

o Contexto (o que estavas a trabalhar, e porquê precisavas de ajuda):
 Estava a criar o inventario e nao sabia exatamente como fazer o hud da hotbar
 o Prompt (texto exato)
 Preciso que me ajudes a criar uma hotbar em javascript mostra um exemplo para entender melhor
 o Output (apenas a parte relevante)

```

// ===== CRIAÇÃO DOS SLOTS =====
for (int slotIndex = 0; slotIndex < HOTBAR_SLOT_COUNT; slotIndex++) {

  BitmapText slotText = new BitmapText(font, false);
  slotText.setSize(FONT_SIZE);

  float slotX = hotbarStartX + slotIndex * (SLOT_WIDTH + SLOT_SPACING);

  // posição do texto do slot
  slotText.setLocalTranslation(slotX, HOTBAR_Y, 0);

  // opcional: mostrar número do slot
  slotText.setText(String.valueOf(slotIndex + 1));

  guiNode.attachChild(slotText);
  hotbarTexts[slotIndex] = slotText;
}
  
```

o Decisão (usar/alterar/rejeitar)
 alterar
 o Reflexão (1–3 frases sobre o que aprendeste)
 Assim consegui percer +- a logica que tinha que usar no meu codigo para obter o resultado desejado

o Contexto (o que estavas a trabalhar, e porquê precisavas de ajuda):

Estava a criar NPC e não sabia como criar os bonecos com uma skins como o zombie

o Prompt (texto exato)
 "Cria um método getModel em Java para JMonkeyEngine. Quero que construas um Zombie visualmente simples, feito apenas com geometrias primitivas parecido com o do minecraft"

o Output (apenas a parte relevante)

```

public Spatial buildZombieModel(AssetManager assetManager) {
  // Nó principal que vai conter todas as partes do corpo
  Node zombieRoot = new Node("ZombieVisualRoot");

  // Definição dos Materiais (Pele e Roupa)
  
```

```

Material skinMaterial = createMaterial(assetManager, new ColorRGBA(0.0f, 0.5f, 0.0f,
1.0f));
Material clothingMaterial = createMaterial(assetManager, new ColorRGBA(0.0f, 0.0f, 0.6f,
1.0f));

// Geometria da Cabeça
Geometry headGeom = new Geometry("Head", new Box(0.25f, 0.25f, 0.25f));
headGeom.setMaterial(skinMaterial);
headGeom.setLocalTranslation(0, 1.55f, 0);
zombieRoot.attachChild(headGeom);

// Geometria do Tronco (Corpo)
Geometry torsoGeom = new Geometry("Torso", new Box(0.3f, 0.45f, 0.15f));
torsoGeom.setMaterial(clothingMaterial);
torsoGeom.setLocalTranslation(0, 0.85f, 0);
zombieRoot.attachChild(torsoGeom);

// Braço Esquerdo
Geometry leftArmGeom = new Geometry("LeftArm", new Box(0.12f, 0.45f, 0.12f));
leftArmGeom.setMaterial(skinMaterial);
// Roda o braço para ficar esticado para a frente
leftArmGeom.setLocalRotation(new Quaternion().fromAngleAxis(-FastMath.HALF_PI,
Vector3f.UNIT_X));
leftArmGeom.setLocalTranslation(-0.42f, 1.15f, 0.45f);
zombieRoot.attachChild(leftArmGeom);

// Braço Direito
Geometry rightArmGeom = new Geometry("RightArm", new Box(0.12f, 0.45f, 0.12f));
rightArmGeom.setMaterial(skinMaterial);
rightArmGeom.setLocalRotation(new Quaternion().fromAngleAxis(-FastMath.HALF_PI,
Vector3f.UNIT_X));
rightArmGeom.setLocalTranslation(0.42f, 1.15f, 0.45f);
zombieRoot.attachChild(rightArmGeom);

// Perna Esquerda
Geometry leftLegGeom = new Geometry("LeftLeg", new Box(0.13f, 0.45f, 0.13f));
leftLegGeom.setMaterial(clothingMaterial);
leftLegGeom.setLocalTranslation(-0.15f, 0.45f, 0);
zombieRoot.attachChild(leftLegGeom);

// Perna Direita
Geometry rightLegGeom = new Geometry("RightLeg", new Box(0.13f, 0.45f, 0.13f));
rightLegGeom.setMaterial(clothingMaterial);
rightLegGeom.setLocalTranslation(0.15f, 0.45f, 0);
zombieRoot.attachChild(rightLegGeom);

// Ajuste final da posição de todo o modelo
zombieRoot.setLocalTranslation(0, -0.9f, 0);

