edu.sdccd.cisc191.wizardGame.utils.images::SpriteSheet
-image: BufferedImage
+SpriteSheet(BufferedImage image): ctor
+grabImage(int col, int row, int width, int height): BufferedImage

edu.sdccd.cisc191.wizardGame.utils.io::ResourceManager
+save(Serializable data, String fileName): void
+load(String fileName): Object

depends on

edu.sdccd.cisc191.wizardGame.utils.images::BufferedImageLoader
-image: BufferedImage
+loadImage(String path): BufferedImage

UML Class Diagram Resources:

https://www.cs.bsu.edu/homepages/pvgestwicki/misc/uml/
https://www.youtube.com/watch?v=UI6lqHOVHic
https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

Relationships
Composition
Aggregation
Association
Inheritance
Realization
Dependency

Visibility
- private
protected
~ package/default
+ public

Multiplicity
0..1 zero to one (optional)
n specific number
0..* zero to many
1..* one to many
m..n specific range

IO Subpackage

Image Subpackage

edu.sdccd.cisc191.wizardGame.save::SaveData
-serialVersionUID = 1L: long
+level: int
+lives: int

NOTE:

Should SaveData or LoadData
be under the same package?
Maybe under Features package?

Utils Package

Save Package

edu.sdccd.cisc191.wizardGame.objects::Bullet
-handler: Handler
+Bullet(int x, int y, ID id, Handler handler, int mx, int my, SpriteSheet ss): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Handler
~object = new LinkedList<GameObject>(): LinkedList<GameObject>
-up = false, down = false, right = false, left = false, pause = false: boolean
+tick(): void
+render(Graphics g): void
+addObject(GameObject tempObject): void
+removeObject(GameObject tempObject): void
+getObject(): LinkedList<GameObject>
+isUp(): boolean
+setUp(boolean up): void
+isDown(): boolean
+setDown(boolean down): void
+isRight(): boolean
+setRight(boolean right): void
+isLeft(): boolean
+setLeft(boolean left): void

edu.sdccd.cisc191.wizardGame.objects::Crate
-crate_image: BufferedImage
+Crate(int x, int y, ID id, SpriteSheet ss): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle

NOTE:

Suggestion: sub-packages for...

-world objects
-characters

edu.sdccd.cisc191.wizardGame.objects::Block
~game: Game
-block_image: BufferedImage
+Block(int x, int y, ID id, SpriteSheet ss, Game game): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle

depends on

edu.sdccd.cisc191.wizardGame.events::MouseInput
-handler: Handler
-camera: Camera
-game: Game
-ss: SpriteSheet
-cs: SpriteSheet
-rtrx, rty: int
-x, y: int
+MouseInput(Handler handler, Camera camera, Game game, SpriteSheet ss, SpriteSheet cs): ctor
+mousePressed(MouseEvent e): void
-fireBullet(): void

edu.sdccd.cisc191.wizardGame::Game
-frame: Window
-ammo, hp, lives: int
-isWizardDead = false: boolean
+Game(): ctor
+init(): void
+getFrame(): Window
+getAmmo(): int
+getLives(): int
+getHp(): int
+hasWizardDied(): boolean
+setHp(int hp): void
+setAmmo(int ammo): void
+setLives(int lives): void
+incAmmo(int inc): void
+decAmmo(): void
+decHp(): void
+decLives(): void
+wizardDied(): void
+wizardRespawn(): void
+main(String[] args): void

Object Package

edu.sdccd.cisc191.wizardGame.objects::GameObject
#x, y: int
#velX = 0, velY = 0: float
#id: ID
#ss: SpriteSheet
+GameObject(int x, int y, ID id, SpriteSheet ss): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getId(): ID
+setId(ID id): void
+getX(): int
+setX(int x): void
+getY(): int
+setY(int y): void
+getVelX(): float
+setVelX(float velX): void
+getVelY(): float
+setVelY(float velY): void

edu.sdccd.cisc191.wizardGame.objects::Wizard
~handler: Handler
~game: Game
-wizard_image = new BufferedImage[4]: BufferedImage[]
~anim: Animation
+Wizard(int x, int y, ID id, Handler handler, Game game, SpriteSheet cs): ctor
+tick(): void
+collision(): void
+place_free(int x, int y, Rectangle myRect, Rectangle otherRect): boolean
+render(Graphics g): void
+getBounds(): Rectangle

edu.sdccd.cisc191.wizardGame.events::KeyInput
~handler: Handler
+KeyInput(Handler handler): ctor
+keyPressed(KeyEvent e): void
+keyReleased(KeyEvent e): void

