

Renderer

- + virtual ~Renderer() + GamePixmapItem * renderGame
- + GamePixmapitem * renderGame Objects(QList< QMap< DataRole,
- QVariant >> objectData) + virtual GamePixmapItem
 - * renderGameObject(QMap
 - < DataRole, QVariant > objectData)
 - + virtual void renderGameObject
 - (QMap < DataRole, QVariant > objectData, GamePixmapItem *item)
- + QPixmap rotatePixmap
- (const QPixmap &originalPixmap, int direction)+ QImage rotateImage
- (const Qlmage &image, int direction)
- # QImage animateHealthPack
- (int health, GamePixmapItem *item) # QPropertyAnimation
- * animateTint(QColor
- final, QColor initial ={0, 0, 0, 0})
- # QPropertyAnimation
 * animateAttack(int
- * animateAttack(int dir, bool attacking)
- # QPropertyAnimation
 * animateBounce()
 - # QPropertyAnimation
 - * animateHealth(Direction dir)
 - # QPropertyAnimation
 * animateHide()

ColorRenderer

Δ

- + ColorRenderer()
- + void renderGameObject(QMap < DataRole, QVariant> objectData, GamePixmapItem
- *item) override
- QPixmap renderTile (QMap < DataRole, QVariant
- object)- QPixmap renderDoorway(OMan< DataBole OVariant
- (QMap< DataRole, QVariant > object)
- QPixmap renderHealthPack (QMap< DataRole, QVariant
- object)- QPixmap renderProtagonist(QMap< DataRole, QVariant
 - > object)
- QPixmap renderEnemy (QMap< DataRole, QVariant > object)
- QPixmap renderPEnemy (QMap< DataRole, QVariant
 - > object)
 OPixman renderMovingEnem
- QPixmap renderMovingEnemy (QMap< DataRole, QVariant > object)