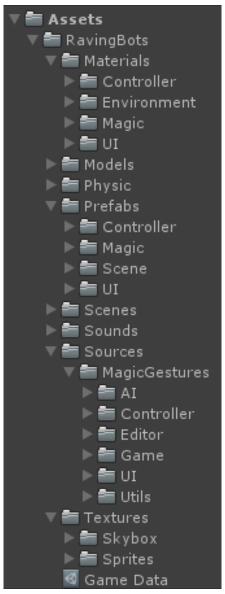
VR Magic Gestures AI

Version 1.2

CONTENT



```
▼ Assets
  ▼ RavingBots
    ▼ Materials
      ▶ Controller
      ▶ Environment
      ▶ Magic
      ▶ UI
    ▶ Models
    ▶ Physic
    ▼ Prefabs
      ▶ Controller
      ▶ Magic
      ▶ Scene
      ▶ UI
    ▶ Scenes
    ▶ Sounds
    ▼ Sources
      ▼ MagicGestures
        ▶ AI
        ▶ Controller
        ▶ Editor
        ▶ Game
        ▶ UI
        ▶ Utils
    ▼ Textures
      ▶ Skybox
      ▶ Sprites
      Game Data
```

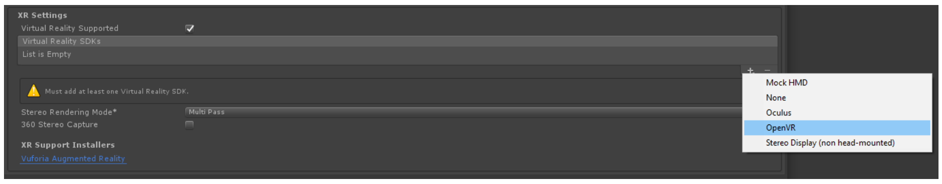
- Materials – different materials for all objects in example scene and prefabs, all use standard shader
- Models – simple object models based on capsules, cones and cylinders, castle mesh
- Physic assets – different physical parameters (bounciness) for prefabs
- Prefabs – UI and player prefabs, scene elements, premade gestures
- Scenes – Example scene with lightning data and reflection probes
- Sounds – Collection of funny and useful sounds (boinking, paffing, popping etc.)
- Sources – Source code divided into categories: AI, Controller, Editor, Game, UI, Utils
- Textures – Skybox and sprites

INSTRUCTIONS

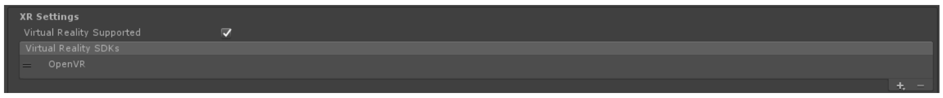
Quick start

1. Import asset to Ravingbots Folder
2. Set Up OpenVR as main VR SDK.

   Go to Edit->Project Settings->Player and Select OpenVR as Virtual Reality SDK



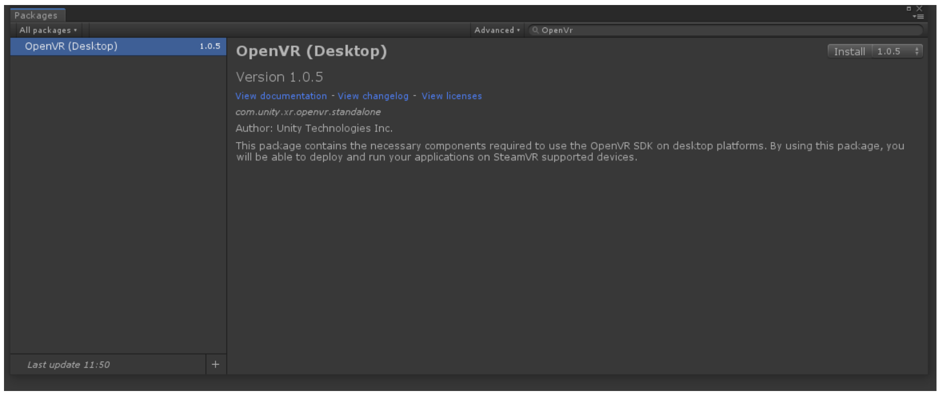
XR Settings
Virtual Reality Supported ✓
Virtual Reality SDKs
List is Empty
Must add at least one Virtual Reality SDK.
Stereo Rendering Mode* Multi Pass
360 Stereo Capture
XR Support Installers
Vuforia Augmented Reality



XR Settings
Virtual Reality Supported ✓
Virtual Reality SDKs
OpenVR

3. Add OpenVR package in Package Manager.

   Go to Window->Package Manager, search for OpenVR and click Install



OpenVR (Desktop) 1.0.5

OpenVR (Desktop)

Version 1.0.5

View documentation - View changelog - View licenses

com.unity.xr.openvr.standalone

Author: Unity Technologies Inc.

