## Catan + Catan() + ~Catan() + void run() + void resetClickPosition() + void setMenu(bool menu) + bool isMenu() const + void nextTurn() + void buySettlement() + void buyRoad() + bool canBuildRoad() const + bool canBuildSettlement () const + int getPlayerCount () const + void incrementPlayerCount() + void decrementPlayerCount() + void rollDice() + void giveRandomCard() + void handleCardUse (Card card) + void initiateTrade (int target) + bool proposeTrade(int targetPlayer, const std::map< ResourceType, int > &offer, const std ::map< ResourceType, int > &request) + void acceptTrade() + void declineTrade() + Player \* getCurrentPlayer() + Player \* getPlayer (int playerIndex) int getCurrentPlayerIndex () const + void setTrading(bool trading) + bool getTrading() const + bool bankTrade(ResourceType giveResource, ResourceType getResource, int giveAmount=4) + bool isBankTrading () const + void setBankTrading (bool trading) + void restartBoard() #game Menu # sf::RenderWindow \* window # sf::View \* view # sf::Font font # float xPart # float yPart # sf::Vector2f mousePosition # const float BUTTON PADDING # const float TITLE\_SCALE # const float BUTTON SCALE # const unsigned int TITLE\_SIZE # const unsigned int REGULAR\_BUTTON\_SIZE # const unsigned int SMALL BUTTON SIZE

## # const most borron \_SCALE # const unsigned int \_TITLE\_SIZE # const unsigned int \_REGULAR\_BUTTON\_SIZE # const unsigned int \_SMALL\_BUTTON\_SIZE # const sf::Color NORMAL \_COLOR # const sf::Color HOVER \_COLOR # const sf::Color ACTIVE \_COLOR # const sf::Color INACTIVE \_COLOR + Menu(sf::RenderWindow \_\*window, sf::View \*view, Catan \*game)

- + virtual ~Menu()=default + virtual void draw()=0
- + virtual void update
  - (sf::Vector2f mousePosition)=0
- # void updateMousePosition()
- # sf::Vector2f getCenterPosition () const
- # float getXPos(float
- multiplier) const
  # float getYPos(float
  multiplier) const



## StartMenu(sf::RenderWindow \*window, sf::View \*view, Catan \*game)

- + void draw() override
- + void update(sf::Vector2f mousePosition) override