Attributes:

* string type;
* string buttonLayout;
* int sensitivity;

**Class: VRController**

Methods:

* void connectToHeadset (VirtualRealityHeadset& headset);
* void sendInput();
* void chargeController();

Methods:

* void turnOn();
* void turnOff();
* void connectToGame(VRGame& game);

Attributes:

* string model;
* string compatibility;
* int batteryLife;

**Class: VirtualRealityHeadset**

Attributes:

* string title;
* string genre;
* int difficultyLevel;

**Class: VRGame**

Methods:

* void startGame();
* void endGame();
* void pauseGame();

**Class: Actor**

Attributes:

* String name;
* String type;
* Int health;

Methods:

* void updatePosition(float x, float y, float z)
* void takeDamage(int damageAmount)

Methods:

* void wearHeadset(VirtualRealityHeadset& headset);
* void playGame(VRGame& game);
* void reactToGame();
* void useController(VRController& controller)

**Class: Sim**

Attributes:

* string currentMood;
* int energyLevel;
* vector<string> skillSet;
* VRInventory\* inventory;

Methods:

* void unlockAchievement (const string& achievementName);
* void trackProgress(double progressIncrement);
* vector<string> listAchievements();

**Class: VRAchievementTracker**

Attributes:

* Vector<string> achievements;
* Int progress;

**Class: VRChatSystem**

Attributes:

* Vector<string> chatHistory;

Methods:

* void sendMessage(const string& message);
* string receiveMessage();
* void OpenChatWindow();

**Class: Bullet**

Attributes:

* Int damage;
* Int speed;

Methods:

* void interact() override;
* Int getDamage () const;
* Int getSpeed() const;

**Class: Inventory**

Attributes:

* vector<VRItem>inventoryItems;
* Int capacity;

Methods:

* bool addItem(const VRItem& item);
* bool removeItem(const VRItem& item);
* int getCapacity() const;
* vector<VRItem> getInventoryItems() const;

**Class: Item**

Attributes:

* String itemName;
* String itemType;
* String virtualDescription;

Methods:

* void interact();
* void collect();