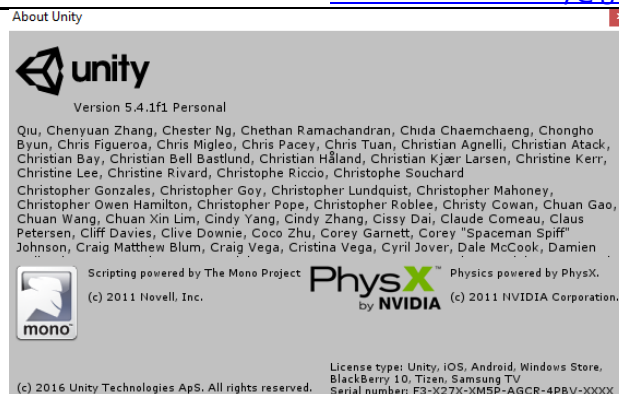


VR Chess Game Concept



1. GENERAL INFORMATION

DATE OF DOCUMENT	19/10/2016
NAME OF THE PROJECT	Vr Chess Game Concept
AUTHOR	Michael Soler
UNITY VERSION	5.4.1.F1 PERSONAL
CONTACT	michael.soler.beatty@gmail.com



Includes video tutorial for script info!



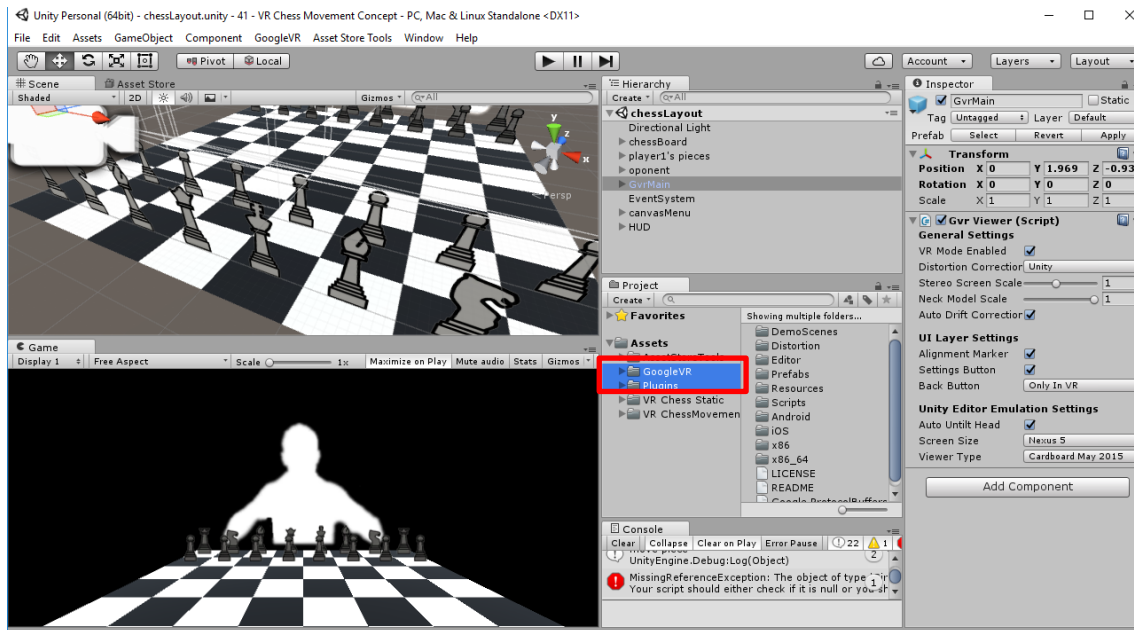
Index

1. GENERAL INFORMATION.....	1
2. IMPORTING INFORMATION	2
3. PROJECT DESCRIPTION	3
4. LAYERS, TAGS AND COLLIDERS.....	3
5. SCRIPTING INFORMATION.....	4

2. IMPORTING INFORMATION

This package needs googleVR SDK 0.8. It can be downloaded from:

<https://drive.google.com/file/d/0B5VocdPbQPqVbTJoRHN5Mmxvams/view?usp=sharing>



3. PROJECT DESCRIPTION

This is a dynamic chess game made for Virtual Reality. The pieces are Quads with /particle/vertexLit Blended materials, which are very efficient for VR applications. All the pieces face the camera and are interactable so it gives a short of 3D realism. Two sets of pieces are presented (white, which are the player's interactable and black, which are static). A loading bar appears when selecting/moving the pieces and the ground changes of color (yellow for selectable and green for possible movements). Natural movement of pieces has been implemented as well as forbidden (collision) movements. Please check our videos for more information.

