HolodeckVR Unity SDK

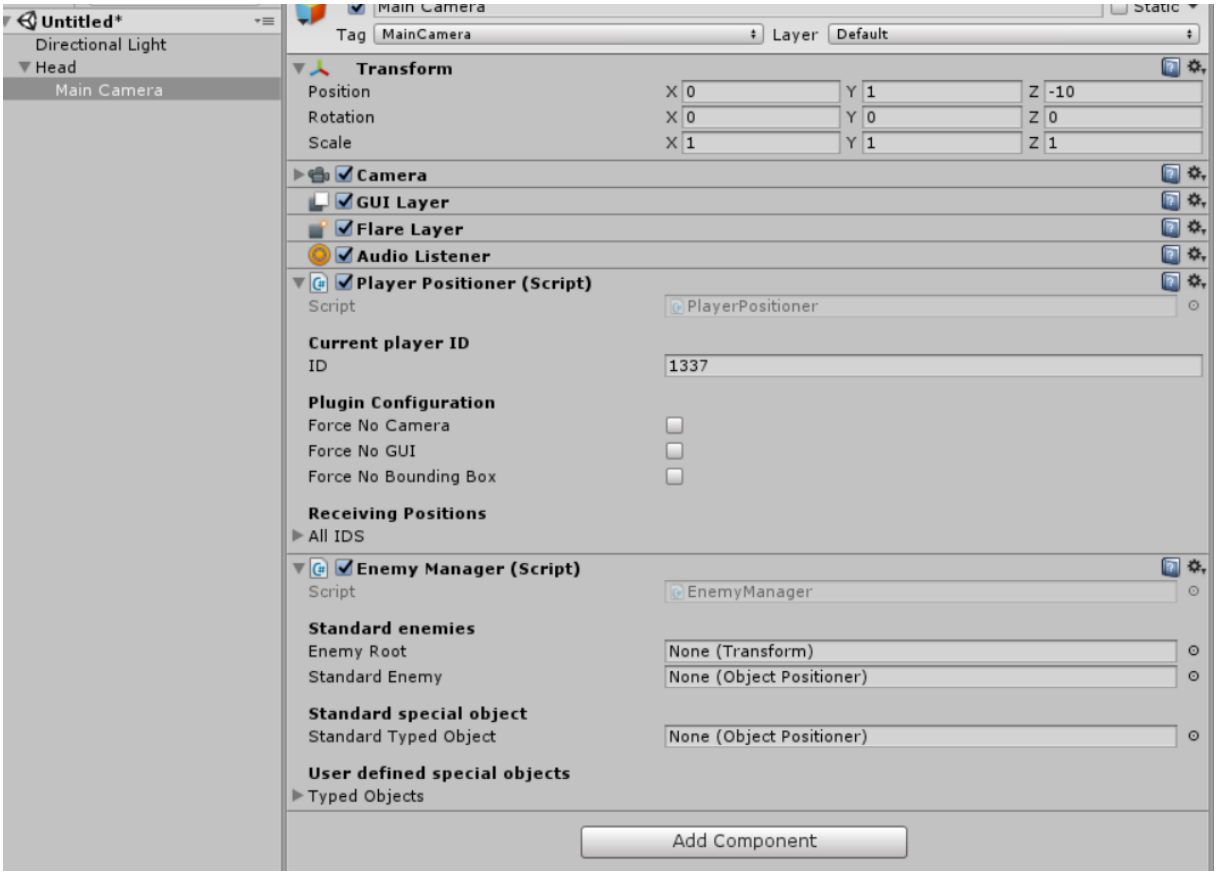
The HolodeckVR SDK provides access to the HolodeckVR tracking and managing solution. In the following are some Descriptions how to setup a basic project with the given Scripts as well the provided API to integrate own solutions.

The current documentation is a first draft and might misses some explanation.

Setting up an Basic Scene

1. Add an Empty GameObject to the scene and name it “Head”
2. Make the Camera a child Object of the Head and Tag it with “MainCamera”
3. Add the Script “PlayerPositioner” to the camera
4. Add the Script “EnemyManager” to the camera

The scene should now look like this



Q: Why do we need the “Head”?

A: The Head is necessary for calibration (yaw correction) and gets the current position. The camera itself will be manipulated by the used Head mounted display and cannot be used.

Q: Why should the Camera be tagged as “MainCamera”?

A: We need constant access to the current Orientation of the Camera without tagging Holodeck startup will not work properly.

Q: What does the “PlayerPositioner” Script?

A: The “PlayerPositioner” Script is the basic script for the Player. The tasks are writing the Position to the Head, tracking current available ID’s and checking the received “PlayerID”.

Q: What does the “EnemyManager” Script?

A: HolodeckVR is meant to be played by multiple Users in the same Space. To Prevent Collisions between users it’s very important to see each other. The “EnemyManager” automatically generates an enemy if another user is in the Tracking Space.

Q: Can I replace the Standard Enemy?

