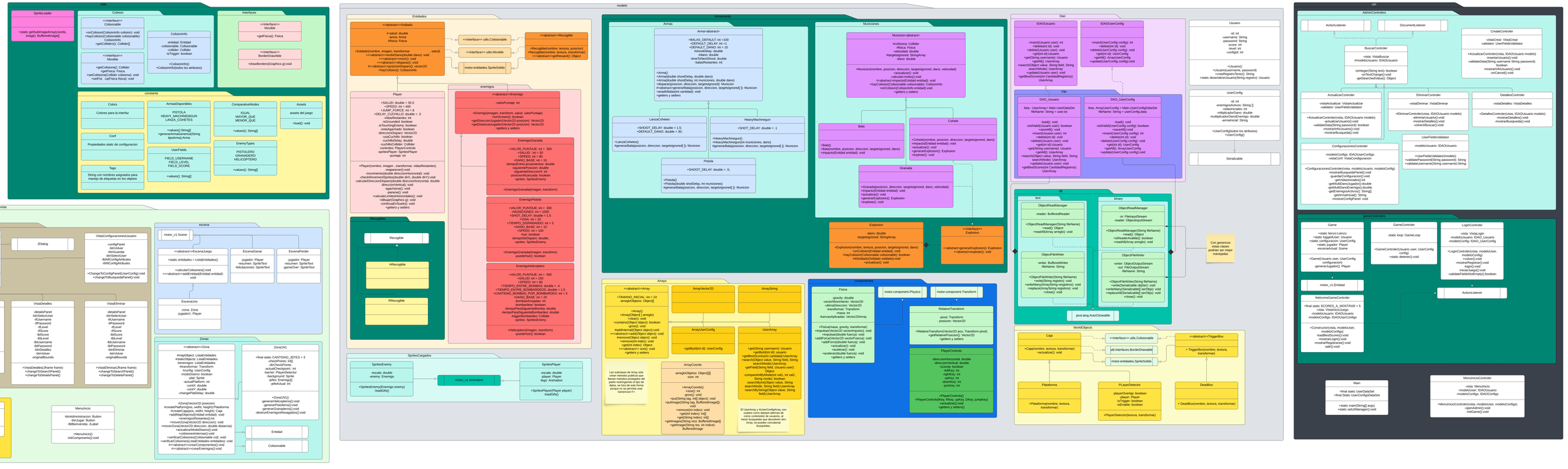
util
SpriteLoader
+static getSubImageArray(coords, image): BufferImage[]
Colision
<<interface>>
Colisionable
+onColision(ColisionInfo colision): void
+hayColision(Colisionable colisionable): Colisionint
+getColisionInfo(): Collider[]
Colisionint
-entidad: Entidad
-colisionable: Colisionable
-collider: Collider
-isTrigger: boolean
+Colisionint(s)
+Colisionint(todos los atributos)
<<interface>>
Movible
+getCollisionsa(): Collider
+setCollisionsa(Collider collision): void
+setFisi ca(Fisica fisica): void
Interfaces
<<interface>>
Movible
+getFisica(): Fisica
<<interface>>
BorderDrawable
+drawBorders(Graphics g): void
constants
Colors
Colores para la interfaz
Conf
Propiedades static de configuracion
Tags
String con nombres asignados para manejo de etiquetas en los objetos
ArmasDisponibles
PISTOLA
HEAVY_MACHINEMGUN
LANZA_COHETES
+values() [String]
+generarArmaGenerica(String tipoArma):Arma
UserFields
FIELD_USERNAME
FIELD_LEVEL
FIELD_SCORE
+values() : String[]
ComparativeModes
IGUAL
MAYOR_QUE
MENOR_QUE
+values() : String[]
EnemyTypes
PISTOLERO
BAZOOKERO
HELICOPTERO
+values() : String[]
Assets
assets del juego
+load() : void
modelo
Entidades
<<abstractor>>Soldado
salud: double
-arma: Arma
#fisica: Fisica
+Soldado(nombre, imagen, transformar, salud)
+<<constructor>>reviver(double dano): void
+<<abstractor>>mover(): void
+<<abstractor>>disparar(): void
#+<<abstractor>>getDireccion(): Vector2D
+hayColision(): ColisionInfo
<<interface>> utils.Colisionable
<<interface>> utils.Movible
motor.entidades.SpriteSolido
<<abstractor>>Recogible
+Recogible(nombre, textura, posicion)
+Recogible(nombre, textura, transformar)
+<<abstractor>>getRewards(): Object
Player
+SALUD: double = 30.0
+SPEED: int = 400
+JUMP_FORCE: int = 8
+DELAY_CUCHILLO: double = .3
-isJumpingPressed: boolean
-isGrounded: boolean
-isTouchingEnemy: boolean
-estaAgachado: boolean
-direccionDispara: Vector2D
-cuchillo: Cuchillo
-cuchilloDelay: double
-cuchilloCollider: Collider
-controles: PlayerControls
-spritesPlayer: SpritesPlayer
-puntaje: int
+Player(nombre, imagen , transformar, vidasRestantes)
+<<constructor>>(): void
-movimiento(double direccionHorizontal): void
-checkMovementSprites(double dirX, double dirY):void
-calcularDireccionDisparo(double direccionHorizontal, double
direccionVertical): void
-agacharse(): void
-levantarse(): void
+calcularLimitesPantalla(): void
+dibujar(Graphics g): void
-continuarEnSuelo(): void
+getters y setters
enemigos
<<abstractor>>Enemigo
-valorPuntaje: int
+Enemigo(imagen, transform, salud, valorPuntaje)
+isInsideGround(): boolean
+getDireccionJugador(Vector2D posicion): Vector2D
+getDistanciaJugador(Vector2D posicion): Vector2D
+getters y setters
EnemigoGranada
+VALOR_PUNTAJE: int = 300
+SALUD: int = 30
+VIDA: int = 30
+DAMO_BASE: int = 10
-temporizadorLanzamiento: double
-siguientePosicion: double
-siguienteDireccionX: int
-posicionAlcanzada: boolean
-sprites: SpritesEnemy
+EnemigoGranada(imagen, transform)
EnemigoPistola
+VALOR_PUNTAJE: int = 300
+SALUD: int = 30
+SHOT_DELAY: double = 1.5
+VIDA: int = 20
+TIEMPO_DISPARANDO: int = 3
+DAMO_BASE: int = 10
-isShot: boolean
-hurt: boolean
-tempoDeDispara: double;
-sprites: SpritesEnemy
+EnemigoGranada(imagen, transform)
-puedeHurt(): boolean
EnemigoHelicoptero
+VALOR_PUNTAJE: int = 500