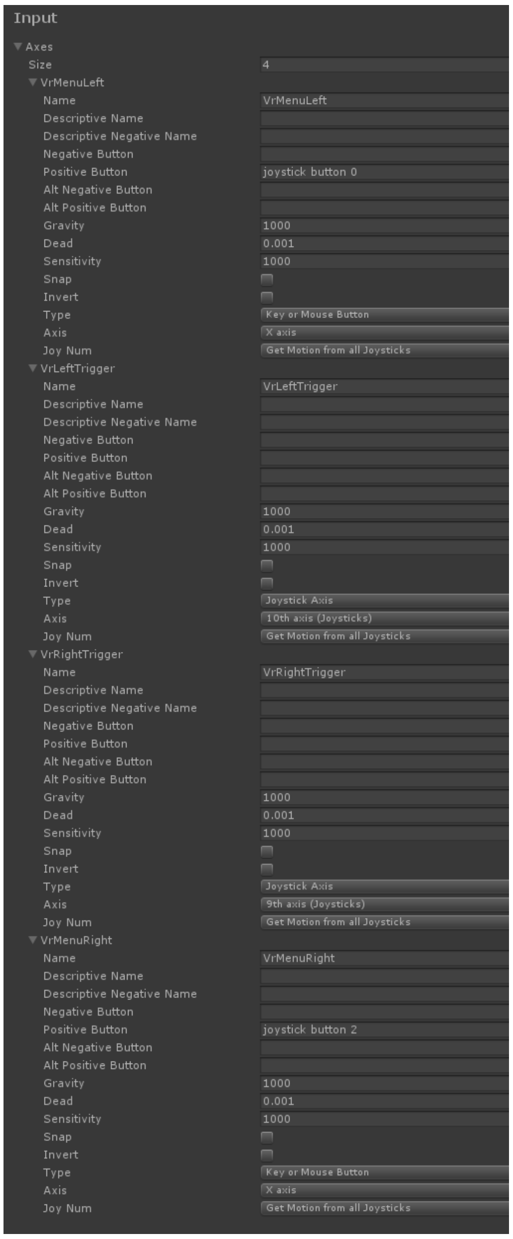
This package contains the necessary components required to use the OpenVR SDK on desktop platforms. By using this package, you will be able to deploy and run your applications on SteamVR supported devices.

4. Configure input by adding axes:
   4.1. VrMenuLeft
   - Positive Button: joystick button 0
   - Type: Key or Mouse Button
   - Axis: X axis
   - Joy Num: Get Motion from all Joysticks

   4.2. VrLeftTrigger
   - Type: Joystick axis
   - Axis: 10th axis (Joysticks)
   - Joy Num: Get Motion from all Joysticks
   - Sensitivity: 1

   4.3. VrMenuRight
   - Positive Button: joystick button 2
   - Type: Key or Mouse Button
   - Axis: X axis
   - Joy Num: Get Motion from all Joysticks

   4.4. VrRightTrigger
   - Type: Joystick axis
   - Axis: 9th axis (Joysticks)
   - Joy Num: Get Motion from all Joysticks
   - Sensitivity: 1

