State

TODO

1. Creates a force field
2. Death will attack enemy with a Scythe

Make it so you cannot teleport to your own platform nor the platform of the enemy

Make the haptic feedback work

Add the explotion particles in all spells, just like you did with the falling ball.

Create a mask for the vision when teleporting.

1/16/2017 Added the life Absolver Spell and fixed the transparency in the seals. I just had to change the shader to Sprite/default.

1/15/2017 Added the buff effect to the assets and made the script for it. Updated the steam vr plugin.

1/9/2017 Added the skull model with flames and proper explotion.

1/8/2017 The seal has a valid combination that creates a falling ball which explodes and shows a particle effect. Made the spell seal closer. The particle effects show in the point of contact. Added the referee. The falling ball gets the position of the enemy from the referee. Started working on the strength booster.

1/2/2016 The seal recognizes when the player selected a valid pattern and after all that validation destroys the spell seal.

1/1/2017 The spell Seal registers input only from the summoner’s hand and if it does not have a spell in it. Fixed a mistake in the teleportation indicator. The lazerPOinter script was looking for extra conditions to indicate that the lazer was leaving the platform. In fact, the lazer was not leaving. I just took out the last conditions and it worked fine. Now you just have to check if there was a previous contact and whether the hit transform is null. If that happens, means that the lazer left the platform. This is still not quite what I want because I need to check whether the platform is empty, but that is good enough for now.

12/30/2016 Modified the way the spell seal was created. Now I am using less expensive triggers and shapes. The symbols call the spell Seal script when the player touches them. I am using delegates and events for this.

12/29/2016 Modified the way the spell seals are going to work. Now they are independent from the character, they disappear after 8 seconds, they will have a list of states that will define what sequence of elements are valid.

12/26/2016 Started adding the spell seal. This seal will be used to cast different spells.

12/23/2016 Fixed the fireball feel. Added the shield. Refactored the spells. Now we do not need SpellInfos.

12/11/2016 The fireball is not going in the right direction. I adjusted the timing and speed.

12/10/2016 Modified the Expandible class. Now spells will be thrown by the spellcaster instead of moving by themselves. Fixed the state machine because you could not have spells in both hands. Added the fireball spell.

12/3/2016 Changed the teleportation to the right touchpad press. Added the state machine in Honovi. Now the state handles the input from the player.

11/26/2016 You have to create a particle effect for the platform that is the target of the teleportation. Make the particle effect disappear if the pointer leaves the platform. Make the seal active if you press the grip. Shows a trail of particles when teleporting. Throws spells that wait until you release the trigger to move.

11/25/2016 Added the seal to the left hand to teleport. The seal has the pointer invisible to know when it touches something.

11/14/2016 Created a new class for Honovi that inherits from the abstract class Character. The game is still working.

11/12/2016 I positioned the hands properly and rotated them.

11/11/2016 Added the JSON File for Honovi and the IOManager that will send it to the player class, which will pass it to the character.

11/7/2016 Added the character script and call him from player. The hands are uploaded at runtime.

11/5/2016 I started the implementation of the Character class. It currently has dummy data with Honovi info. You have to call the class from Player. And implement the reactions.

8/29/2016 The Prototype of the game is teleporting using Raycast. Removed the ability to jump to avoid motion sickness. Added the ability to throw balls.

References

http://flafla2.github.io/2016/05/17/viveteleport.html