+SALUD: int = 150
+VIDA: int = 150
+TIEMPO_ENTRE_BOMBARDEO: double = .4
+TIEMPO_ENTRE_BOMBARDEOS: double = 1.5
+CANTIDAD_BOMBAS_POR_BOMBARDEO: int = 3
+DAMO_BASE: int = 20
-bombasArmadas: int
-bombasLanzadas: int
-tempoPuedeSaquarenBombardeo: double
-tempoPuedeSaquarenBombardeo: double
-triggerBombardeo: Collider
-sprites: SpritesEnemy
+Helicoptero(imagen, transform)
-puedeHurt(): boolean
Recogibles
Recogible
HRecogible
RRecogible
Armas
Arma<<abstract>
+BALAS_DEFAULT: int =100
+DAMO_DEFAULT: int = 10
+DEFAULT_DANO: int = 10
#shootDelay: double
#dano: double
-timeToNextShoot: double
-balasRestantes: int
+Arma()
+Arma(double shootDelay, double dano)
+Arma(double shootDelay, int municiones, double dano)
+dispararposicion, direccion, targetsignored): Muncion
+<<abstract>>generarBala(posicion, direccion, targetsignored[]): Muncion
+arma(balasRestantes)
+getters y setters
LanzaCohetes
+SHOOT_DELAY: double = 1.5;
-DEFAULT_DANO: double = 30;
+LanzaCohetes()
#generarBala(posicion, direccion, targetsignored[]): Muncion
HeavyMachinegun
+SHOT_DELAY: double = .1
+HeavyMachinegun()
+HeavyMachinegun(int municiones, dano)
#generarBala(posicion, direccion, targetsignored[]): Muncion
Pistola
+SHOOT_DELAY: double = .5;
+Pistola()
+Pistola(double shotDelay, int municiones)
#generarBala(posicion, direccion, targetsignored[]): Muncion
Municiones
Municion<<abstract>
#colisiona: Collider
#fisica: Fisica
-velocidad: double
#targetsignored: StringArray
-dano: double
+Muncion(nombre, posicion, direccion, targetsignored, dano, velocidad)
+actualizar(): void
-calcularLimites(): void
#+abstractor>colisiona(Colisionable entidad): void
+hayColision(Colisionable colisionable): ColisionInfo
+onColision(ColisionInfo entidad) void
+getters y setters
Bala
+Bala()
+Bala(nombre, posicion, direccion, targetsignored, dano)
+impacto(Entidad entidad): void
Cohete
+Cohete(nombre, posicion, direccion, targetsignored, dano)
+impacto(Entidad entidad): void
+actualizar() void
+generarExplosion(): Explosion
+explotar() void
Granada
+Granada(posicion, direccion, targetsignored, dano, velocidad)
+impacto(Entidad entidad): void
+actualizar() void
+generarExplosion(): Explosion
+explotar() void
Explosion
-dano: double
-targetsignored: StringArray
+Explosion(nombre, textura, posicion, targetsignored, dano)
+actualizar(): void
+hayColision(Colisionable colisionable): boolean
#+abstractor>onColision(Colisionable entidad): void
+actualizar(): void
<<interface>>
Explosivo
+<abstractor>generarExplosion(): Explosion
+<abstractor>explotar(): void
Arrays
<<abstractor>>Array
+TAMANIO_INICIAL: int = 10
-arregloObjetos: Object[]
+Array()
+Array(int[] pargs)
+clear(): void
+contains(Object object): boolean
+grow(): void
#addInternal(Object object): void
+<<abstractor>>add(Object object): void
#removeInternal(Object object): void
+remove(int index): void
+get(int index): Object
+<<abstractor>> sort(): void
+getters y setters
ArrayVector2D
ArrayString
ArrayUserConfig
+getBy(int id): UserConfig
UserArray
+getBy(String username): Usuario
+getBy(int id): Usuario
+getBetween(int centidad):UserArray
+search(Object value, String field, String searchMode):UserArray
-getWhit(String field, Usuario user): boolean
-compareWhitSearchMode(val1, int val2, String mode): boolean
-searchBy(int[]Object value, String searchMode, String field):UserArray
