    }catch(PreConditionError e){
        assertFalse(e.toString(),true);
    }
    assertTrue("id a tester '"+id+"', sizeColony = "+sizeColony+
            " id<sizeColony ?",
            id < sizeColony);
}
```

Tests MBT


Runs: 17/17
Errors: 0
Failures: 12
tests.TestGameEng [Runner: JUnit 4] (0,031 s)
testGetLemVivantById2 (0,000 s)
testGetLemVivantById3 (0,000 s)
testGetLemVivantById4 (0,000 s)
containsIdColony (0,000 s)
containsIdColony2 (0,000 s)
containsIdColony3 (0,016 s)
testInit2 (0,000 s)
testInit3 (0,000 s)
testInit (0,000 s)
testGetLemVivantById (0,000 s)
saveLemming (0,000 s)
killLemming (0,000 s)
killLemming2 (0,000 s)
killLemming3 (0,000 s)
saveLemming2 (0,000 s)
saveLemming3 (0,000 s)
saveLemming4 (0,000 s)
Failure Trace
java.lang.AssertionError: id a tester '-2' id>=0 ?
at tests.TestGameEng.testGetLemVivantById2(TestGameEng.java:95)

Implémentations avec bugs

```
for(int i=0; i<h; i++){
    for(int j=0; j<w; j++){
        if(i==0 || i==(h-1) || j==0 || j==(w-1)) plateau[i][j]=Nature.DIRT; //BUG
        else plateau[i][j]=Nature.EMPTY;
    }
}
```

```
for(int i=0; i < getHeight(); i++){
    if(! (getNature(i, 0)==Nature.METAL)) throw new PreConditionError("getNature("+i+", 0) != Nature.METAL)");
    if(! (getNature(i, getWidth()-1)==Nature.METAL)) throw new PreConditionError("getNature("+i+", "+(getWidth()-1)+") != Nature.METAL)");
}
```

```
thread "JavaFX Application Thread" errors.PreConditionError: PreConditionError (goPlay) : getNature(0, 0) != Nature.METAL)
ontrat.bug.LevelContratBug.goPlay(LevelContratBug.java:125)
ain.LemmingMainGUIBug$2.handle(LemmingMainGUIBug.java:151)
ain.LemmingMainGUIBug$2.handle(LemmingMainGUIBug.java:1)
om.sun.javafx.event.CompositeEventHandler.dispatchBubblingEvent(CompositeEventHandler.java:86)
```

Implémentations avec bugs

```java
@Override
public void killLemming(int idLem) {
    allLem[idLem] = null;
    //nbLemVivants--; //BUG
}
```

```java
if(delegate.getNbLemMorts() < 0) throw new InvariantError("getNbLemMorts() < 0");
```

```
Exception in thread "JavaFX Application Thread" errors.InvariantError: InvariantError (getNbTours) : getNbLemMorts() < 0
        at contrat.bug.GameEngContratBug.checkInvariants(GameEngContratBug.java:265)
        at contrat.bug.GameEngContratBug.getNbTours(GameEngContratBug.java:73)
        at contrat.bug.GameEngContratBug.step(GameEngContratBug.java:211)
        at main.LemmingMainGUIBug$8$1.run(LemmingMainGUIBug.java:317)
```

Démo

Les objectifs

côté « projet »

- Créer un jeu « robuste » et fiable
- IHM soignée
- Répondre à la problématique

Les objectifs

côté « projet »

- Créer un jeu « robuste » et fiable
- IHM soignée
- Répondre à la problématique

côté « étudiant »

- Nouvelle façon de concevoir
- Travailler en binôme
- Soutenance
- Projet long à réaliser, en parallèle d'autres projets

Conclusion