Event Package

edu.sdccd.cisc191.wizardGame.objects::Ent
-handler: Handler
-game: Game
-ent_image = new BufferedImage[8]: BufferedImage[]
-cs: SpriteSheet
~anim: Animation
~r = new Random(): Random
~choose = 0: int
~hp = 100: int
+Ent(int x, int y, ID id, Handler handler, Game game, SpriteSheet cs): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getBoundsBig(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Totem
-totem_image: BufferedImage
+Totem(int x, int y, ID id, SpriteSheet ss): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Enemy
-handler: Handler
-enemy_image = new BufferedImage[8]: BufferedImage[]
~anim: Animation
~r = new Random(): Random
~choose = 0: int
~hp = 100: int
+Enemy(int x, int y, ID id, Handler handler, SpriteSheet cs): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getBoundsBig(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Hound
-handler: Handler
-hound_image = new BufferedImage[8]: BufferedImage[]
~anim: Animation
~r = new Random(): Random
~choose = 0: int
~hp = 100: int
+Hound(int x, int y, ID id, Handler handler, SpriteSheet cs): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getBoundsBig(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Knight
-handler: Handler
-game: Game
-knight_image = new BufferedImage[8]: BufferedImage[]
~anim: Animation
~r = new Random(): Random
~choose = 0: int
~hp = 100: int
~counter = 0: int
~px: int
~py: int
+Knight(int x, int y, ID id, Handler handler, Game game, SpriteSheet cs): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getBoundsBig(): Rectangle

edu.sdccd.cisc191.wizardGame.objects::Minion
-handler: Handler
-minion_image = new BufferedImage[8]: BufferedImage[]
~anim: Animation
~r = new Random(): Random
~choose = 0: int
~hp = 100: int
+Minion(int x, int y, ID id, Handler handler, SpriteSheet cs): ctor
+tick(): void
+render(Graphics g): void
+getBounds(): Rectangle
+getBoundsBig(): Rectangle

Features Package

TO BE IMPLEMENTED

GUI Package

Animation Subpackage

edu.sdccd.cisc191.wizardGame.gui.anim::Camera
-x, y: float
+Camera(float x, float y): ctor
+tick(GameObject object): void
+getX(): float
+setX(float x): void
+getY(): float
+setY(float y): void

edu.sdccd.cisc191.wizardGame.gui.anim::Animation
-images: BufferedImage[]
-interval, index: int
-timer, now, lastTime: long
+Animation(BufferedImage[] images, int interval): ctor
+tick(): void
+render(Graphics g, int x, int y, int width, int height): void

Screen Subpackage

edu.sdccd.cisc191.wizardGame.gui.screen::Window
-game: Game
-frameWidth: int
-frameHeight: int
-layeredPane: JLayeredPane
-allPanels = new HashMap<String, GeneralPanel>(): Map<String, GeneralPanel>
-lastOpenPanel: String
-currOpenPanel: String
-device = GraphicsEnvironment.getLocalGraphicsEnvironment().getScreenDevices()[0]: GraphicsDevice
+Window(Game game, int width, int height, String title): ctor
+init(): void
#changePanel(String panelName): void
#showLoadScreen(): void
+quitGame(): void
+getGame(): Game
+getAllPanels(): Map<String, GeneralPanel>
+getPanel(String panelName): GeneralPanel
+getLastOpenPanel(): String
+getCurrOpenPanel(): String

edu.sdccd.cisc191.wizardGame.gui.screen::GeneralPanel
-background: BufferedImage
-orientation: BGOrientation
#frame: Window
+GeneralPanel(Window frame): ctor
#addBackground(String backgroundName, BGOrientation orientation): void
#paintComponent(Graphics g): void

