Projet CPS
- Dungeon Master -

Naim Choullit
Charaf Lachachi

Plan

- Spécification
- Test
- Implémenation

Spécification

- Ajout du service Ressources (trésor),
- Service Key refine Ressources ,
- Ajout de OpenDoor et CloseDoor dans Player,
- Ajout d’attaque dans le service Mob
- Ajout de pourchasser dans le service Cow,

Service Ressources

Service: Ressources
Observators : …

Constructors: init : …

Observations
Invariants: Environnement::CellNature(getEnv(R),Col(R), Row(R)) in {EMP*;DNO; DWO*}
Environemment::CellNature(getEnv(K), 0,0) in {IN}
and Environnement::isReachable(Envi(R),0,0, Col(R), Row(R))

[init]: Envi(init(e)) = e
Environnement::CellNature(Envi(R), Col(R), Row(R) in *{EMP; DNO; DWO}*

Service Player (1)

Operators :

openDoor:[Player] -> [Player]
pre OpenDoor(P) **require** Key(P) = true **and**
Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) *in {DWC}*
and Environement::CellContent(Envi(P), Col(p), Row(p) - 1) = NO

Observations :

Key(OpenDoor(P)) = true
Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) *in {DWO}*

Service Mob

Operators :

Attack: [Mob] ->[Mob] // définie dans Mob

Observations: // definie dans Entity

[attack]: Face(E) = N and Environment::CellContent(Envi(E),Col(E),Row(E)-1) != No implies HP(Attack(Environment::CellContent(Envi(E),Col(E),Row(E)-1))) = HP(Environment::CellContent(Envi(E),Col(E),Row(E)-1)) - 1

ForAll cell in Environment::getCells(Envi(E)), e <- Environment::CellContent(Cell::row(cell),Cell::col(cell)) and e != NO and Mob::row (e) != row and Mob::Col(e) != col -> HP(Mob::attack(e)) = HP(e)

Service COW

Operators :

chase[Cow] ->[Cow] // définie dans Mob

Observations: // definie dans Entity
[chase]:

ForAll cell in Environment::getCells(Envi(E)), p <-Environment::CellContent(Cell::row(cell),Cell::col(cell)) and p is Player and distance(Player:row (p) , Player::Col(p); Row(C), Col(C)) < 4 ->

Face(C) = N -> Environnement::CellContent(Row(C)-1,Col(C)) in {EMP,CWO} ->Forward(C)

Player::Row (p) < Row(C) and Player::Col(P) < Col(C) -> Cow::moveLeft(C) …

Player::Row (p) > Row(C) and Player::Col(P) < Col(C) -> Cow::turnLeft(C) ,Cow::turnLeft(C) Cow::forward(C) …

TEST

```
@Test
public void attackTestPost1Pos(){
1       EngineService labyrinthe = new EngineContract(new Engine());
2       CowService cow = new CowContract(new Cow());
3       EnvironnementService env = new EnvironnementContract(new Environement());
4       env.init(15, 15);
5       labyrinthe.init(env);
6       player.init(env, 0, 0, Dir.E);
7       labyrinthe.addEntity(player);
8       int row = player.getRow();
9       int col = player.getCol()  + 1;
10      env.setNature(row, col,Cell.EMP);
11      cow.init(env, row, col, Dir.W, 4);
12      labyrinthe.addEntity(cow);
13      int hp = cow.getHp();
14      player.attack();
15      assertTrue(cow.getHp() == hp - 1);
}
```

Implémentation

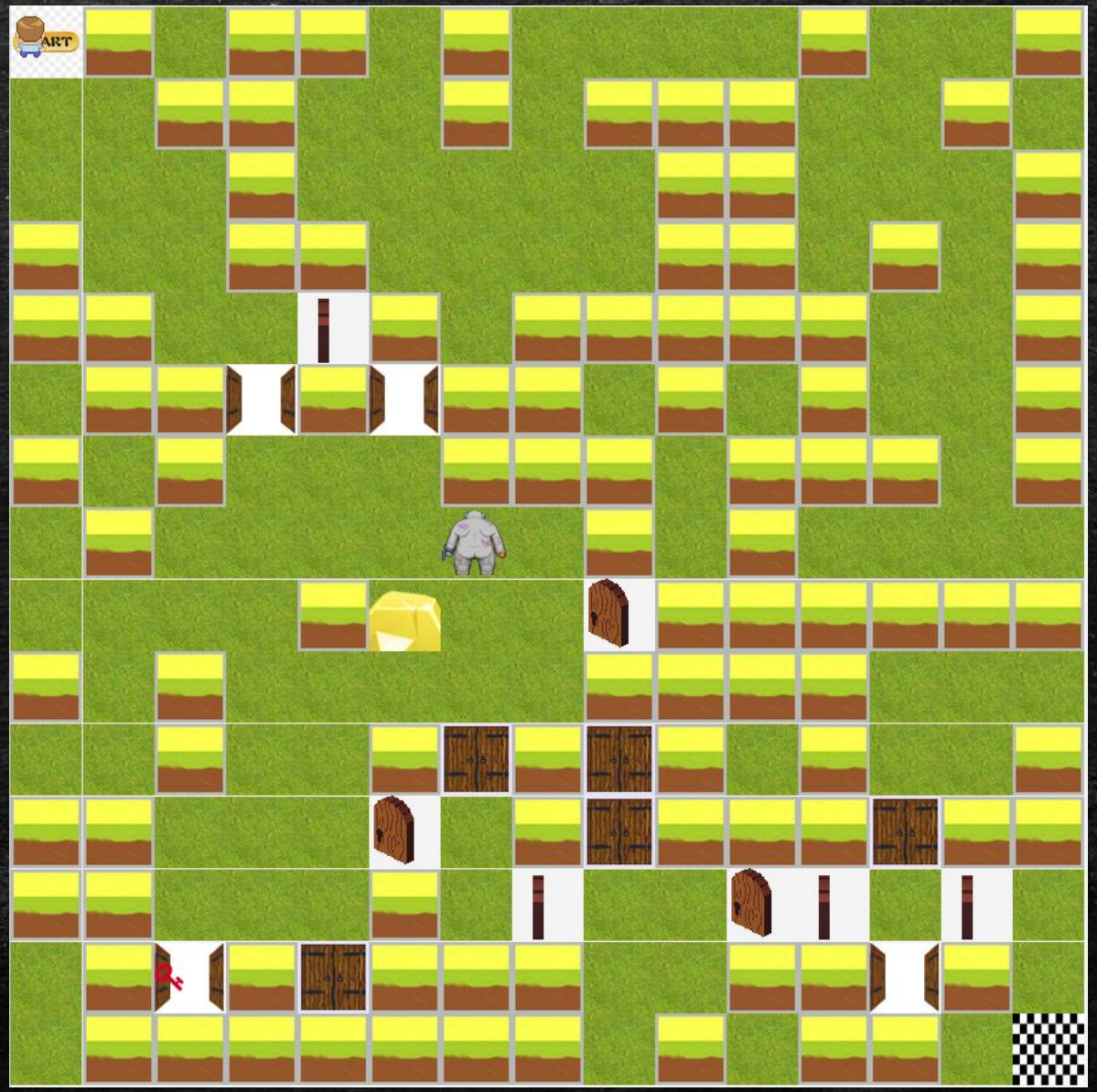
Fonctionnalités

- Jeu
- Monstres et combat
- Trésor
- Affichage et interface 2D
- Edition des grilles
- Clef

Services

- Cow
- EditMap
- Environnement
- Key
- Ressources
- Player

IHM principale



IHM EditMap