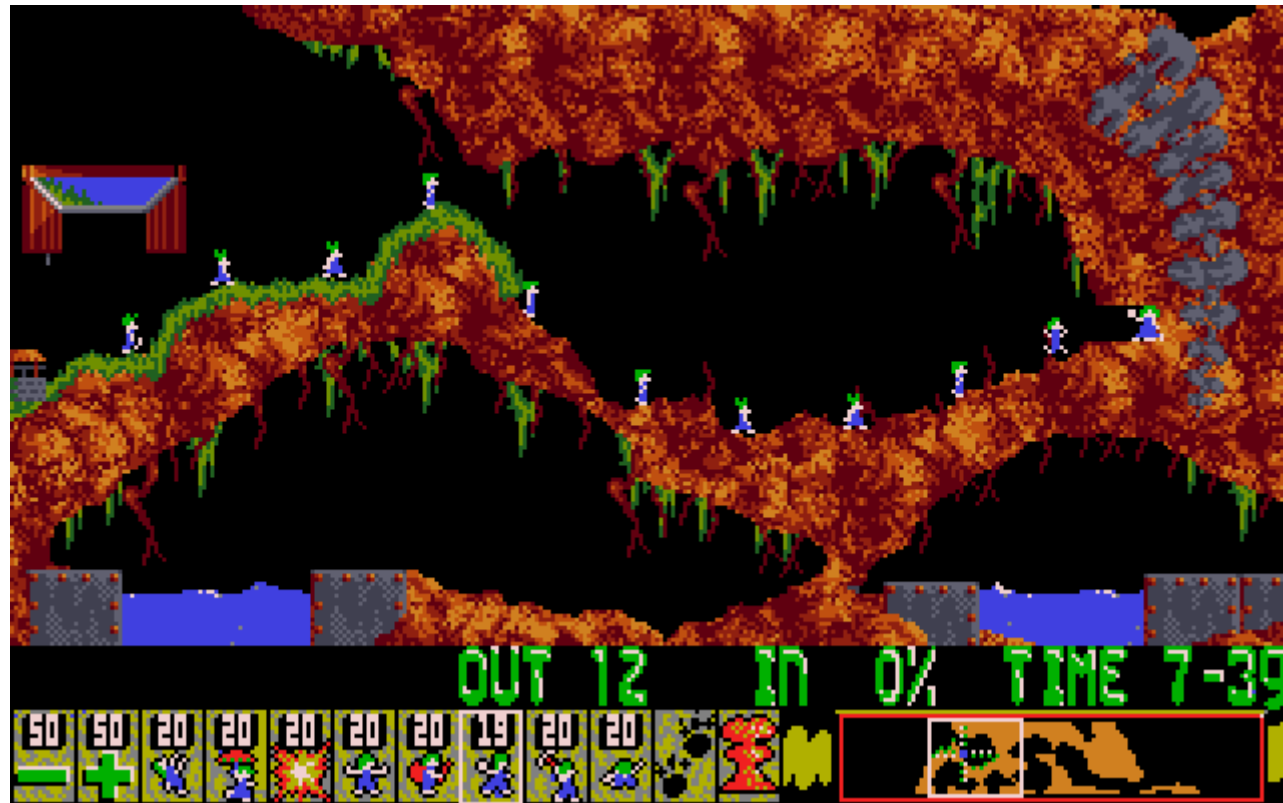


Lemmings



Sommaire

- Spécifications
- MBT
- Extensions

Spécifications

- Services :
 - Gestion du jeu : GameEngine
 - Gestion de l'environnement : Level
 - Gestion des interactions : Joueur
 - Gestion des lemmings : Lemming

Model Based Testing

Test des initialisations des services

Test du setNature du service Level

Test du goPlay du service Level

Test du step du service gameEngine

Test du spendToken du service Joueur

MBT : Les Tests

- Implémentation fonctionnelle :

```
Tests run: 14, Failures: 0, Errors: 0, Skipped: 0
```

- Implémentation non fonctionnelle :
 - On peut repérer l'origine des erreurs

```
Tests run: 14, Failures: 1, Errors: 12, Skipped: 0
```

```
Testcase: initAutoTest took 0,003 sec
```

```
    Caused an ERROR
```

```
init GameEng, tour not starting at 1
```

```
error.PostConditionError: init GameEng, tour not starting at 1
```

```
    at contrat.GameEngineContrat.init(GameEngineContrat.java:59)
```

```
    at
```