-searchBy(String[]Object value, String field):UserArray
El UserArray y ArrayConfigArray son usados como dataset ademas de ser usados como arrays. Los usuarios se hacen busquedas que devuelven otro Array, se pueden concatenar busquedas
AreasDeJuego
Fisica
-gravity: double
-vectorMovimiento: Vector2D
-velocidad: double = 0.0
-transformar: Transform
-masa: int
-fuerzasAplicadas: Vector2DArray
+Fisica(masa, gravity, transformar)
+impulsar(Vector2D vectorImpulso): void
+impulsar(double fuerza): void
+addForce(Vector2D vectorFuerza): void
+addForce(double fuerza): void
+actualizar(): void
+aceleracion(): void
+aceleracion(double fuerza): void
+getters y setters
motor.component.Physics
motor.component.Transform
RelativeTransform
-giro: Transform
- posicion: Vector2D
+RelativeTransform(Vector2D pos, Transform giro)
+getRelativePosicion(): Vector2D
+getters y setters
PlayerControls
-direccionHorizontal: double
-direccionVertical: double
-isJump: boolean
-isDrag: boolean
-rightKey: int
-upKey: int
-downKey: int
-jumpKey: int
+PlayerControls()
+PlayerControls(Key RKey, upKey, DKey, jumpKey)
+actualizar() void
+getters y setters
Dao
IDAOUsuario
+insert(Usuario user): int
+delete(int id): void
+delete(Usuario user): void
+get(int id): Usuario
+getCliteng username): Usuario
+getAll(): UserArray
+search(Object value, String field, String searchMode): UserArray
+update(Usuario user): void
+getBestScoreInt CantidadRegistros): UserArray
IDAOUserConfig
+insert(UserConfig config): int
+delete(int id): void
+delete(UserConfig config): void
+get(int id): UserConfig
+getAll(): ArrayUserConfig
+update(UserConfig config): void
File
DAO_Usuario
lista : UserArray= Main.UserDataSet
fileName: String = user.txt
-load() : void
-isInvalid(Usuario user): boolean
-saveAll() : void
+insert(Usuario user): int
+delete(int id): void
+delete(Usuario user): void
+get(int id):Usuario
+get(String username): Usuario
+getAll() :UserArray
+search(Object value, String field, String searchMode): UserArray
+update(Usuario user): void
+getBestScoreInt CantidadRegistros): UserArray
DAO_UserConfig
lista: ArrayUserConfig = Main.UserConfigDataSet
fileName: String = userConfig.data
-load() : void
-isInvalid(UserConfig config): boolean
-saveAll() : void
+insert(UserConfig config): int
+delete(int id): void
+delete(UserConfig config): void
+get(int id): UserConfig
+getAll() :ArrayUserConfig
+update(UserConfig config): void
Usuario
-id: int
-username: String
-password: String
-score: int
-level: int
-configId: int
+Usuario()
+Usuario(username, password)
+<<readRegistersTexto>> String
+static deserializeUsuario(String registro): Usuario
UserConfig
-id: int
-enemigosActivos: String []
-vidaJugador: double
-velocidadJugador: double
-multiplicadorDanoEnemigo: double
-armaInicial: String
+UserConfig()
+UserConfig(todos los atributos)
+UserConfig()
Serializable
io
text
ObjectReadManager
-reader: BufferedReader
+<ObjectReadManager(String fileName)
+read(): Object
+readAll(Array array): void
ObjectWriterWriter
-writer: BufferWriterWriter
-fileName: String
+ObjectWriterWriter(String fileName)
+write(String registro): void
+writeMany(Array/String registros): void
+replaceArray(Array/String registros): void
+close(): void
binary
ObjectReadManager
in : FileInputStream
-reader: ObjectInputStream
