http://www.plantuml.com/plantuml/png/lLbHR-Cu3dxFhn3Q5mysBUPm3miC1iMcQTe4IDiWyModjnWiL4UDZN4iGrBI3NhzxsVAj6JAIUxcjfYNrFf8aHH5KXBxcpPScSqwxvKy-SvNWftydecN4\_RQOqdEjMOmV90gNmB4Vc6BKcL6N7AJf4m\_MUXUcLIk52\_JB1dojL0iWLyEH3y\_-c37IXHBeQBdN79J8xNqbJ2ExQsjFDuOsUWsOA66t\_\_F5XHtq9Ijp9PCf\_AbOyMjA3RlPGMA-q4hhhDS35CfjL1eIA84Ds8gjYAFiiAm7BxibBUUty-rN8hSRgY3afGhdXYXBCpuapOmHaN3Xcg9x9I1xBIilv-ap0Kl63UcakRffTHVoUZHZvQP4ed9P455c5J9pIfzu7aEXAuZZtko6rnTP-uBhxzdvIbQoRJzuo1d-1Fy0epxk5Jy9Q8h-z0Yh-LMpBDLQg8GdOhdBXZBieDTIcFaci1ceug8BAfDNGg872t6PSt3rtmbsWnQcA-Mz5U0FbfqXsZr3NwTLLiVTQ17X9mZCnrxLmRyup801ZQyo9go0gsUMIDImNEJ0aCLx0pc2VNC4y5kAvUiHM4mUhjkzYXryobrCyMecq-FkHdimLIwbON6H6hMWPapb7cUwMf6LCVd2\_pkdwrnkXj70wNuRfffyncX2xP6LtL9u4utCz1maqhj7FMg3JTPycRtO36RJ-BHOlAlaPLRSzOnqMSO14JKBESxjr8gQpYB1tUJsq4ykRyBoKJLm-Hc71EERm77kE4usySHtyyEqYxlu\_Z-zY1vEhe-B7VkjLeFrD7fWPq7mWgvg3PskOzmALzyxIaswoUXghshmL\_GcyJVntAXlCcseWYIzQe9ocuQGNvOROIq50T9QMR8k920q9ozyuiWl4S3dhAXloWBhQMWzt0vTRuzcfQngem9GpKC-PfhRyY\_jSxId3\_jL4LDzv-beb5qvlPVopp3aB32\_USEZnDJ2Lmnioi5ZPnC3whbRBaHjOKWVL-Cl3PMlA5NSDOnl-vrAfe2knjNrWGR7bRznoY8mCPQB3kD-qVhqkpG\_VLWtoByeXVWC9HzV45UnwpwWVrvzLdAbSgsuci8FDPoSKBKN6RwxehGRyzk3nwFM-hoK6YJ5HpYaAppnnOKArwwrRXJknO1D2Awts9v5tV6K6usIk2kkV7xRzm3zwa9t--luPQliWHLr8FtqN4ydwW9EpYE8GV-ZYvCkw4iZ2XCz4CMN4cfsEjFM2hi8MI0sbIIdd8NGI7-C\_Q0vkMTDBNOHas1015ZSoWAr-GlTVNWfqVGj3bsoDtIeH4yyVxuziStPlZJ5D-5pP41GTCzEYZzJlpLvUapTwUuvecHQhV\_WP4LU3cmugmPg5B\_Rr7QUlRwtiCPSuzSDqfu0NTp63j\_btoZHNFZzVEuDjOH1oGfATT4JB9HIXJrdEjgHJf5ijjLlisA5G6Vn2eh\_1j-nTSGEiVNIzvoOw5aTCJ9b1iNIUywzLw9NAmjDqQmlr0YK1DDnLTAPGxSDr7NIrOWGmjCKf5ydqej2CbvZnVm48jyCo4v7Zt-rWknucKUVz8YCOUsI1GA0MSYUFH5X598FyN3zgNPyazQTgO4mcOI-9wpOcalDpQDsZjN5tyhB6dXHPLoK9QH5wkxGznI-R8pXjuGUswXcc\_H7owZhpEbpGFlfFGXdseZIx8DU1S4CuTS\_zUjzQBgZ8nJ9QZfPINTezIbY1DNpEMcM5xcClako0nzb7JqrKnPfqsDY4Z584Pu7XhmcRw4vE\_O9TSHbfTGcLVabOJLXSIDA3Pp0Nb0tRBVcc6dV4L1h85jRKniPRuDOzcTAA393RhxvOMvab7vsxKmVNkcVYhZyNbww8NepaIO7HvvKycNhWaQeA7PQ9wt79\_RkB5UkejcAlrutK3fRYwLVgYcWL9eqM8FXgy5QlF-fCT6fajRtqXVDv65WHUgKBvRR-qL1Pf\_ncutEalOklpd0lB3GfSO8H17jYJCbD0QbBd9aMkhqg3wBdPNyiLMuGxXIxaNXbVZqbv6ACd8JP8smlgC\_EpX1Edx-L2IugjLfUYm721rMkDNt72iLoqVnbLkR9H00llycKMB\_kdrzQ-\_VlpuWIsghCo5aGMxk40lZxbERtdvkIwGVPn\_mJ8uPslNKfaG3PD26ruazKQV1GmHhgq6BJl8YceD644z9gfGwrRUrnvfQq1JZ03O2ICOjByeWZql8kdB\_0wbKT1BfBYJIb2K34i8C\_l04iQZhZ4Fawju7423wPGYD\_FxBtTN5BkUp4SKkHfDum65PeF5Od0JyDqERYR3FT1uD9Y6jiocWyVHFB1cVBy8\_J6\_3ss-k\_ppRlBxQF3bQ-24sPzEA5bqrPySLikCCRh3GdPsTlAFawOCVMAfK78VpyV--Nc\_EC4cCt2kcokqAmSeg\_x-L5VeNlDFDIJQSlR9NdrxxXzdNLBxlrbTQbE9aEGCgEZRvjilWfsV\_\_l2zVrxGQvP2HIhnCYDQZAmXqksOjpvX2YOKA6eoyEwnd7KgnlfNIYGYRiGYBIDHZ8P8JhPNYPnhiZpNjFYyWZqWSWGcpPdzbyV3hSj5izMzqFyk6bEU2HygnCI-WZ7LrAsX\_uSFeD\_j5HXTdvUd5XQjGb9zGUskeVHrOrKHy69tUWaS-bb0DXoYkpd9cYG6s0cHJmtHgq3623kvGaav6eIeVURA9QRTVuV

@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

+ {static} WATERCOUNT

}

}

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