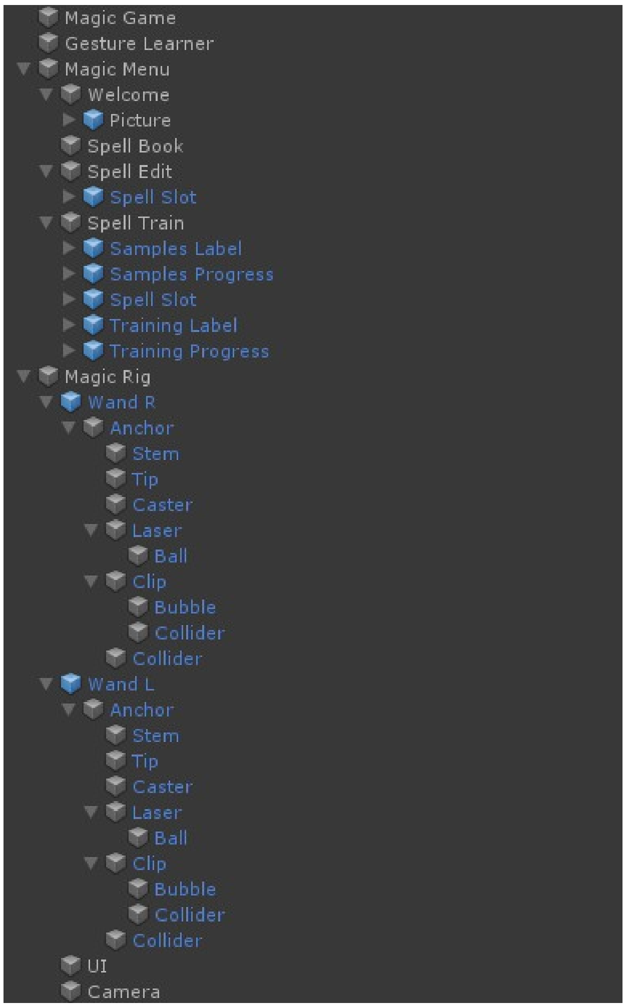


```
Input
▼ Axes
  Size                          4
  ▼ VrMenuLeft
    Name                        VrMenuLeft
    Descriptive Name
    Descriptive Negative Name
    Negative Button
    Positive Button             joystick button 0
    Alt Negative Button
    Alt Positive Button
    Gravity                     1000
    Dead                        0.001
    Sensitivity                 1000
    Snap
    Invert
    Type                        Key or Mouse Button
    Axis                        X axis
    Joy Num                     Get Motion from all Joysticks
  ▼ VrLeftTrigger
    Name                        VrLeftTrigger
    Descriptive Name
    Descriptive Negative Name
    Negative Button
    Positive Button
    Alt Negative Button
    Alt Positive Button
    Gravity                     1000
    Dead                        0.001
    Sensitivity                 1000
    Snap
    Invert
    Type                        Joystick Axis
    Axis                        10th axis (Joysticks)
    Joy Num                     Get Motion from all Joysticks
  ▼ VrRightTrigger
    Name                        VrRightTrigger
    Descriptive Name
    Descriptive Negative Name
    Negative Button
    Positive Button
    Alt Negative Button
    Alt Positive Button
    Gravity                     1000
    Dead                        0.001
    Sensitivity                 1000
    Snap
    Invert
    Type                        Joystick Axis
    Axis                        9th axis (Joysticks)
    Joy Num                     Get Motion from all Joysticks
  ▼ VrMenuRight
    Name                        VrMenuRight
    Descriptive Name
    Descriptive Negative Name
    Negative Button
    Positive Button             joystick button 2
    Alt Negative Button
    Alt Positive Button
    Gravity                     1000
    Dead                        0.001
    Sensitivity                 1000
    Snap
    Invert
    Type                        Key or Mouse Button
    Axis                        X axis
    Joy Num                     Get Motion from all Joysticks
```

All other axes as well as unmentioned fields may stay default/unchanged.

5. Open example scene
6. Profit

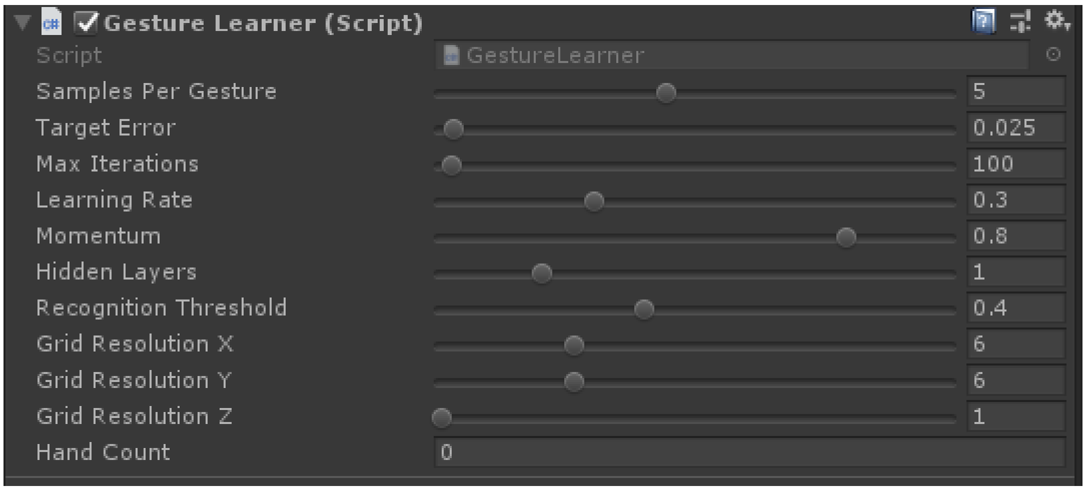
If you want to add Magic Gestures functionality to already created scene you need to include following GameObjects that you can copy from example scene or create yourself (be aware that shown GameObject names are not arbitrary, they match component names though to make things more transparent):



```
Magic Game
Gesture Learner
Magic Menu
  Welcome
    Picture
  Spell Book
  Spell Edit
    Spell Slot
  Spell Train
    Samples Label
    Samples Progress
    Spell Slot
    Training Label
    Training Progress
Magic Rig
  Wand R
    Anchor
      Stem
      Tip
      Caster
      Laser
        Ball
      Clip
        Bubble
        Collider
      Collider
  Wand L
    Anchor
      Stem
      Tip
      Caster
      Laser
        Ball
      Clip
        Bubble
        Collider
      Collider
  UI
  Camera
```