This package contains the following:

- The necessary FBX, textures, models, scripts shown in the video.
- Scripts manage the interaction with the pieces and scripts that manage the moment of the pieces in the board.
- Two different scenes: one with a complete chessboard and another with one short of each piece in order to easily test the natural movement of pieces.
- Complete documentation/video tutorial to understand the principles of each package and full email support at: michael.soler.beatty@gmail.com.

This package needs the following

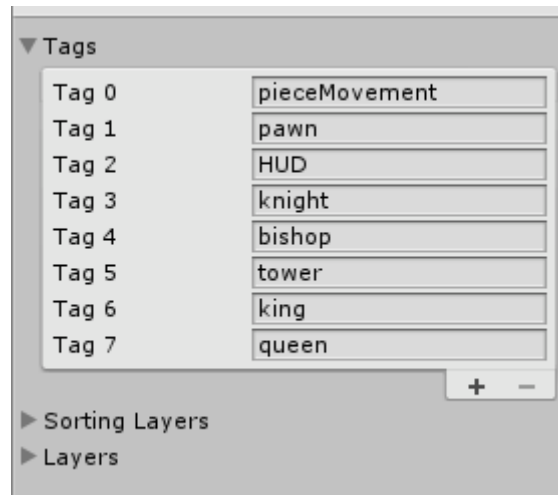
Unity googleVR SDK 0.8 (at least this version).

<https://drive.google.com/file/d/0B5VocdPbQPqVbTJoRHN5Mmxvams/view?usp=sharing>

4. TAGS

TAGS:

Verify that the following tags exist and are well assigned:

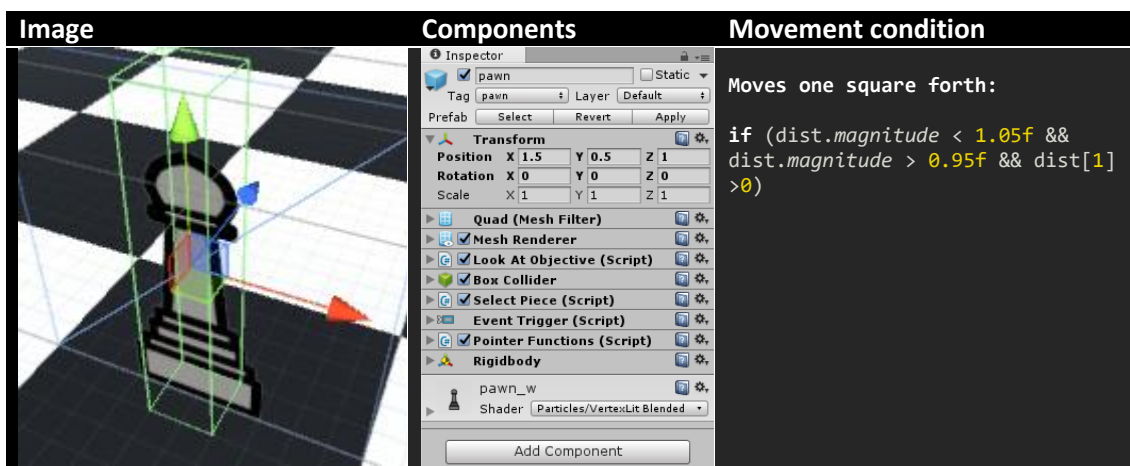


5. PREFABS & SCRIPT INFORMATION

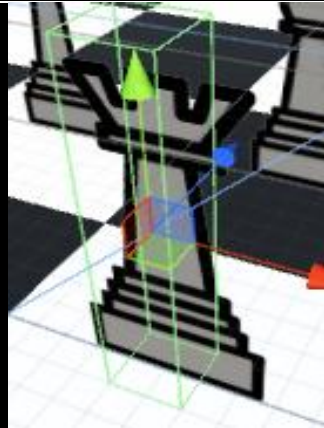
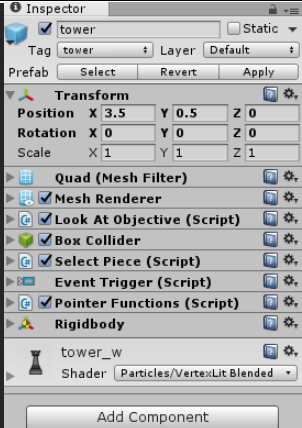
The condition for the movement of pieces is obtained by subtracting the objective position from the current position and adding a condition movement. We also add a tolerance of 0.01f.

```
Vector2 pos1=new Vector2(piecemovScript.selectedGo.transform.position[0],piecemovScript.selectedGo.transform.position[2]);
Vector2 pos2=new Vector2(gameObject.transform.position[0],gameObject.transform.position[2]);
Vector2 dist = ( pos2- pos1);
```

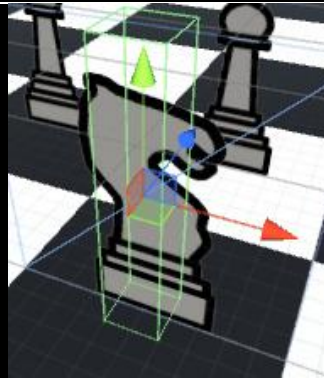

Pawn.prefab



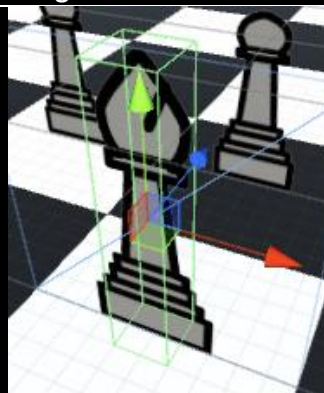
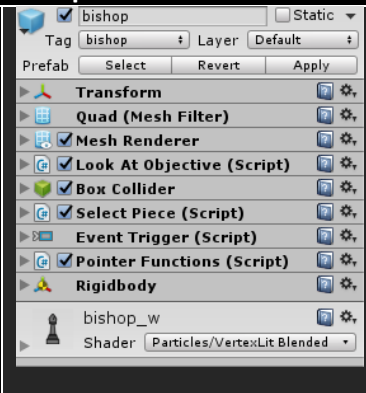
Tower.prefab

Image	Components	Movement condition
		<p>Moves “n” square in a linear way:</p> <pre>if (Mathf.Abs(dist[1])<0.01f Mathf.Abs(dist[0])<0.01f)</pre>

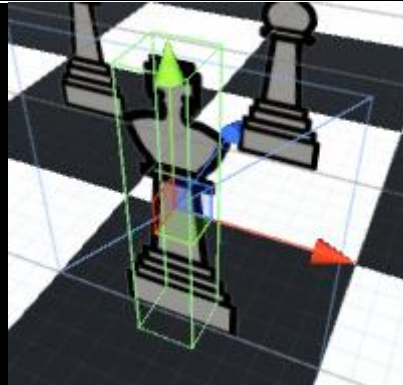
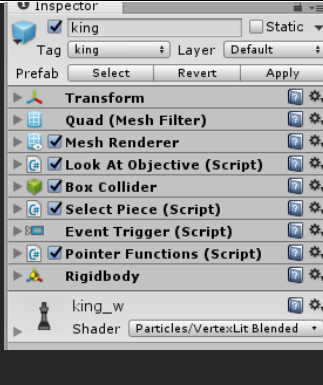
knight.prefab

Image	Components	Movement condition
		<p>Moves in a “L” shape. Notice that it means 2squares in one direction and 1 in the following. $2^2+1^2=5$</p> <pre>if (dist.magnitude < Mathf.Sqrt(5) + 0.01 && dist.magnitude > Mathf.Sqrt(5) - 0.01)</pre>

