CustomObjects2018 - Instructions

Script inclusion code: <Include CustomObjects2018/Scripts/r1_script.txt>

Box signs function: This function allows you to create box-shaped signs. This is like the exit signs and room signs found in some of my buildings, but more customizable.

Code: AddBoxSign(Name, MainNegativeTex, MainPositiveTex, BoxTexture, Thickness, Direction, DefWallOrient, DefDrawWalls, CenterX, CenterZ, SignWidth, SignHeight, Offset, IsExternal)

Parameters:

1. Name: Sets the name of the sign to be created
2. MainNegativeTex and MainPositiveTex: Sets the texture to be used on the sign's main negative and positive sides respectively.
3. BoxTexture: Sets the texture to be used on all other sides of the sign.
4. Thickness: Sets the sign's thickness. This is the same as in the AddWall command.
5. Direction: Sets the sign's direction. Can be true for front/back facing signs, or false for left/right facing signs.
6. DefWallOrient: If this is set to true, this function uses its default wall orientation setting (center). If set to false, you can add a WallOrientation command before calling this function to set it on your own like when creating regular walls and floors.
7. DefDrawWalls: If this is set to true, this function uses its default draw walls setting (all sides drawn). If set to false, you can add a DrawWalls command before calling this function to set it on your own like when creating regular walls and floors.
8. CenterX and CenterZ: Sets the sign's center point.
9. SignWidth and SignHeight: Sets the sign's width and height.
10. Offset: Sets the sign's offset from the floor's base. If the IsExternal parameter is set to true (see below), this parameter sets the sign's offset from the floor's altitude instead.
11. IsExternal: Sets whether the sign is in the floor mesh or the external mesh.

Example: AddBoxSign(Sign1, ExitSignRight, ExitSignLeft, StainlessSteel, 0.1, false, true, true, 22, 25, 1.8, 0.9, %height% - 0.9, false)

Walls with custom textures function: This function allows you to create walls with different textures on each part. You can set the textures for up to 3 parts with this function.

Code: AddWall_CustomTex(Name, Parts, LTexture, MTexture, UTexture, Thickness, X1, Z1, X2, Z2, LHeight, MHeight, UHeight, Offset, External)

Parameters:

1. Name: Sets the name of the wall to be created.
2. Parts: Sets how many wall's parts to be created. Can be set to 2 or 3.
3. LTexture, MTexture and UTexture: Sets the textures used on the wall's lower, middle, and upper parts respectively. If the Parts parameter is set to 2, then the upper part's texture is specified in the MTexture parameter, and the UTexture parameter is ignored.
4. Thickness: Sets the wall's thickness, like in the AddWall command.
5. X1, Z1, X2 and Z2: Sets the wall's co-ordinates, like in the AddWall command.
6. LHeight, MHeight and UHeight: Sets the height of the wall's lower, middle, and upper parts respectively. If the Parts parameter is set to 2, then the upper part's height is specified in the MHeight parameter, and the UHeight parameter is ignored.
7. Offset: Sets the wall's offset from the floor's base. If the IsExternal parameter is set to true (see below), this parameter sets the wall's offset from the floor's altitude instead.
8. IsExternal: Sets whether the wall is in the floor mesh or the external mesh.

Example: AddWall_CustomTex(Left, 2, LobbyWallLower, LobbyWallUpper, Blank, 0.5, -120, -2, -120, 5, 4, %height% - 4, 0, 0, false)