# Call graph

main

Launcher.launch

makeGame

gf = getGameFactory

getPlayerFactory

getSpriteStore

level = makeLevel

getMapParser

getLevelFactory

getSpriteStore

getGhostFactory

getSpriteStore

getBoardFactory

getSpriteStore

game = gf.createSinglePlayerGame

createPacMan

getSprites.getPacmanSprites

getSprites.getPacManDeathAnimation

PacManUiBuilder builder = new PacManUiBuilder.withDefaultButtons

addSinglePlayerKeys(builder)

pacManUI = builder.build(getGame)

pacManUI.start

PacManUI.nextFrame

The level of methods of this call graph varies depending on the method. The call graph goes as deep as the functions are accessing the resources package. This visualizes the structure of this project in a clear way.

iii) What we understood:

the packages game, player, level, board and ghost all contain a “Factory” class. External methods are accessing the packages via this class. The “Factory” class is responsible to create and return the corresponding object. All the “Factory” classes are accessing the spriteStore which is accessing the resources package. This is where it comes back together, all the information is stored in one central place (graphics, board layout etc.)