static constexpr int CELL SIZE + virtual ~Renderer() + GamePixmapItem * renderGame Objects(QList < QMap < DataRole, OVariant >> objectData) + virtual GamePixmapItem * renderGameObject(QMap < DataRole, QVariant > objectData) + virtual void renderGameObject (QMap < DataRole, QVariant > objectData, GamePixmapItem *item) + QPixmap rotatePixmap (const QPixmap &originalPixmap, int direction) + Qlmage rotatelmage (const Qlmage &image, int direction) # QImage animateHealthPack (int health, GamePixmapItem *item) # QPropertyAnimation * animateTint(QColor final, QColor initial $=\{0, 0, 0, 0\}$ # QPropertyAnimation * animateAttack(int dir, bool attacking) # QPropertyAnimation animateBounce() # QPropertyAnimation animateHealth(Direction dir) QPropertyAnimation animateHide() SpriteRenderer + static const QMap< ObjectType, CharacterData > m charMap - QImage m tiles - Qlmage m characters - QSize m_charSize - QSize m tileSize TextRenderer + SpriteRenderer() + void renderGameObject (QMap < DataRole, QVariant > data, GamePixmapItem + void renderGameObject **OverlayRenderer** *item) override (QMap < DataRole, QVariant + GamePixmapItem * renderGame > data, GamePixmapItem Object(QMap < DataRole, QVariant *item) override > data) override - QImage renderTile(QMap + OverlayRenderer() QImage sliceFrames < DataRole, QVariant > (QImage image, QLine data) diagonal, QPoint frameSize) - QImage renderCharacter - QRect getTileRect(QMap (QString str, QColor color, < DataRole, QVariant > int direction=0) data) QRect getCharacterRect (ObjectType type) - int calculateFrame (QVariant direction, int POVnum) - QPropertyAnimation

* animateDeath(QPoint

frame)

Renderer

ColorRenderer

- + ColorRenderer()
- + void renderGameObject (QMap< DataRole, QVariant
- > objectData, GamePixmapItem
- *item) override
- QPixmap renderTile (QMap< DataRole, QVariant
- object)- QPixmap renderDoorway(OMap< DataRole, OVariant
- > 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)
- QPixmap renderMovingEnemy (QMap< DataRole, QVariant > object)