+ObjectReadManager(String fileName)
+read(): Object
+isReadableAvailable() : boolean
+readAll(Array array): void
ObjectWriterWriter
-writer: ObjectOutputStream
-fileName: String
+ObjectWriterWriter(String fileName)
+write(Serializable obj,bar): void
+writeMany(Serializable[] serObjs): void
+replaceAll(Serializable[] serObjs): void
+close(): void
java.lang.AutoCloseable
WorkObjects
Caja
+Caja(nombre, textura, transformar)
+actualizar() :void
<<interface>> utils.Colisionable
utils.interfaces.BorderDrawable
motor.entidades.SpriteSolido
<<abstractor>>TriggerBox
+ TriggerBox(nombre, textura, transformar)
Plataforma
+Plataforma(nombre, textura, transformar)
PLayerDetector
-playerOverlap: boolean
-player: Player
-isTrigger: boolean
-isEnable: boolean
+PlayerDetector(textura, transformar)
DeadBox
+ DeadBox(nombre, textura, transformar)
Con genericos estas clases podran ser mejor manejadas
ctrl
AdminControlers
ActionListener
DocumentListener
BuscarController
-vista: VistaBuscar
#modelo:Usuario IDAOUsuario
+listHrger(String text): boolean
-onTextChange(): void
-getSearchedValue(): Object
CreateController
-vistaCrear: VistaCrear
-validator: UserFieldsValidator
+ActualizarController(vista, IDAOUsuario modelo)
+startartUsuario() void
-validateDataSize(String password): boolean
-mostrarInfoUsuario() void
-onCancel() void
ActualizarController
-vistaActualizar: VistaActualizar
-validator: UserFieldsValidator
+ActualizarController(vista, IDAOUsuario modelo)
+actualizarUsuario() void
-validateDataSize(String password): boolean
-mostrarInfoUsuario() void
-mostrarBusquedaA() void
EliminarController
-vistaEliminar: VistaEliminar
+EliminarController(vista, IDAOUsuario modelo)
-eliminarUsuario() void
-mostrarDetalles() void
-volverAlBuscar() void
DetallesController
-vistaDetalles: VistaDetalles
+DetallesController(vista, IDAOUsuario modelo)
-mostrarDetalles() void
-mostrarBusquedaAs() void
ConfiguracionesController
-modeloConfig: IDAOUserConfig
-vistaCont: VistaConfiguracion
+ConfiguracionesController(vista, modeloUsuario, modeloConfig)
-mostrarBusquedaPanel() void
-guardarConfiguracion() void
-getVidasIniciales() int
-getVelocidadJugador() double
-getMultiplicadorDanoEnemigo() double
-getEnemigosActivos() String[]
-getArmaInicial() String
-mostrarConfigPanel: void
UserFieldsValidator
-modeloUsuarios: IDAOUsuario
+UserFieldsValidator(modelo)
+validatePassword(String password): String
+validateUsername(String username): String
Main
+final static UserDataSet
+final Static UserConfigDataSet
+static main(String[] args)
+static setUsManager() void
MenuInicioController
-vista: MenuInicio
-modeloUser: IDAOUsuario
-modeloConfig: IDAOConfig
+MenuInicioController(vista, modeloUser, modeloConfig)
-openAdmin() void
-initGame() void
vista
Admin
AdminPanel
-contentPane: JPanel
-BTNAlto: JLabel
-BtnBuscar: Button
-BtnBuscar: Button
-btnBuscar: Button
-btnActualizar: Button
-btnEliminar: Button
-btnConfiguraciones: Button
+AdminPanel()
-initComponents(): void
VistaBuscar
-busquedaPanel: JPanel
-searchTextField: JTextField
-listaBusquedaA: JScrollPane
-lblCriterios: JLabel
-lblSearchMode: JLabel
-lblBusqueda: JLabel
-comboBoxCriterios: JComboBox<String>
-comboBoxSlModes:JComboBox<String>
+VistaBuscar()
+changeToConfigPanel(UserConfig): void
+changeToBusquedaPanel() void
JDialog
