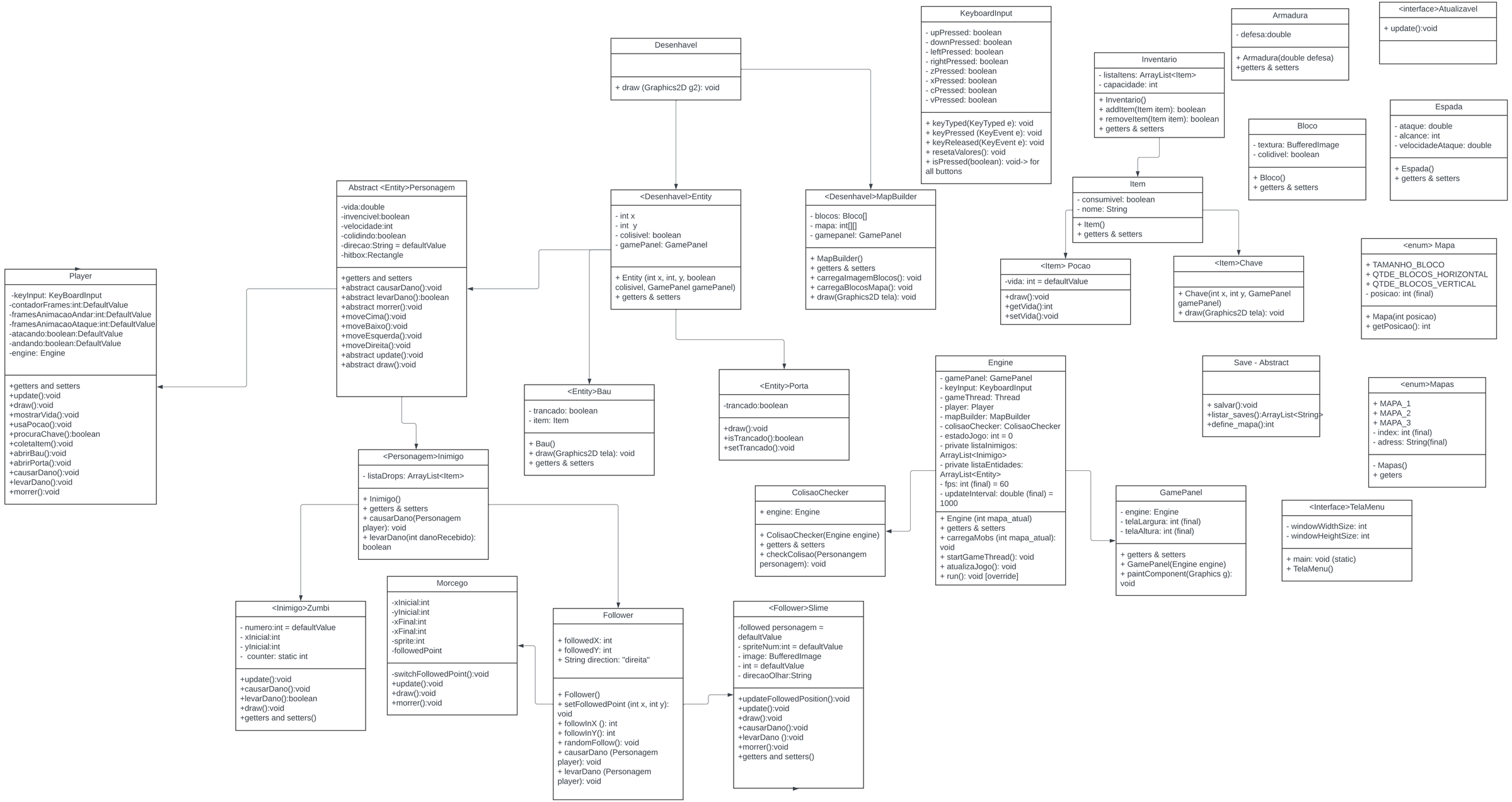
Desenhavel
+ draw (Graphics2D g2): void
KeyboardInput
- upPressed: boolean
- downPressed: boolean
- leftPressed: boolean
- rightPressed: boolean
- zPressed: boolean
- xPressed: boolean
- cPressed: boolean
- vPressed: boolean
+ keyTyped(KeyTyped e): void
+ keyPressed (KeyEvent e): void
+ keyReleased(KeyEvent e): void
+ resetaValores(): void
+ isPressed(boolean): void-> for all buttons
Armadura
- defesa:double
+ Armadura(double defesa)
+getters & setters
<interface>Atualizavel
+ update():void
Inventario
- listaItens: ArrayList<Item>
- capacidade: int
+ Inventario()
+ addItem(Item item): boolean
+ removeItem(Item item): boolean
+ getters & setters
Bloco
- textura: BufferedImage
- colidivel: boolean
+ Bloco()
+ getters & setters
Espada
- ataque: double
- alcance: int
- velocidadeAtaque: double
+ Espada()
+ getters & setters
Item
- consumivel: boolean
- nome: String
+ Item()
+ getters & setters
<Item> Pocao
-vida: int = defaultValue
+draw():void
+getVida():int
+setVida():void
<Item>Chave
+ Chave(int x, int y, GamePanel gamePanel)
+ draw(Graphics2D tela): void
<enum> Mapa
+ TAMANHO_BLOCO
+ QTDE_BLOCOS_HORIZONTAL
+ QTDE_BLOCOS_VERTICAL
- posicao: int (final)
+ Mapa(int posicao)
+ getPosicao(): int
<Desenhavel>Entity
- int x
- int y
- colisivel: boolean
- gamePanel: GamePanel
+ Entity (int x, int, y, boolean colisivel, GamePanel gamePanel)
+ getters & setters
<Desenhavel>MapBuilder
- blocos: Bloco[]
- mapa: int[][]
- gamepanel: GamePanel
+ MapBuilder()
+ getters & setters
+ carregaImagemBlocos(): void
+ carregaBlocosMapa(): void
+ draw(Graphics2D tela): void
Abstract <Entity>Personagem
-vida:double
-invencivel:boolean
-velocidade:int
-colidindo:boolean
-direcao:String = defaultValue
-hitbox:Rectangle
+getters and setters
+abstract causarDano():void
+abstract levarDano():boolean
+abstract morrer():void
+moveCima():void
+moveBaixo():void
+moveEsquerda():void
+moveDireita():void
+abstract update():void
+abstract draw():void
Player
-keyInput: KeyBoardInput
-contadorFrames:int:DefaultValue
-framesAnimacaoAndar:int:DefaultValue
-framesAnimacaoAtaque:int:DefaultValue
-atacando:boolean:DefaultValue
-andando:boolean:DefaultValue
-engine: Engine
+getters and setters
+update():void
+draw():void
+mostrarVida():void
+usaPocao():void
+procuraChave():boolean
+coletaItem():void
+abrirBau():void
+abrirPorta():void
