

Сделал все так что при присоединении игрока ему сетался рандомный материал. Рандомное число индекса материала задается на сервере

```
export class Player extends Schema {
    @type("number") x = Math.floor(Math.random() * 256) - 128;
    @type("number") z = Math.floor(Math.random() * 256) - 128;
    @type("uint8") d = 2;
    @type("uint8") clr = Math.floor(Math.random() * 6);
}
Alexander [6 days ago] - first commit
```

Далее в классе MultiplayerManager при создании змейки игрока и енеми мы передаем это число:

```
Snake snake = Instantiate(_snakePrefab, position, rotation);
snake.Init(player.d, player.clr);
```

И затем при создании всех частей змейки (Snake, Detail, Tail) передаю это число:

```
public void Init(int detailCount, byte color) {
    GetComponent<SetSkin>().SetMaterial(color);

    _tail = Instantiate(_tailPrefab, transform.position, Quaternion.identity);
    _tail.Init(_head, _speed, detailCount, color);
}
```

есть 2 класса: Skins и SetSkin

Класс Skins просто весит на сцене, хранит возможные материалы и возвращает необходимый, вынес его отдельно, чтобы добавлять материалы можно было в одном месте

```
public class Skins : MonoBehaviour {
    public static Skins Instance;

    [SerializeField] private Material[] _materials; // Serializable

    private void Awake() {
        if (Instance == null) {
            Instance = this;
            DontDestroyOnLoad(gameObject);
        }
        else {
            Destroy(gameObject);
        }
    }

    public Material GetMaterial(byte index) => _materials[index];
}
```

А класс SetSkin весит на этих префабах, хранит необходимые меши, на которые устанавливает выбранный материал

```
public class SetSkin : MonoBehaviour {
    [SerializeField] private MeshRenderer[] _meshRenderers; // Serializable

    public void Set(Material material) {
        for (int i = 0; i < _meshRenderers.Length; i++) {
            _meshRenderers[i].material = material;
        }
    }

    public void SetMaterial(byte index) => Set(Skins.Instance.GetMaterial(index));
}
```