A: Yes, of course. The Standard Enemy is Saved in Holodeck->Plugin->Resources. The Only needed Script is the “ObjectPositioner” Script, the you can replace the Enemy within the Enemy Manager

API Description

CalibrationAPI

Contains functions used for calibration

Function	**Description**
event CalibrationProgress CalibrationProgress	This event is called (repeatedly) during calibration to give information about the calibration progress. Bind to this evet.
event CalibrationCompleted CalibrationCompleted	This event is called when player calibration was completed, either successful or unsuccessful.
void ManualCalibration()	When Manual Calibration is available (last option).
Quaternion GetCalibrationOffsetAsQuaternion()	Getting the current Offset as Quaternion
float GetCalibrationOffsetAsFloat()	Getting the current Offset as float

Debug API

Only used for Debugging Holodeck Functions without Holodeck available. Don't use it in Release.

Function	Description
void SimulateID(int id, Vector3 position, Quaternion rotation)	Simulates a value sent by the position system
void SimulateID(int id, Vector3 position)	Simulates a value sent by the position system
void SimulateConfig(ConfigData config)	Simulate a system configuration

GeneralAPI

Contains functions to setup Holodeck an get common information.

Function	Description
event HoloBackendReady HolodeckIsReady	Event gets calls if Holodeck is ready
event HolodeckStateChange HolodeckStateChanged	Event gets calls if Holodeck Status Changed
HolodeckState GetState()	Get Current Stage of Holodeck
bool IsHolodeckReady()	true if Holodeck is ready
void Start()	Must be called at SceneLoad. Setup all the elements necessary in the Scene for the holodeck to work
void Stop()	Suspend internal Holodeck functions
Vector3 GetTrackingSpaceOrigin()	Return the current Zero Position of the tracking space
Quaternion GetTrackingSpaceOrientation()	Return the current Orientation of the Tracking space
bool GetOfflineMode()	Returns if the Holodeck system is currently in offline mode
void SetNoCamera(bool cameraSetter)	Set the bool noCamera to parameter value
bool GetNoCamera()	Get the current value of bool noCamera
void SetNoGUI(bool guiSetter)	Set the bool noGUI to parameter value
bool GetNoGUI()	Get the current value of bool noGUI
void SetNoBoundingBox(bool bbxSetter)	Set the bool noBBX to parameter value
bool GetNoBoundingBox()	Get the current value of bool noBBX

IDAPI

Contains Function regarding tracked Positions

Function	Description
Vector3 GetRawPosition()	Returns the raw position values of the local player.
Vector3 GetRawPosition(int ID)	Returns the raw position values of an enemy or object, specified by an ID. Enemies and objects are all tags in Holodeck except the local player.
Vector3 GetLocalPosition()	Returns the Unity local position of the local player.
Vector3 GetLocalPosition(int ID)	Returns the Unity local position of an enemy or object, specified by an ID. Enemies and objects are all tags in Holodeck except the local player.
Vector3 GetGlobalPosition()	Returns the Unity global position of the local player
Vector3 GetGlobalPosition(int ID)	Returns the Unity global position of an enemy or object, specified by an ID. Enemies and objects are all tags in Holodeck except the local player.
bool HasSignal()	Gives information about the current position server connection
bool IsAvailable(int id)	Returns true if the id is currently available
bool HasPosition(int id)	Returns true if the id is currently available
bool HasRotation(int id)	Returns true if the system can deliver an orientation for the given id
int GetPlayerId()	Returns the ID of the player that is associated with this device
List<int> GetIds()	Returns the currently available ids
Quaternion GetRotation(int id)	Returns the rotation for the given id
string GetTypeofID(int id)	Returns the type that is associated with the id (special tags)
List<int> GetIDsOfType(string type)	Returns a list of ids of the given type (special tags)
event IDChange IDEntered	This event is invoked when a new id is detected
event IDChange IDLeft	This event is invoked when an id is no longer sent
event IDChange IDTypeAdded	This event is invoked when a new sender id is detected
IDChange IDTypeRemoved	This event is invoked when a sender id is no longer sent
event IDChange IDTypeChanged	This event is invoked when a sender ids type has changed, probably due to changes made in the config server.
event IDTagHandler IDTagsReceived	This event is invoked when ID tags of the server are read

SafteyAPI

Get Informations about saftey

Function	Description
event SafetyInformationUpdatedHandler SafetyInformationUpdated	Gets called when The Safety information got an Update
bool HasInformation()	Returns true if the System is connected to the Server and the Config is loaded
Bounds GetTrackingArea()	Get bounds of the tracking area set on holodeck Server
bool PlayerIsInside()	Returns if the Player is in Safety Area or not
bool PlayerIsOutside()	Returns if the Player is in Safety Area or not
bool ShowCamera()	If Camera Is available, you can use this variable to check whether it should be switched on or not
bool ShowGrid()	Return a bool value whether you should show the grid