http://www.plantuml.com/plantuml/png/lLbVRziu4d\_tfo2I5nNh13tSmm954QoJEB419p5iTzFSebZGCcC9bKI1f9qrSldkfw56f8QoVTVRe2-E-9lXS3X\_XkJaDssuCfiywvKy\_ixNWftydegN4\_RQOt76jMOmV9GgMm74Vc7pKgL6N78J9qmlBVIWJ2BNYfT96b\_nN2WMmoy7efyVVR1Z9OgLKD5p9hcfaLhwMXZ7zjPUVB4nibdRW8OQVly\_NL3SGLMsCbqndSYNZXPtejYybnOexWUrk4apSPL8gOL2HM8bk14JiHLPb1Q6PV1bfxprl3zpkHAPTQY3ueGh7XkXBCpuKXiOuq9N3TKIsIa3sKbPVo-bp0Gl63UcakRffTHVoUZ9ZrQf4h59PK55c4J9pJfvv5a6XAuXZrko6rnTO-uBhx-dvIbgoRJzuo2d-19-0KO\_hXH\_YUZEFhJ8kToAMRfEJ1IY4\_7S1HUox62NqXYP4zXiL16HJRMfAm6HeyMeh7bupjUYpQ25-Mf9VmNeaqLtY5RVODTfvVge0pqXv0oPweKtPS0\_AWDWOCEBx2ajKEkfLI8HF3C9C5J1pc2UKCyy5kokCaak2eFHspMpHwcPJwcPAKRDV7hCpA0FfjATB3GcKhCFp9mhcMMfhcP4TNo-mE\_-sHgdkt4qK8hl9gasdn6wO3cQgai2SxgPmGgtYTJEKA\_QS9F6RzO7y-bil1ZEn\_yQrd8jPnqJVOP1G4HDCxxpEoMohgQBmVtuRh0OFzo7PBBKu\_Xsj20StmAEKSDnjezZyJ0zIBjyM2mUxWwI9yERmt9dVbLhWJewFR3pG5WXvvLZL\_i8b\_B5rvvYao-5gdnNWx-WDOczZ-L2UPjkHH4awtKJbDqqWlomgn7IIXma9QaXuq82GdFstc-2y1uDUCg6zg8ij9R2kWVBgRVjqRHSg4gPC5J3aA-vzeRyM-iipVXofofgilyi5Sr2Pyx\_McOfXeGLxZztU9oOIk2QcLqfQEIaUb1jPykDg3K4wVjYvBNHuWsjWhEEsNMlKL4LyEwYqYPmU5Zrdw8W0XjjiUmqvX\_cfTcX-Ul1latuJS\_1O2Zx-8Ry6jFg1\_ppwhEKAvLknTSGUAhbueIekSpqtHMXtvxT7ZmTrTJdeT0cBJZ48Ttd3sreeNZfTkDExLe4q8ZeVelbNSovW7ApKGAzviRlxxX7xbCJljz\_XJk-JcDSeXwypnh7yucgi8FZ67BWxwo5QNSb2oCA4\_sG1jTIAlRw4xOAFeGCK9jAqZ9p4HJYFyC7D2\_lfQd5DiiK2BJuC8UYS4\_-KbSFVde4JPfZXzmj7Hh14--FRtzyOuOl9\_WkR8uC29hkqK7fz-AlBa-VkJl53O-DLBlz3uoqmCk15MVLm2Nr\_nQbhMUlxpsSCVV8TQEO5t0tXx6pTyatMZGtNZ-FQsCDSK2IaZ8dOkADKgAevznKEzA98wDN-JOjrWHSYdLQ-3VyckSGEiVtIzvo8w5aTCp8X1itIUywjQz49dBBZH7iBvG8r4HJyPLIcGFtJLph9IsGeGN6YOY\_JwGMXEIinmjuY4M-cH0VZnv\_wuLOyJAFFscHc45R90g5W34HFFeYGYcad-9X-z9isISfErC2OLC9VCzfiRAN6vj6RS\_LnTyAYrjuKQKSb6NanUhk4BTLlkgCeJV4dbkevbkqXylecrHfyyWxANt8fjh8ahW1xuAWvXNN\_zMrNbITaOj42QfwMKcdg3SfmT4KCxafLfUPZByBCaCV9Hr\_DHFMILCZ8b8nY1AUXmPygYyX-JjwoJp2yX8kvXVoYuJLXSIDA3Op0Nb0pR9VcwjE-Og2M0FTsfZOocmRnB8\_KM0bD-Zwomdp9QEojsjX-lPC\_LJ6u\_Fqq0lHduamEpZo9fAlN1CqG4ErKJslEJwtqR5UkejcAlrutK3fRYwHVh2cmQBGeiKU3Cy5hkRjIO-DL9Uslf6-RYmB0ozKeNottkeh2bJ\_ZDrjT1gplFpd7FB3GfSO8H17jYHCbT0Q5dEJ8zTMfK7rNUokvOkjmXt2bt8l3A\_6bRsCK9AHcpXfXFKP-Td32TBtyw4anTVhQg73Se3KQOrVSyEnNhLikAXoOwC44jZdpooQzqzlRdxzzUF73snUPMKcZ2pOnGLQUSHrSiVBptM1xEFy2vR2ELkRbig4Q1WNsl0YhXrz5Z2WzqqDMdQG5TKhO0JrcAX2hLjvNtkahG5DCG9W9unWqFwY2FQyYAGlitkKHa4laUA1L4HHCcmXpEm3MnWDkyeyZgyNem0QJ2OKkPqzVBc\_fjZDU3QaoFLmiXXGQ3gOpmUt0T\_Tu7PyjGSQ3GUJK9VfPF0qd0NQZ1xceJrc3w7EzvT\_tez\_7mw-V0sCCFsp5efH71QiixEJVvmqvUOJIuII-tW-zi\_F-yK9DfE1Cs-koYxjKLRz\_QcknBtcdsT8j6NhaxtYzjm\_oBgazd-jkjIcuY39AL3HjysttmGxF\_\_tXUllzuASiX8eLecH6zLaMW-tRCMuSmXHKA52SIqFwnh7KQzkc7UXG2Pw8H1f6ufaCa9qihqquboHPRscbUKHwFUG8JPdpko\_E1nkMocUhUvx-77JXF18-2Od9BGHZg-bR0\_zUNi6\_r2fmkpyl3YnjCe9SVK7NDt3wEf6ggFWn6vqahdqq0UsZ2BxkGaQv0QeIH7azb7h00Q8Us422RcQHEZz9ehL9i\_-0m00

