Contains string tokens used to map an input string nto an internal data structure type -IThumb, rThumb, ITrigger, rTrigger: [std::string] holds strings "LTHUMB", "RTHUMB", "LTRIGGER", "RTRIGGER" respectively -IShoulder, rShoulder: [std::string] holds strings "LSHOULDER", "RSHOULDER" respectively -dpad, left, down, up, right, none: [std::string] holds strings "DPAD", "LEFT", "DOWN", "UP", "RIGHT", "NONE" respectively

struct PlayerInfo +PlayerInfo(): Constructor -left_dz : int, left thumbstick deadzone -right_dz: int, right thumbstick deadzone -left_trigger_dz: int, left trigger deadzone -right_trigger_dz: int, right trigger deadzone -player_id: int, unused player ID placeholder -x_mouse_pps: unsigned, x mouse pixels per second

-y_mouse_pps: unsigned, y mouse pixels per second

MultiBool()

<<Enumeration>>

STATE_ONE

STATE_THREE

STATE_FOUR STATE_FIVE STATE_SIX

STATE_TWO

BUTTONSTATE (nested in

MultiBool)

MultiBool(const MultiBool &rhs)

current_state : BUTTONSTATE

GamepadUser ~GamepadUser() GamepadUser() mapper: Mapper* mouse: XInputBoostMouse* poller : InputPoller* transl: XInputTranslater* InputPoller : public XInputBoostAsync XInputTranslater InputPoller() (+1 overload) IsControllerConnected(): bool ~XInputTranslater() ProcessState(): ActionDetails IsRunning(): bool ProcessState(): void XInputTranslater() Start(): bool btn: ButtonAction* Stop(): bool player: PlayerInfo* thmb: ThumbstickAction* trg: TriggerAction* ButtonAction Mapper Down(): bool GetMapInfo(): MapInformation ProcessActionDetails(): void humbstickAction SetMapInfo(): bool Down(): bool XInputBoostMouse : public XInputBoostAsync riggerAction WordData (nested in Mapper) ~XinputBoostMouse() Down(): bool EnableProcessing(): void WordData() GetSensitivity(): int ProcessState(): void control : string SetSensitivity(): void down: bool XInputBoostMouse() fsm: MultiBool info: string <<Enumeration>> stickMapInfo : volatile MouseMap sim_type: string MouseMap (nested in value : string XInputBoostMouse) NEITHER_STICK RIGHT_STICK LEFT_STICK XInputBoostAsync ~XInputBoostAsync() ProcessState(): void

workThread(): void

XInputBoostAsync()

struct ActionDescriptors

-x, y, a, b : [std::string] holds strings "X","Y","A","B" respectively

-start, back, vk : [std::string] holds strings "START", "BACK", "VK" respectively

-mappedTo, norm, toggle, rapid: [std::string] holds strings "=", "NORM", "TOGGLE", "RAPID" respectively

-moreInfo, delimiter: [char] holds characters ':' and ' ' (space) respectively