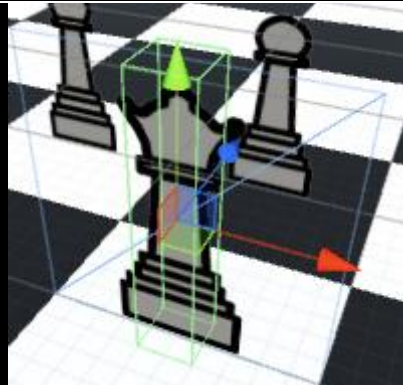
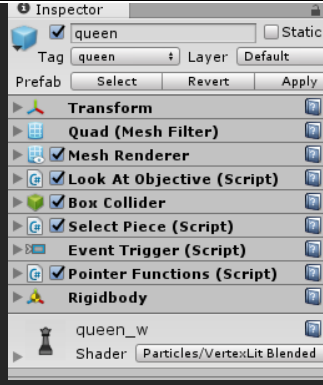
bishop.prefab

Image	Components	Movement condition
		<p>Moves in a “diagonal” way. Notice that it means that both vector components must be equal</p> <pre>if (Mathf.Abs(dist[1]-dist[0])<0.01f Mathf.Abs(dist[1]+dist[0])<0.01f)</pre>

king.prefab

Image	Components	Movement condition
		<p>Moves in a any direction only one square.</p> <pre>dist.magnitude > 1- 0.01 && dist.magnitude < 1 + 0.01 dist.magnitude > Mathf.Sqrt(2) - 0.01 && dist.magnitude < Mathf.Sqrt(2) + 0.01</pre>

Queen.prefab

Image	Components	Movement condition
		<p>Moves in a any direction “n” number of squares. Its movement is like a tower + bishop.</p> <pre>Mathf.Abs(dist[1])<0.01f Mathf.Abs(dist[0])<0.01f Mathf.Abs(dist[1]- dist[0])<0.01f Mathf.Abs(dist[1]+dist[0])<0.01f</pre>

SCRIPTS CLASS REFERENCE

PieceMovement.cs	
It contains the corrutine that allows the movement of pieces in the board from one square to another.	
Important variables	Important functions
<pre>// this is the piece that is selected public GameObject selectedGo; public int timeToMove=2; //these variables are used to un- do the movement in case of a collision between pi eces public bool thereIsCollision=false; public Vector3 lastPosition;</pre>	<pre>// this is the main function that generate s the movement of the piece IEnumerator moveToObjective (Vector3 posit ion)</pre>

SelectPiece.cs	
It says to the PieceMovement.cs which is the selected script. It also generates the “over” effect to select (yellow ground).	
Important variables	Important functions
<pre>// the render is changed to give the selection sensation public Renderer groundRend; // this is the script that manages the piece movement PieceMovement piecemovScript;</pre>	<pre>//this is the function that allows piece selection public void onPieceClick () // these are the actions performed by trigger calls that simulate the over effect for selection public void onPieceOver () public void onPieceExit ()</pre>

ChessSquare.cs	
It manages the “green” squares for the possible movement of the pieces according to tags “pawn”, “bishop”, “queen”... It calls the PieceMovement script’s moveToObjective() function when the movement is possible.	
Important variables	Important functions
<pre>// this is the script that manages the piece movement PieceMovement piecemovScript; // this is the over material used when focusing on the square public Material overMat; // the original material of the square Material origMat; // pointer functions is used to call the prepare to click PointerFunctions pointerF;</pre>	<pre>//this function is trigger on the onOver() trigger event public void onOverSquare () //***** // similar structure that the on over, but with call to the movement function in this case //***** public void onClickSquare ()</pre>

6. VIDEO TUTORIAL FOR SCRIPT/TECHNICAL DETAILS

We are trying to give users a more advanced technical service by creating some videos that explain the main aspects of our packages and gives a detailed explanation about how the package works. You will find this tutorial at the following link:



<https://youtu.be/Rs8RZjfvgDA>