@startuml

package "View" {

class ViewWorld {

+ SpriteBatch sb

+ OrthographicCamer camera

+ ViewWorld()

+ render(float)

+ World getWorld()

}

class ViewTuto {

+ Stage stage

+ SpriteBatch sb

+ OrthographicCamer camera

+ ViewTuto()

+ render(float)

+ void show()

}

class ViewMenu {

+ Stage stage

+ SpriteBatch sb

+ OrthographicCamer camera

+ ViewMenu()

+ render(float)

+ void show()

}

class FileChooser {

+ createLevel(int level)

}

}

package "model" {

package "character" {

abstract class Character {

# int hp

# boolean attack

# int posX

# int posY

# int direction

# boolean throughWall

+ SpriteBatch sb

+ OrthographicCamer camera

+ ViewWorld()

+ render(float)

+ World getWorld()

# int posY

# int direction

# boolean throughWall

# Sprite sprite

# SpriteBatch batch

+ void draw(SpriteBatch sb)

+ void moveRight()

+ void moveLeft()

+ void moveTop()

+ void moveBottom()

+ void turn(int direction)

+ void decreaseHp(int damage)

+ void setX(int x)

+ void setY(int y)

+ int getPosX()

+ int getPosY()

+ boolean isThroughWall()

+ int getHp()