1. MagicGame with components: MagicGame, WandManager and GestureTracker
2. Gesture Learner with component: GestureLearner
3. Magic Menu with component Magic Menu and children: Welcome, Spell Book, Spell Edit, Spell Train (hierarchy shown in the screenshot)
4. Magic Rig or a player object with children: two instances of Wand prefab and two cameras

Tuning neural network


Gesture Learner (Script)

Script	GestureLearner
Samples Per Gesture	5
Target Error	0.025
Max Iterations	100
Learning Rate	0.3
Momentum	0.8
Hidden Layers	1
Recognition Threshold	0.4
Grid Resolution X	6
Grid Resolution Y	6
Grid Resolution Z	1
Hand Count	0

Samples per gesture: The number of gesture repeats used for training a single spell. Each repeated gesture is added to the training set. The neural network recognizes spells by generalizing patterns provided in the training set. The bigger the training set, the better tolerance and accuracy of recognition.

GridResolutionX
The width of the grid onto which the gesture is projected during the preprocessing phase.

GridResolutionY
The height of the grid onto which the gesture is projected during the preprocessing phase.

GridResolutionZ
The depth of the grid onto which the gesture is projected during the preprocessing phase.

HiddenLayers
The number of hidden layers in the neural network.

LearningRate
The learning rate is used by the back-propagation training algorithm. The parameter determines the speed of the training.

MaxIterations
The training is continued for the specified number of iterations unless the target error is reached earlier.

Momentum
The momentum parameter is used by the backpropagation training algorithm. The parameter accelerates the training process.

RecognitionThreshold

The threshold that must be crossed by one of the outputs of the neural network to identify a valid gesture.

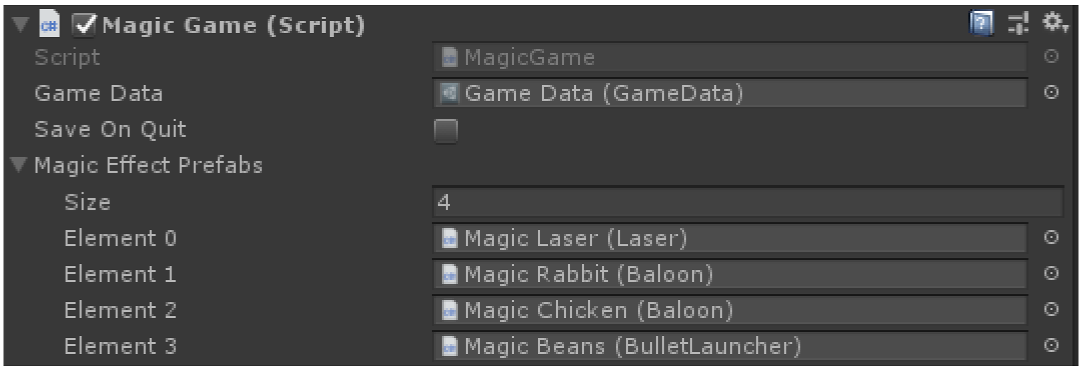
SamplesPerGesture

The number of gesture repeats used for training a single spell. Each repeated gesture is added to the training set. The neural network recognizes spells by generalizing patterns provided in the training set.

TargetError

The training process is stopped when the training error reaches the specified value.

Array of Magic Effect Prefab:



Magic Game (Script)

Script	MagicGame
Game Data	Game Data (GameData)
Save On Quit	☐
Magic Effect Prefabs	
Size	4
Element 0	Magic Laser (Laser)
Element 1	Magic Rabbit (Baloon)
Element 2	Magic Chicken (Baloon)
Element 3	Magic Beans (BulletLauncher)

Gestures are connected by index to elements of Magic Effect Prefabs array which contains prefabs with scripts that inherit from Magic Effect class (e.g. Laser, Baloon, BulletLauncher). You can add your own prefabs with changed meshes and personalized parameters or play around and create your own unique magic effect.

UPGRADE NOTES

Due to some structural changes inside the source code, especially WandManager script, it's advised to remove all old instances of prefabs used in scenes and copy prefabs from example scene.