level Subpackage

edu.sdccd.cisc191.wizardGame.gui.screen::LoadPanel
~timer = new Timer(50, this): Timer
~rand = new Random(): Random
~x = 0: int
~y = 0: int
~xCenter = this.getWidth() / 2: double
~yCenter = this.getHeight() / 2: double
~furthestDistance = (int) Math.sqrt(Math.pow(xCenter - 0, 2) + Math.pow(yCenter - 0, 2)): int
~scene: int
~coverage = 0: int
~isShrinking = false: boolean
~label: JLabel
-charSheet: SpriteSheet
-mainSheet: SpriteSheet
-wizardImages = new Image[4]: Image[]
-mushroomImages = new Image[8]: Image[]
-knightImages = new Image[8]: Image[]
-rotateFireImages = new Image[4]: Image[]
-terrain: Image
~wizardIndex = 0: int
~mushroomIndex = 0: int
~knightIndex = 0: int
~rotateFireIndex = 0: int
+LoadPanel(Window frame): ctor
+start(): void
+stop(): void
+fillSpriteImage(Image[] imageArr, int startLoc, int row, int width, int height, SpriteSheet ss): void
+vignette(Color color, Graphics2D g2d, int coverage): void
+drawScene(Graphics2D g2d, int scene): void
+drawTerrain(Graphics2D g2d, Image terrain): void
+updateSpriteIndex(Image[] spriteImages, int index): int
+paintComponent(Graphics g): void
+actionPerformed(ActionEvent e): void

edu.sdccd.cisc191.wizardGame.gui.screen::PausePanel
#resumeBtn, helpBtn, quitBtn: JButton
-buttonSize = new Dimension(200, 75): Dimension
+PausePanel(Window frame): ctor
#addButtonListeners(): void

edu.sdccd.cisc191.wizardGame.gui.screen::HelpPanel
#backBtn: JButton
-buttonSize = new Dimension(400, 50): Dimension
+HelpPanel(Window frame): ctor
#addButtonListeners(): void

edu.sdccd.cisc191.wizardGame.gui.screen::MenuPanel
#playBtn, helpBtn, quitBtn: JButton
-buttonSize = new Dimension(200, 75): Dimension
+MenuPanel(Window frame): ctor
#addButtonListeners(): void

edu.sdccd.cisc191.wizardGame.gui.screen::GamePanel
-game: Game
-frame: Window
-canvas = new Canvas(): Canvas
-isRunning = false: boolean
-gameThread: Thread
-currLevel: AbstractLevel
-camera: Camera
-handler: Handler
-loader = new BufferedImageLoader(): BufferedImageLoader
-ss: SpriteSheet
-cs: SpriteSheet
-pauseBtn, respawnBtn: JButton
+GamePanel(Window frame): ctor
+start(): void
+stop(): void
+run(): void
#tick(): void
+render(): void
+update(): void
+resetGame(): void
+showRespawn(): void
+respawn(): void
+isGameRunning(): boolean
+getLevel(): AbstractLevel
+getHandler(): Handler
+getCamera(): Camera
+setHandler(): void
+setLevel(int levelNumb): void
#addButtonListeners(): void
#paintComponent(Graphics g): void

edu.sdccd.cisc191.wizardGame.gui.screen.levels::AbstractLevel
#game: Game
#gamePanel: GamePanel
#levelNum: int
#loader = new BufferedImageLoader(): BufferedImageLoader
#camera: Camera
#handler: Handler
#ss = new SpriteSheet(loader.loadImage("/main_sheet.png")): SpriteSheet
#cs = new SpriteSheet(loader.loadImage("/wizard_sheet.png")): SpriteSheet
#levelImages = new LinkedList<BufferedImage>(): LinkedList<BufferedImage>
#floorOne = ss.grabImage(6, 6, 32, 32): BufferedImage
#floorTwo = ss.grabImage(7, 2, 32, 32): BufferedImage
#floorThree = ss.grabImage(6, 6, 32, 32): BufferedImage
#floorFour = ss.grabImage(7, 2, 32, 32): BufferedImage
#floorFive = ss.grabImage(6, 6, 32, 32): BufferedImage
#currentMap: BufferedImage
#level: BufferedImage
#livesImage = cs.grabImage(13, 8, 32, 32): BufferedImage
+AbstractLevel(Game game, GamePanel gamePanel): ctor
+bufferLevelImages(): void
+loadLevel(BufferedImage image): void
+respawn(): void
+tick(): void
+render(Graphics2D g): void
+resetLevel(): void
+getHp(): int
+getAmmo(): int
+getLives(): int
+getHandler(): Handler
+getCamera(): Camera
+getLevel(int numLevel): BufferedImage
+decHp(): void
+decLives(): void
+incAmmo(int inc): void

edu.sdccd.cisc191.wizardGame.gui.screen.levels::LevelOne
+LevelOne(Game game, GamePanel gamePanel): ctor

edu.sdccd.cisc191.wizardGame.gui.screen.levels::LevelTwo
+LevelTwo(Game game, GamePanel gamePanel): ctor