+ int getAttack()

+ void increaseHP(int health)

}

interface Movement {

# void moveRight()

# void moveLeft()

# void moveTop()

# void moveBottom()

# void turn(int direction)

}

class Monster {

+ void attackCollision(World world)

}

class Monsters {

+ Monsters(ArrayList<Monster> m)

+ ArrayList<Monster> getMonsters()

}

class Ghost {

# {static} int SPRITESIZE

+ Ghost(int x, int y)

}

class Player {

+ {static} int CPTANIMATION

+ {static} int SPRITESIZEWIGHT

+ {static] int SPRITESIZEHIGHT

- {static] int SPRITETOP

- {static] int SPRITEBOTTOM

- {static] int SPRITELEFT

- {static] int SPRITERIGHT

- int xAttack

- int yAttack

# boolean attackSword

# boolean attackBow

# int numberArrow

+ Player(int x, int y)

+ void increaseHP(int hpGiven)

+ void setDirection()

+ void incrementArrow()

+ void decrementArrow()

+ void hit()

+ void nohit()

+ int getNumberArrow()

+ void attackCollision(World world)

+ void attackSword(World world)

+ void drawhit(SpriteBatch sb)

+ void attackBow(int direction)

}

class Troll {

+ {static} SPRITESIZE

+ Troll(int x, int y)

}

}

package "labyrinth" {

abstract class Cell {

- boolean solid

# int x

# int y

# String type

# boolean isActivate

+ Cell(int x, int y)

+ {abstract} void activate(World world)

+ void draw(SpriteBatch sb)

+ String getType()

+ int getX()

+ int getY()

+ void setActivate()

}

class Empty {

+ Empty(int x, int y)

+ boolean isSolid()

+ void activate(World world)

}

class Passage {

- int arriveX

- int arriveY

+ Passage(int posX, int posY, int arriveX, int arriveY)

+ boolean isSolid()

+ void setDestination(int x, int y)

+ void activate(World world)

}

class Trap {

- int damage

+ Trap(int x, int y, int damage)

+ boolean isSolid()

+ void activate(World world)

}

class Treasure {

+ Treasure(int x, int y)

+ boolean isSolid()

+ void activate(World world)

}

class Wall {

+ Wall(int x, int y)

+ boolean isSolid()

+ void activate(World world)

}

class Magic {

+ Magic(int x, int y)

+ boolean isSolid()

+ void activate(World world)

+ void draw(SpriteBatch sb)

}

class Mystery {

+ boolean isSolid()

+ void activate(World world)

+ int getContent()

+ void draw(SpriteBatch sb)

}

class Door {

+ boolean isSolid()

+ void activate(World world)

+ void draw(SpriteBatch sb)

}

class WallDestructible {

+ WallDestructible(int x, int y)

+ boolean isSolid()

+ void activate(World world)

+ void draw(SpriteBatch sb)

+ void setNotSolid(boolean notSolid)

+ void damage()

}

class Water {

+ Water(int x, int y)

+ boolean isSolid()

+ void activate(World world)

+ void draw(SpriteBatch sb)

}

class Labyrinth {

+ void Labyrinth(char[][] tabLevel)

+ Cell getCell(int x, int y)

+ Cell getNextCell(int x, int y, DirectionFactory direction)

+ void init()

Cell[][] getCells()

+ void draw(SpriteBatch sb)

}

}

class World {

- int level

- int roundBlocked

- boolean pauseMonsters

- int lastLevel

- int level

- int room

- boolean currentLevelFinish

- boolean isSaving

- boolean begin

+ void game()

+ void draw(SpriteBatch sb)

+ void moveHero(DirectionFactory direction)

+ void moveMonster(int direction, int elem)

+ damagePlayer(int damage)

+ teleportPlayer(int x, int y)

+ void winPlayer()

+ void checkLosePlayer()

+ boolean canMove(Character c, DirectionFactory direction)

+ Player getPlayer()

+ void turn(int direction)

+ void loose()

+ Player getHero()

