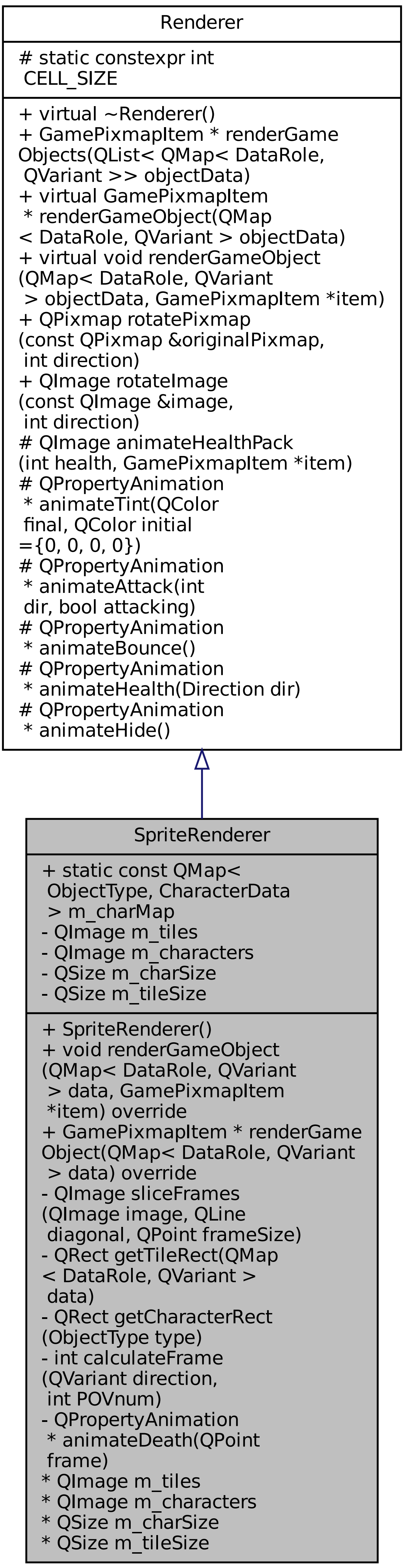
Renderer
# static constexpr int CELL_SIZE
+ virtual ~Renderer()
+ GamePixmapItem * renderGameObjects(QList< QMap< DataRole, QVariant >> objectData)
+ virtual GamePixmapItem * renderGameObject(QMap< DataRole, QVariant > objectData)
+ virtual void renderGameObject(QMap< DataRole, QVariant > objectData, GamePixmapItem *item)
+ QPixmap rotatePixmap(const QPixmap &originalPixmap, int direction)
+ QImage rotateImage(const QImage &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
- QImage m_characters
- QSize m_charSize
- QSize m_tileSize
+ SpriteRenderer()
+ void renderGameObject(QMap< DataRole, QVariant > data, GamePixmapItem *item) override
+ GamePixmapItem * renderGameObject(QMap< DataRole, QVariant > data) override
- QImage sliceFrames(QImage image, QLine diagonal, QPoint frameSize)
- QRect getTileRect(QMap< DataRole, QVariant > data)
- QRect getCharacterRect(ObjectType type)
- int calculateFrame(QVariant direction, int POVnum)
- QPropertyAnimation * animateDeath(QPoint frame)
* QImage m_tiles
* QImage m_characters
* QSize m_charSize
* QSize m_tileSize