+causarDano():void
+levarDano():void
+morrer():void
<Entity>Bau
- trancado: boolean
- item: Item
+ Bau()
+ draw(Graphics2D tela): void
+ getters & setters
<Entity>Porta
-trancado:boolean
+draw():void
+isTrancado():boolean
+setTrancado():void
Engine
- gamePanel: GamePanel
- keyInput: KeyboardInput
- gameThread: Thread
- player: Player
- mapBuilder: MapBuilder
- colisaoChecker: ColisaoChecker
- estadoJogo: int = 0
- private listaInimigos: ArrayList<Inimigo>
- private listaEntidades: ArrayList<Entity>
- fps: int (final) = 60
- updateInterval: double (final) = 1000
+ Engine (int mapa_atual)
+ getters & setters
+ carregaMobs (int mapa_atual): void
+ startGameThread(): void
+ atualizaJogo(): void
+ run(): void [override]
Save - Abstract
+ salvar():void
+listar_saves():ArrayList<String>
+define_mapa():int
<enum>Mapas
+ MAPA_1
+ MAPA_2
+ MAPA_3
- index: int (final)
- adress: String(final)
- Mapas()
+ geters
<Personagem>Inimigo
- listaDrops: ArrayList<Item>
+ Inimigo()
+ getters & setters
+ causarDano(Personagem player): void
+ levarDano(int danoRecebido): boolean
ColisaoChecker
+ engine: Engine
+ ColisaoChecker(Engine engine)
+ getters & setters
+ checkColisao(Personangem personagem): void
GamePanel
- engine: Engine
- telaLargura: int (final)
- telaAltura: int (final)
+ getters & setters
+ GamePanel(Engine engine)
+ paintComponent(Graphics g): void
<Interface>TelaMenu
- windowWidthSize: int
- windowHeightSize: int
+ main: void (static)
+ TelaMenu()
<Inimigo>Zumbi
- numero:int = defaultValue
- xInicial:int
- yInicial:int
- counter: static int
+update():void
+causarDano():void
+levarDano():boolean
+draw():void
+getters and setters()
Morcego
-xInicial:int
-yInicial:int
-xFinal:int
-xFinal:int
-sprite:int
-followedPoint
-switchFollowedPoint():void
+update():void
+draw():void
+morrer():void
Follower
+ followedX: int
+ followedY: int
+ String direction: "direita"
+ Follower()
+ setFollowedPoint (int x, int y): void
+ followInX (): int
+ followInY(): int
+ randomFollow(): void
+ causarDano (Personagem player): void
+ levarDano (Personagem player): void
<Follower>Slime
-followed personagem = defaultValue
- spriteNum:int = defaultValue
- image: BufferedImage
- int = defaultValue
- direcaoOlhar:String
+updateFollowedPosition():void
+update():void
+draw():void
+causarDano():void
+levarDano ():void
+morrer():void
+getters and setters()