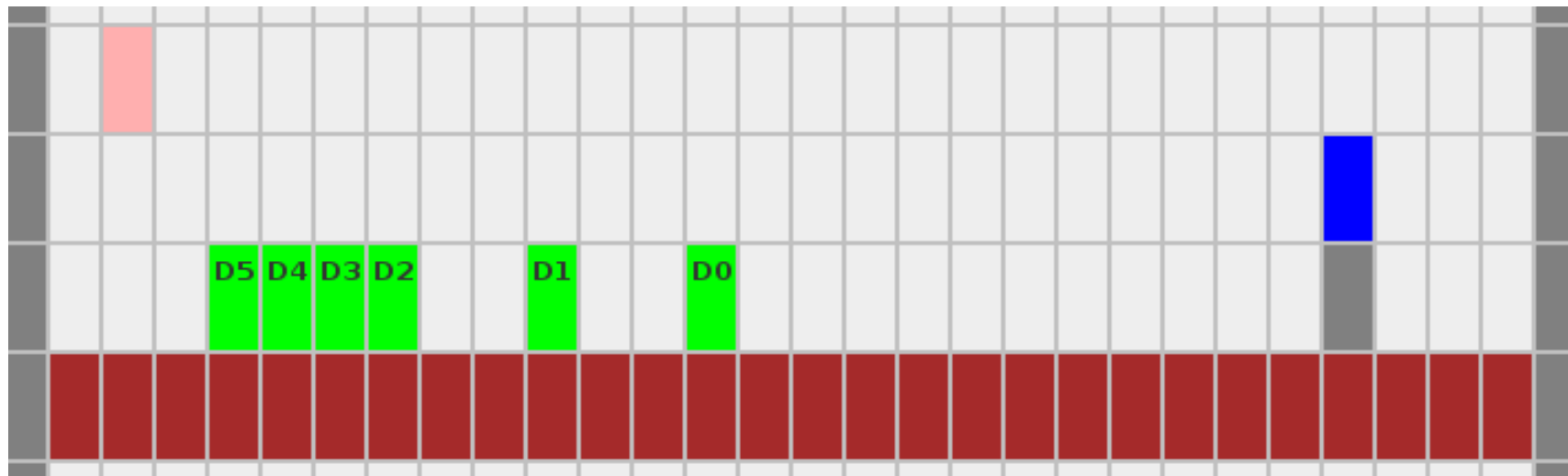
```
test.AbstractLemmingsTest.initAutoTest(AbstractLemmingsTest.java:82)
```

Extensions

- Vitesse de Création
- Annihilation

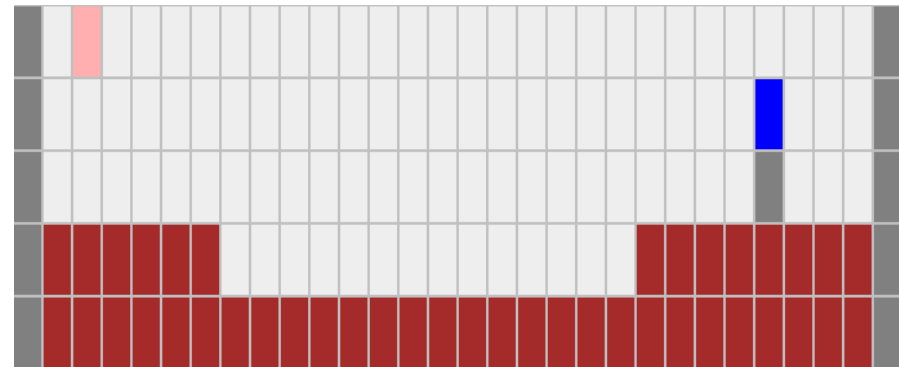
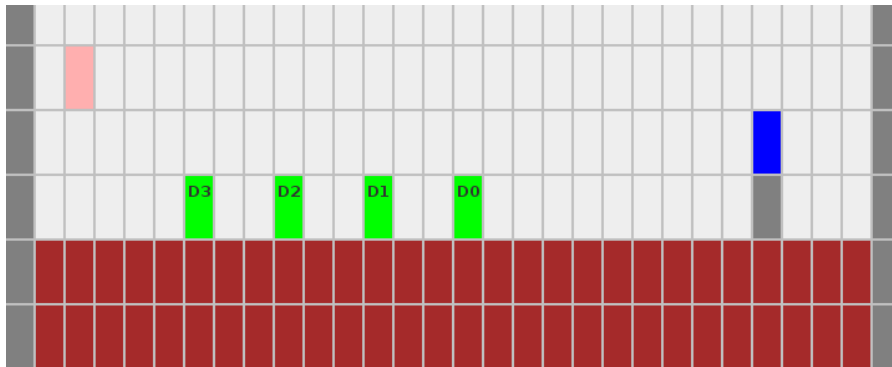
Vitesse de Création

- `setSpawnSpeed: [gameEng] * int → [gameEng]`
 - `pre setSpawnSpeed(i)` require $i > 0$
- `[setSpawnSpeed]`
 - `spawnSpeed(setSpawnSpeed(G, i)) = i`
 - `sizeColony(setSpawnSpeed(G, i)) = sizeColony(G)`
 - `spawned(setSpawnSpeed(G, i)) = spawned(G)`
 - `tours(setSpawnSpeed(G, i)) = tours(G)`
 - `nbSauves(setSpawnSpeed(G, i)) = nbSauves(G)`



Annihilaton

- annihilate: [gameEng] \rightarrow [gameEng]
- [annihilate]
 - $\forall i \ 0 \leq i < \text{sizeColony}(G)$:
 - $\text{isBomber}(\text{getLemming}(\text{annihilate}(G)), i) = \text{true}$
 - $\text{spawned}(G) = \text{sizeColony}(G)$



Conclusion

- Avantages :
 - Simplicité de réalisation
 - Simplicité de debugging
- Inconvénients :
 - Démarrage de projet lent
 - Nécessité d'être précis et quasi-exhaustif
 - Modification de fonctionnement lourde