

# GameMenu Constants + HIGHLIGHTER COLOUR: int TEXT COLOUR: int Fields «final» largeFont: PFont midLevel: boolean «final» sketch : DontDrown «final» smallFont : PFont **Properties** menuState: MenuState Constructors GameMenu( DontDrown ): void Methods render(): void resetPage(): void resolveClick(): void scroll( float ) : void scrollWrapper(int): void updateLevelSelector(): void MenuPage **Fields** back : ClickableText «final» backable: boolean linesOfText : ArrayList<LineOfText> page: Page «final» title: String yOrigin: float Constructors MenuPage( String, boolean ) : void Methods <u>getInstructionsMenu( ) : MenuPage</u> <u>getLevelSelectorMenu( ) : MenuPage</u> <u>· getMainMenu( ) : MenuPage</u> <u>· getPauseMenu( ) : MenuPage</u> <u> getSettingsMenu( ) : MenuPage</u> populateClickables( DontDrown ): void render( DontDrown ) : void resolveClick( DontDrown, GameMenu ) : void ClickableText **Fields** «final» alignment : int height: float pos : PVector «final» text: String width : float Constructors · ClickableText( String, int ): void Methods · hover( DontDrown ) : boolean render( DontDrown ): void Note **Fields** «final» · durationMult : int «final» pitchName : String Constructors Note( String ) : void - Note( String, int ): void

# Sketcher Constants RSS MAX : int RSS\_MIN: int - RSV\_MAX : float RSV\_MIN: float RSW DEF DIV: float WAVE : int Fields RSW DEF: float roughStrokeShakiness : int - roughStrokeVariabilityRate : float roughStrokeWeight : float Constructors - Sketcher(): void Methods drawDualWeightedLine( PVector, PVector, float, float ): PShape handDraw(int, float[]): PShape «transient» · handDraw( int, int, int, float[] ) : PShape handDrawLine( PVector, PVector ): PShape handDrawLine( int, PVector, PVector ) : PShape sinWave( float, int, float, int, int ) : PVector[]

**PApplet** 

# CollisionDetector Fields + pcOldPos: PVector «final» - sketch: DontDrown + sortedPlatforms: ArrayList<Platform> + sortedTokens: ArrayList<Token> Constructors + CollisionDetector( DontDrown ): void Methods + detectCollisions(): void - detectPlatformCollisions( PlayerCharacter, PVector ): void - detectTokenCollisions( PlayerCharacter, PVector ): void

getXAtYOverlap( PlayerCharacter, PVector, float ) : float

- sortLists(): void

· playMusic( ) : void

Constants

DEBUG\_TEXT\_DIV : float

Fields

«final»
sketch : DontDrown
«final»
textSize : float

Constructors

+ DebugOverlay( DontDrown ) : void

Methods

+ render( ) : void

MusicPlayer

# MusicPlayer Constants BPM: float MSPB: float notes : Note[] Fields «final» minim: Minim nextPlayMillis: long noteIndex:int «final» output : AudioOutput «final» sketch : DontDrown Constructors · MusicPlayer( DontDrown ) : void Methods

## DontDrown Constants BULLET\_POINT : char EXTENSION\_TIME\_MULT: int + FONT\_PATH : String LOADING\_TEXT\_DIV : float REPERCUSSION\_TIME\_MULT : int SCROLL\_DIV : int Fields arcadeMode : boolean collisionDetector : CollisionDetector - debugOverlay : DebugOverlay · debugging : boolean endOfExtension: int endOfRepercussion: int extensionFrames: int extensionUsed : boolean gameMenu : GameMenu gameState : GameState level : Level - levelStartTimeMillis : long levelState : StressAndTokenState · levels : Level[][] - musicPlayer : MusicPlayer pc : PlayerCharacter F playingMusic : boolean repercussionFrames: int repercussionMult : float risingWave : Wave scoreOverlay : ScoreOverlay scrollIncr : int - staticStress : boolean staticWave : Wave Constructors · DontDrown(): void Methods colorModeHSB(): void

Methods

+ colorModeHSB(): void
+ colorModeRGB(): void
+ draw(): void
+ endLevel( boolean ): void
+ generateLevels(): void
- integrateWave(): void
+ keyPressed(): void
+ keyReleased(): void
+ main( String[]): void
+ mouseClicked(): void
+ mouseWheel( MouseEvent ): void
+ settings(): void
+ startLevel( Level ): void

Page Constants LINE\_COLOUR: int MARGIN\_COLOUR : int - MARGIN\_DIV: int PAGE\_COLOUR : int Fields «final» height: int «final» · lineGap : float - lines : PShape ⊦ marginX : float «final» sketch : DontDrown «final» startAtTop : boolean «final» + topLineY : float Constructors Page( DontDrown ): void Page( DontDrown, int, boolean ) : void Methods drawLine( PVector, PVector, float, int ): PShape generateLines(): void render(): void setMargin( DontDrown ) : void

«final»	
+ endOfPadding : float	ľ
«final»	
- levelInfoFont : PFont «final»	l 1
- scoreFont : PFont	ľ
«final»	ľ
- sketch : DontDrown	
«final»	
- stressBar : StressBar	
Constructors	
+ ScoreOverlay( DontDrown ) : void	
Methods	
+ render( ) : void	
	'   <sub>-</sub>
	ľ
	l ľ
StressBar	
Constants	
OTDEOG DAD HEIGHT DIV. (I	ľ
- STRESS_BAR_HEIGHT_DIV : float + STRESS_BAR_RESOLUTION : int	
- STRESS_BAR_WIDTH_DIV : float	
- STRESS_BAR_WIDTH_DIV : IIOat	
Fields	
«final»	ľ
+ fill : StressBarFill	
- height : float	
«final»	
+ outer : StressBarOuter	
«final»	
+ pos : PVector	

Constructors

Methods

ScoreOverlay

Constants

Fields

LEVEL\_INFO\_TEXT\_DIV : float

SCORE\_TEXT\_DIV: float

- bigToken : BigToken

«final»

- width : float

render(): void

StressBar( DontDrown ) : void

## Constants + ABS\_MAX\_STRESS : int + DEFAULT\_STRESS\_EFFECT\_THRESHOLD : int + FRAMES\_PER\_RESKETCH\_MAX : int + FRAMES\_PER\_RESKETCH\_MIN : int + FRAMES\_PER\_RESKETCH\_RANGE : int + STRESS\_DECR\_RATE : float + STRESS\_INCR\_RANGE\_DIV : float + STRESS\_INCR\_RATE : float Fields + debuff : Debuff + framesPerResketch : int «final» - framesPerResketchMultiplier : float + maxStress : int + minStress : int + oldStress : float + pcFriction : float + pcFrictionMultiplier : float + pcMinSpeed : float + pcThrust : float + pcThrustMultiplier : float «final» - sketch : DontDrown + stress : float + stressEffectThreshold : int + stressHSBColour : float[] «final» - stressHueMultiplier : float + stressIncrRange : float «final» - stressLightMultiplier : float - stressRange : float + stressRating : float «final» - stressSatMultiplier : float «final» - strokeShakinessMultiplier : float «final» - strokeVariabilityMultiplier : float + tokensAvailable : int + tokensCollected : int + waveLastSeen : int **Properties** «readOnly» noteDuration : float Constructors - StressAndTokenState( DontDrown ) : void Methods - calcStressHSBColour( ) : void + collectToken( Token ) : void pcCalcs(): void pcFriction(): void pcMinSpeed(): void pcThrust(): void reset(): void + reset( Level ) : void + sketchiness( ) : void + update( ) : void

- updateStress( ) : void

StressAndTokenState