```

```
    return zombieRoot;
}
```

o Decisão (usar/alterar/rejeitar)

Aceitar

o Reflexão (1–3 frases sobre o que aprendeste)

Com isso entendi como fazer o zombie no meu projeto

Contexto (o que estavas a trabalhar, e porquê precisavas de ajuda)

Estava a criar NPC e não sabia como criar os bonecos com uma skins como o Gigante do clash Royale

o Prompt (texto exato)

"Cria um método getModel em Java para JMonkeyEngine. Quero que construas um Gigante visualmente simples, feito apenas com geometrias primitivas parecido com gigante do clash royale"

o Output (apenas a parte relevante)

```
public Spatial buildGiantModel(AssetManager assetManager) {
    // Nó principal do Gigante
    Node giantRootNode = new Node("GiantVisualRoot");

    // Definição dos Materiais
    Material skinMaterial = createMaterial(assetManager, new ColorRGBA(0.96f, 0.76f, 0.62f,
1f));
    Material coatMaterial = createMaterial(assetManager, new ColorRGBA(0.6f, 0.4f, 0.2f,
1f));
    Material pantsMaterial = createMaterial(assetManager, new ColorRGBA(0.2f, 0.2f, 0.2f,
1f));
    Material beardMaterial = createMaterial(assetManager, new ColorRGBA(0.3f, 0.15f, 0.05f,
1f));

    // Fator de escala (Gigante é 1.8x maior)
    float sizeScale = 1.8f;
```

```

// Cabeça
Geometry headGeom = new Geometry("Head", new Box(0.25f * sizeScale, 0.28f *
sizeScale, 0.25f * sizeScale));
headGeom.setMaterial(skinMaterial);
headGeom.setLocalTranslation(0, 1.65f * sizeScale, 0);
giantRootNode.attachChild(headGeom);

// Barba
Geometry beardGeom = new Geometry("Beard", new Box(0.26f * sizeScale, 0.1f *
sizeScale, 0.15f * sizeScale));
beardGeom.setMaterial(beardMaterial);
beardGeom.setLocalTranslation(0, 1.55f * sizeScale, -0.1f * sizeScale);
giantRootNode.attachChild(beardGeom);

// Tronco
Geometry torsoGeom = new Geometry("Torso", new Box(0.5f * sizeScale, 0.55f *
sizeScale, 0.35f * sizeScale));
torsoGeom.setMaterial(coatMaterial);
torsoGeom.setLocalTranslation(0, 0.85f * sizeScale, 0);
giantRootNode.attachChild(torsoGeom);

// --- Braços ---

// Ombro Esquerdo
Geometry leftShoulderGeom = new Geometry("LeftShoulder", new Box(0.2f * sizeScale,
0.2f * sizeScale, 0.2f * sizeScale));
leftShoulderGeom.setMaterial(coatMaterial);
leftShoulderGeom.setLocalTranslation(-0.6f * sizeScale, 1.2f * sizeScale, 0);
giantRootNode.attachChild(leftShoulderGeom);

// Braço Esquerdo
Geometry leftArmGeom = new Geometry("LeftArm", new Box(0.18f * sizeScale, 0.45f *
sizeScale, 0.18f * sizeScale));
leftArmGeom.setMaterial(skinMaterial);
leftArmGeom.setLocalTranslation(-0.6f * sizeScale, 0.6f * sizeScale, 0.2f * sizeScale);
// Rotação ligeira no braço
leftArmGeom.setLocalRotation(new Quaternion().fromAngleAxis(-0.3f, Vector3f.UNIT_X));
giantRootNode.attachChild(leftArmGeom);

// Ombro Direito
Geometry rightShoulderGeom = new Geometry("RightShoulder", new Box(0.2f *
sizeScale, 0.2f * sizeScale, 0.2f * sizeScale));
rightShoulderGeom.setMaterial(coatMaterial);
rightShoulderGeom.setLocalTranslation(0.6f * sizeScale, 1.2f * sizeScale, 0);
giantRootNode.attachChild(rightShoulderGeom);

// Braço Direito

