Player + Player(sf::RenderWindow *window, sf::View *view, std::unordered_map< std ::string, sf::Texture > *textures, int id, sf::Color color) + void draw() + void update() + void addCard(Card card) + const std::vector< Card > & getCards() const + void drawCardUI() + bool canBuildRoad() const + void buildRoad() + bool canBuildSettlement () const + void buildSettlement() void takeResources (ResourceType resourceType, int amount) + const std::map< Resource Type, int > & getResources () const + int getVictoryPoints () const + bool hasWon() const + void setLargestArmy (bool has) + bool hasLargestArmyCard () const + bool canBuyDevelopmentCard () const + void buyDevelopmentCard() + int getID() const + sf::Color getColor () const + bool useKnightCard() + bool useYearOfPlentyCard (ResourceType first, ResourceType second) + bool useRoadBuildingCard() + bool useMonopolyCard (ResourceType resource) + bool useVictoryPointCard() + bool hasCard(Card cardType) + void removeCard(Card cardType) + Card handleCardClick (sf::Vector2f clickPosition) + void addResource(Resource Type type, int amount=1) + void removeResource (ResourceType type, int amount=1) + int getResourceCount (ResourceType type) const + void incrementKnightsPlayed() + int getKnightsPlayed () const + bool hasResources() const int getTotalResources () const + ResourceType getRandomResource () const + void discardResources (int count) + void setMustMoveRobber (bool must) + bool mustMoveRobber () const + void setLongestRoad (bool has) + bool hasLongestRoadCard () const + void incrementRoadCount() + int getRoadCount() const void clearNewlyPurchased Cards() + bool isCardPlayable (Card card) const #owner Building

sf::Color color

+ Building(Player *owner, sf::Color color)

+ Player * getOwner() const

const
+ sf::Color getColor
() const

Settlement

+ Settlement(Player

*owner, sf::Color color)
+ virtual void addResource
(ResourceType type)