

Observer,
Factory

ProcessInputType: Enum ProgState: Enum + (virtual) render(): void + (virtual) update(float deltaTime); void ButtonShape: Enum using XYPair = std::pair<unsigned, unsigned> Prog StateManager GUIBullder m instance: std::unique_ptr<Prog> m state:std::unique ptr<State> addButton/ GUI const & qui. ButtonShape -m_stateManager: std::unique_ptr<StateManager> shape, Color color, XYPair size, XYPair displayCenterPointAddres); void + StateManager() - Prog() + addTextBox(GUI const & qui. Color color. XYPair size, XYPair displayCenterPointAddres): + update(float deltaTime); void -Prog() + render(): void + getInstance(): std::unique_ptr<Prog> + addKeyboard(GUI const & gui, Color color. processInput(): void XYPair size, XYPair displayCenterPointAddres): + update(float deltaTime): void + render(): void + processInput(): void GUI TextBox m_buttons: std::vector<Button> m color: Color <<abstract>> State m textboxes; std::vector<TextBox> m_size: XYPair m_stateManager: std::unique_ptr<StateManager> - m_backgroundColor: Color m_displayAddress: XYPair -m_gui: std::unique_ptr<GUI> m text: string + processInput(): void -m_gui: std::unique_ptr<GUlBuilder> + render(): void + render(): void + (virtual) update(float deltaTime): void + processInput(); void + (virtual) render(); void + (virtual) process(nput(): void Button m_shape: Shape m color: Color MainMenu State NewDeviceControlling SignalMenu State m_displayAddress: XYPair + MainMenu() + NewDeviceControllingSignalMenuState() m size: XYPair +-MainMenu() + -NewDeviceControllingSignalMenuState() + processInput(); void + update(float deltaTime): void + update(float deltaTime): void + render(): void + render(): void + render(): void + processInput(): void + processInput(): void ChangeRoomNameMenu State Room Settings Menu State RoomChooseMenu State + ChangeRoomNameMenuState() + RoomSettingsMenuState() + RoomChooseMenuState() + -ChangeRoomNameMenuState() + -RoomSettingsMenuState() + -RoomChooseMenuState() + update(float deltaTime): void + update(float deltaTime): void + update(float deltaTime): void + render(); void + render(): void + render(): void + processInput(): void + processInput(): void + processInput(): void

<<abctract>>

Renderable

Updafable

W builderze do

worzenia przycisków

treścią, sprawdzić przy implementacji

moglaby byč metoda wytwórcza lae tym razem to chyba przerost formy nad