VistaConfiguracionesUsuario
-contentPane
-btnSaber
-btnCancel
-btnGuardar
-btnSelectUser
-lblAltoCelda
-lblAttributes
-lblAlcoHolAttributes
-lblAlCoHolAttributes
+VistaBuscar(JFrame frame)
-initComponents() void
+changeToConfigPanel(UserConfig): void
+changeToBusquedaPanel() void
VistaActualizar
-actualizarPanel
-btnSelectUser
-tfUsername
-tfUsername
-tfUsername
-tfPassword
-btnActualizar
-btnCancelar
-lblUsernameError
-lblPasswordError
+VistaActualizar(JFrame frame)
+changeToSearchPanel()
+changeToActualizarPanel()
VistaCrear
-createPanel
-btnSelectUser
-tfUsername
-tfUsername
-tfUsername
-tfPassword
-btnCreate
-lblUsernameError
-lblPasswordError
+VistaCrear(JFrame frame)
+changeToSearchPanel()
+changeToCreatePanel()
VistaDetalles
-detailsPanel
-btnSelectUser
-tfUsername
-tfUsername
-tfLevel
-tfScore
-tfScore
-tfLevel
-lblUsernameError
-lblPasswordError
-btnDetalles
-btnSaber
-originalBounds
+VistaDetalles(JframeFrame)
+changeToSearchPanel()
+changeToDetallePanel()
VistaEliminar
-detailsPanel
-btnSelectUser
-tfUsername
-tfUsername
-tfLevel
-tfScore
-tfScore
-tfLevel
-lblUsernameError
-lblPasswordError
-btnEliminar
-btnSaber
-originalBounds
+VistaEliminar(JframeFrame)
+changeToSearchPanel()
+changeToDeletePanel()
game
VistaInicioJuego
-contentPane: JPanel
-lblNombre: JLabel
-lblPuntaje: JLabel<String>
-lblPuntajes: JLabel
-btnRegistrar: Button
-btnIniciarJuego: Button
-btnSaber: Button
+VistaInicioJuego()
-initComponents()
VistaLogin
-lbl y lf de password y username
-btnRegistrar
-btnlogin
MenuInicio
-btnAdministracion: Button
-btnJugar: Button
-lblBienvenida: JLabel
+MenuInicio()
-initComponents() void
escena
motor_v1.Scene
<<abstractor>>EscenaJuego
+static entidades = List<IHitadesc>
+calcularColisiones():void
<<>>abstractor>>addIHitadsEntidad(Entidad entidad): void
EscenaGanar
-jugador: Player
-resumen: SpriteText
-felicitaciones: SpriteText
EscenaPerder
-jugador: Player
-resumen: SpriteText
-gameOver: SpriteText
EscenaUno
-zona: Zona
-jugador1: Player
Zonas
<<abstractor>>Zona
#mapObject: List<Entidades
#staticObjects: List<Entidades
#enemigos: List<Entidades
#transformar: Transform
-isDrawBorders: boolean
-modoDisarro: boolean
-plat: Sprite
-actualPlatform: int
-contX: double
-contY: double
-changePlatDelay: double
#Zona(Vector2D posicion)
#crearPlataformas(ps, width, height):Plataforma
#crearCaja(ps, width, height): Caja
+addMapObjects(IEntidad entidad): void
+enemigosDisarrollados()int
+moverZona(Vector2D direccion): void
+moverZona(Vector2D direccion, double distancia)
+actualizarModoDisarro():void
+colisionesInternas():void
+verificarColisiones(Colisionable col): void
+verificarColisiones(List<Entidades entidades): void
+verificarCollisionesEnemigos(Entidad e): void
+<<abstractor>>crearEnemigos():void
Zona1N1
+final static CANTIDAD_JEFES = 3
-isDrawBorders: boolean
-deCheckPoints:
-actualCheckpoint: int
-barrier: PlayerDetector
-background: Sprite
-plata: Energing[]
-getAcUsed: int
+Zona1N1()
-generarmiEscalerasc():void
-generarPrototeros():void
-generarGranaderos():void
-desalvarEnemigosResagados():void
Entidad
Colisionable
SpritesEnemy
-escala: double
-enemy: Enemigo
+SpritesEnemy(Enemigo enemy)
-loadGifs()
motor_v1.Animation
SpritesPlayer
-escala: double
-player: Player
-legs: Animation
+SpritesPlayer(Player player)
-loadGifs()
SpritesCargados
Clases