Palace modules

Palace modules is a set of elements for building levels for 2.5D and 3D games. Each block is fitted with 1 x 1 meter in such a way as to build a monolithic building. Thus from a single square floor you can build a whole floor, from one wall a long wall, in the gaps of the walls you can put the door. The textures are also adjusted so that the different elements have a continuation!

The height of the walls, doors and columns is 2 m, the height of the stairs is 1 m (so that you can make corners)

Palace modules include the following elements:
• Column
• Door1
• Door2
• Doors with grate
• Doors with curtain
• Windows (Doors with stained glass, including 12 textures)
• Floor
• Floor with spikes
• Stairs
• Stand
• Stand with Brazier
• Switch platform (for opening the door)
• Wall
• Brazier
• Fire
• Palm tree (in a pot)
• Torch
• Carpet patches for floor (several variants), doors and stairs
• Ceilings

Blocks have textures and materials for the Palace, Ruins and Royal, prefabs have been created for each species and sorted by corresponding folders.

The ceilings are made in 15 different variants for all types of room walls (straight walls, internal corners, external corners, columns standing side by side). To increase the ceiling level, you need to move it by 0.5m along the Y axis.

Doors

Doors have animated grates that rise when the player stays on the switch platform.

Each door has its own switch, which is configured in the inspector, **keyhole** (int) for the door and **key** (int) for the switch. When the player becomes on the switch with the given key, all doors are opened, for which keyhole == key. After some time (adjusted in the switch), the door closes again, 0 - Infinity.

In order for the player to open the doors, in the player's script in the **OnTriggerEnter ()** function, it is necessary to refer to the **Switch** component and call the **Open ()** function

```
if (other.GetComponent<Switch>())
{
        other.GetComponent<Switch>().Open();
}
```

Spikes

When the player stands on the floor with spikes, the spikes will quickly jump out and pierce the player.

To activate the spikes, in the player's script in the **OnTriggerEnter ()** function, you need to refer to the **Spikes** component and call the **SpikesOn ()** function

```
if (other.GetComponent<Spikes>())
{
        other.GetComponent<Spikes>().SpikesOn();
        PlayerDead();
}
```

Fire

The fire contains an animation of 16 alternating pictures in the texture of the fire 4 x 4. The fire is in 4 planes, for viewing from different angles.

Royal Treasures

Excellent for awards, prizes and premium in any medieval or fantasy games!

This package contains:

• Banners

• Chalice

• Chests (Red, Wood and Marble)

• Luxury Pillows

• Antique Shield

• Simple Gems

• Sword

• Carpet

• Crown

Chests are animated and open when the player approaches them closely.

When the player's collider touches the collider of the chest, in the player's script **OnTriggerEnter ()** function is called, in which the **Chest** component calls the **Open ()** function, which starts the desired animation.

```
if (other.GetComponent<Chest>())
{
        other.GetComponent<Chest>().Open();
}
```

Also, the old version of the treasure pack is kept, in folder "(old version) Royal Treasures" contains:

• Crown

• Chests

• Heaps of Treasure

• Swords

• Greek Shield

• Ancient Pots

All textures PNG format 512, 1024, and 2048.

Materials includes diffuse, normals and occlusion maps.

Player Death

A player can die in the following cases:

• when he gets to the floor with spikes

• when the grate of the door falls on him

• when the player falls from a height of ~ 4m

The height of the fall can be configured by changing the parameter in the **PlayerMove.cs** script on line 55:

else if (rbody.velocity.y < **-8f**)

CamInvisible

When there are objects overlapping the view between the player and the camera, they are hidden. The script works regardless of whether the objects have colliders or not, it hides all objects that have a MeshRenderer component.

Technical support:

https://sergeyiwanski.wixsite.com/gamedev/contacts