

T
DrawableObject
+ destroy() :boolean + drawConfigurationPanel() :void + drawObject() :void + getObject() :T + getPane() :Node + onCreate() :void + onStart() :void + setObject(T) :void

CGTGameOppositeDrawable
- oppositeTitledPane :OppositeTitledPane
+ CGTGameOppositeDrawable(Pane, Pane) + getPane() :Node + onCreate() :void + onStart() :void + toString() :String

CGTGameEnemyDrawable
- enemyTitledPane :EnemyTitledPane
+ CGTGameEnemyDrawable(Pane, Pane) + CGTGameEnemyDrawable(CGTEnergy, String, Pane, Pane) + getPane() :Node + onCreate() :void + onStart() :void + toString() :String

CGTGameActorDrawable	
-	actorTitledPane :ActorTitledPane
+	CGTGameActorDrawable(Pane, Pane)
+	CGTGameActorDrawable(CGTActor, String, Pane, Pane)
+	getPane() :Node
+	onCreate() :void
+	onStart() :void
+	toString() :String

CGTGameProjectileDrawable	
-	pane :ProjectileTitledPane
+	CGTGameProjectileDrawable(Pane, Pane)
+	CGTGameProjectileDrawable(CGTProjectile, String, Pane, Pane)
+	getPane() :Node
+	onCreate() :void
+	onStart() :void
+	toString() :String

CGTHUDDrawable	
-	lives :ArrayList<CGTLifeBarDrawable>
-	name :String
+	CGTHUDDrawable(Pane, Pane, String)
+	destroy() :boolean
+	drawConfigurationPanel() :void
+	drawObject() :void
+	getLives() :ArrayList<CGTLifeBarDrawable>
+	getName() :String
+	getPane() :Node
+	onCreate() :void
+	onStart() :void
+	toString() :String

CGTLifeBarDrawable
<ul style="list-style-type: none"> - life :IndividualLifeBar - lifePane :ConfigLifePane - preview :Draggable = new Draggable()
<ul style="list-style-type: none"> + CGTLifeBarDrawable(Pane, Pane) + destroy() :boolean + drawConfigurationPanel() :void + drawObject() :void + getDraggable() :Draggable + getLife() :IndividualLifeBar + getPane() :Node + onCreate() :void + onStart() :void + setSizeLife() :void + toString() :String

	T
AbstractDrawableObject	
<ul style="list-style-type: none"> - drawableConfigurationsPane :Pane - drawableObjectPane :Pane - object :T 	
<ul style="list-style-type: none"> + AbstractDrawableObject(Pane, Pane) + AbstractDrawableObject(T, Pane, Pane) + getDrawableConfigurationsPane() :Pane + getDrawableObjectPane() :Pane + getObject() :T + setObject(T) :void + updateConfigPane(Pane) :void + updateConfigPane(Node) :void + updateDrawPane(Node) :void + updateDrawPaneClear(Node) :void 	

CGTEnemyGroupLifeBarDrawable	
-	life :EnemyGroupLifeBar lifePane :ConfigGroupLifePane preview :Draggable = new Draggable()
+	CGTEnemyGroupLifeBarDrawable(Pane, Pane) destroy() :boolean drawConfigurationPanel() :void drawObject() :void getDraggable() :Draggable getLife() :EnemyGroupLifeBar getPane() :Node onCreate() :void onStart() :void setSizeLife() :void toString() :String

CGTGameScreenDrawable	
-	screenPane :ConfigScreenPreviewPane
+	CGTGameScreenDrawable(Pane, Pane) CGTGameScreenDrawable(Pane, Pane, int, int) CGTGameScreenDrawable(CGTScreen, Pane, Pane, int, int) CGTGameScreenDrawable(CGTScreen, Pane, Pane) destroy() :boolean drawConfigurationPanel() :void drawObject() :void getPane() :Node onCreate() :void onStart() :void toString() :String

CGTButtonScreenPreview	
-	buttonPane :ConfigButtonPreviewPane name :String preview :Draggable = new Draggable() screenName :String
+	CGTButtonScreenPreview(Pane, Pane) CGTButtonScreenPreview(CGTButtonScreen, String, Pane, Pane) destroy() :boolean drawConfigurationPanel() :void drawObject() :void getImage() :Draggable getPane() :Node getScreenName() :String onCreate() :void onStart() :void setSizeButton() :void toString() :String

		<i>T > CGTGameObject</i>
CGTGameObjectDrawable		
-	bounds :Rectangle collision :Rectangle gameObjectTitledPane :GameObjectPane preview :Draggable = new Draggable() worldName :String	
+	CGTGameObjectDrawable(Pane, Pane) CGTGameObjectDrawable(T, String, Pane, Pane) destroy() :boolean drawConfigurationPanel() :void drawObject() :void getDraggable() :Draggable getObjectPane() :GameObjectPane getPane() :Node getWorldName() :String onStart() :void setSizeObject() :void setWorldName(String) :void showGameObjectDialog() :Optional<Pair<String, String>>	

	<i>BorderPane</i>
PreviewPane	
<ul style="list-style-type: none"> + <u>DESKTOP_JAR_PATH</u> :String = "desktop/desкто... {readOnly} + <u>DESKTOP_ZIP_PATH</u> :String = "desktop/desкто... {readOnly} + drawableConfigurationsPane :Pane + drawableObjectPane :Pane - openRecentMenu :Menu - rootItem :CGTProjectDrawable - running :boolean - tree :TreeView<DrawableObject> 	
<ul style="list-style-type: none"> + about() :void + addBonus() :void + addButtonScreen() :void + addEnemy() :void + addEnemyLifeBar() :void + addGearInformation() :void + addObjectLifeBar() :void + addOpponent() :void + addProjectile() :void + addScreen() :void + addSpriteSheet() :void + addWorld() :void + beforeClosing() :void + closeProject() :void - copyDesktopFiles() :void + editSpriteSheet() :void + exit() :void + exportProject() :void - getActorWorldNode(String) :TreeItem<DrawableObject> - getHUDNode(String) :TreeItem<DrawableObject> - getScreenNode(String) :TreeItem<DrawableObject> - getWorldNode(String) :TreeItem<DrawableObject> - isWin() :boolean - localDefaultDirectory() :String + newProject() :void - open(File) :void + openProject() :void + PreviewPane() - runDesktop() :void + runProject() :void + saveProject() :void + saveProjectAs() :void - setupTree() :void - showValidateDialog(List<CGTError>) :void - updateRecent() :void - updateTree() :void 	