+ List<Monster> getMonsters()

+ void createLevel()

+ void healPlayer(int hp)

+ void createMonsters()

+ Point findEmptyCell()

+ void mystery(int c)

+ void nextDoor()

+ int getDoor()

+ void load()

+ void load(File f)

+ void save()

+ Cell getFirstWall(int x, int y, int direction)

+ void stopPlayer()

+ String nextCase(Character c, DirectionFactory d)

+ void pauseThread()

+ void playThread()

+ void createThread()

+ int getRoundBlocked()

+ void setRoundBlocked()

+ void setMonsters(ArrayList<Monster> m)

+ void setLastLevel(int lastLevel)

+ int getLastLevel()

+ boolean isBegin()

+ void setRoom(int room)

+ void setLevel(int level)

+ boolean isSaving()

+ void menuReturn()

+ int getLevel()

+ boolean isCurrentLevelFinish()

+ void nextRoom()

+ void evolveMonsters()

+ Labyrinth getLabyrinth()

+ void createRoom()

+ void winArrow()

}

interface Attacker {

# void attackCollision(Wold world)

}

class LabyrinthLoader {

+ LabyrinthLoader()

+ Labyrinth createLabyrinth(int numLevel)

}

class CharacterLoader {

+ List<Monster> getMonsters()

+ createCharacter(int numLevel)

+ Player getPlayer()

}

}

class Game {

+ void create()

+ void dispose()

}

package "controller" {

class Listener {

- Music mp3Sound

- Boolean turnFirstPress

+ Listener(ViewWorld)

+ boolean keyDown(int)

+ boolean keyUp(int)

+ boolean keyTyped(char)

+ boolean touchDown(int, int, int, int)

+ boolean touchUp(int, int, int, int)

+ touchDragged(int, int, int)

+ boolean(int)

}

}

package "DataFactory" {

class TextureFactory << (S,#FF7700) Singleton >> {

- HashMap<String,Texture> images

+ TextureFactory getInstance()

- TextureFactory()

+ texture getImage(String)

}

enum DirectionFactory {

+ {static} TOP

+ {static} BOTTOM

+ {static} LEFT

+ {static} RIGHT

+ {static} TURNTOP

+ {static} TURNBOTTOM

+ {static} TURNLEFT

+ {static} TURNRIGHT

}

class LabyrinthFactory {

+ {static} HEIGHT

+ {static} WIDTH

+ {static} WALL

+ {static} GROUND

+ {static} FIRE

+ {static} DELTA

+ {static} PASSAGE

+ {static} MAGIC

+ {static} MAGICHEALTH

+ {static} PLAYER

+ {static} GHOST

+ {static} TROLL

+ {static} NB\_NIVEAUX

+ {static} HP\_PLAYER

}

}

CharacterLoader --"1" Player : hero

CharacterLoader --"0,..,n" Monster : monsters

ViewWorld --"1" World : world

ViewTuto --"1" Game : game

ViewMenu --"1" Game : game

FileChooser --"1" Game : game

Listener --"1" ViewWorld : viewWorld

Character ..|> Movement

TextureFactory -"1" TextureFactory : ourInstance

Monster ..|> Attacker

Monster --|> Character

Monsters --|> Monster

Ghost --|> Monster

Ghost ..|> Attacker

Player --|> Character

Troll --|> Monster

Troll ..|> Attacker

Empty --|> Cell

Passage --|> Cell

Trap --|> Cell

Treasure -Right-|> Cell

Wall --|> Cell

Magic --|> Cell

Mystery --|> Cell

WallDestructible --|> Cell

Water --|> Cell

Door --|> Cell

Labyrinth -Left-"0..n" Cell : cellList

World --"1" Player : hero

World --"0..n" Monster : monsters

World --"1" LabyrinthLoader : labyrinthLoader

World --"1" Labyrinth

World --"1" CharacterLoader : characterLoader

World --"1" Game

World -- "1" Monsters

Game --"1" ViewMenu

@enduml