```

```

        Geometry rightArmGeom = new Geometry("RightArm", new Box(0.18f * sizeScale, 0.45f *
sizeScale, 0.18f * sizeScale));
        rightArmGeom.setMaterial(skinMaterial);
        rightArmGeom.setLocalTranslation(0.6f * sizeScale, 0.6f * sizeScale, 0.2f * sizeScale);
        rightArmGeom.setLocalRotation(new Quaternion().fromAngleAxis(-0.3f,
Vector3f.UNIT_X));
        giantRootNode.attachChild(rightArmGeom);

        // --- Pernas ---

        // Perna Esquerda
        Geometry leftLegGeom = new Geometry("LeftLeg", new Box(0.22f * sizeScale, 0.35f *
sizeScale, 0.22f * sizeScale));
        leftLegGeom.setMaterial(pantsMaterial);
        leftLegGeom.setLocalTranslation(-0.25f * sizeScale, 0.35f * sizeScale, 0);
        giantRootNode.attachChild(leftLegGeom);

        // Perna Direita
        Geometry rightLegGeom = new Geometry("RightLeg", new Box(0.22f * sizeScale, 0.35f *
sizeScale, 0.22f * sizeScale));
        rightLegGeom.setMaterial(pantsMaterial);
        rightLegGeom.setLocalTranslation(0.25f * sizeScale, 0.35f * sizeScale, 0);
        giantRootNode.attachChild(rightLegGeom);

        // Ajuste final da posição
        giantRootNode.setLocalTranslation(0, -0.9f, 0);

        return giantRootNode;
    }

```

o Decisão (usar/alterar/rejeitar)

Aceitar

o Reflexão (1–3 frases sobre o que aprendeste)

Com isso entendi como fazer o Gigante no meu projeto

Contexto (o que estavas a trabalhar, e porquê precisavas de ajuda)

Estava a criar NPCs e não sabia como criar os bonecos com skins como um vilager e uma homem diferente

o Prompt (texto exato)

"Cria um método getModel em Java para JMonkeyEngine. Quero que construas dois villagers diferentes visualmente simples, feito apenas com geometrias primitivas parecido com o vilager do minecraft"

o Output (apenas a parte relevante)

```
@Override
public Spatial buildFarmerModel(AssetManager assetManager) {
    // Nó principal do Agricultor
    Node farmerRootNode = new Node("FarmerVisualRoot");

    // Definição dos Materiais
    Material skinMaterial = createMaterial(assetManager, new ColorRGBA(0.96f, 0.76f, 0.62f,
1f));
    Material shirtMaterial = createMaterial(assetManager, ColorRGBA.Yellow);
    Material pantsMaterial = createMaterial(assetManager, new ColorRGBA(0.4f, 0.25f, 0.1f,
1f));
    Material hatMaterial = createMaterial(assetManager, new ColorRGBA(0.8f, 0.6f, 0.2f, 1f));

    // Cabeça
    Geometry headGeom = new Geometry("Head", new Box(0.25f, 0.25f, 0.25f));
    headGeom.setMaterial(skinMaterial);
    headGeom.setLocalTranslation(0, 1.55f, 0);
    farmerRootNode.attachChild(headGeom);

    // --- Chapéu (composto por duas partes) ---

    // Aba do chapéu
    Geometry hatBrimGeom = new Geometry("HatBrim", new Box(0.35f, 0.02f, 0.35f));
    hatBrimGeom.setMaterial(hatMaterial);
    hatBrimGeom.setLocalTranslation(0, 1.78f, 0);
    farmerRootNode.attachChild(hatBrimGeom);

    // Topo do chapéu
    Geometry hatTopGeom = new Geometry("HatTop", new Box(0.20f, 0.12f, 0.20f));
    hatTopGeom.setMaterial(hatMaterial);
    hatTopGeom.setLocalTranslation(0, 1.9f, 0);
    farmerRootNode.attachChild(hatTopGeom);

    // Tronco
    Geometry torsoGeom = new Geometry("Torso", new Box(0.3f, 0.45f, 0.15f));
    torsoGeom.setMaterial(shirtMaterial);
    torsoGeom.setLocalTranslation(0, 0.85f, 0);
    farmerRootNode.attachChild(torsoGeom);
```

```

// Braço Esquerdo
Geometry leftArmGeom = new Geometry("LeftArm", new Box(0.12f, 0.45f, 0.12f));
leftArmGeom.setMaterial(shirtMaterial);
leftArmGeom.setLocalTranslation(-0.42f, 0.85f, 0);
farmer rootNode.attachChild(leftArmGeom);

// Braço Direito
Geometry rightArmGeom = new Geometry("RightArm", new Box(0.12f, 0.45f, 0.12f));
rightArmGeom.setMaterial(shirtMaterial);
rightArmGeom.setLocalTranslation(0.42f, 0.85f, 0);
farmer rootNode.attachChild(rightArmGeom);

// Perna Esquerda
Geometry leftLegGeom = new Geometry("LeftLeg", new Box(0.13f, 0.45f, 0.13f));
leftLegGeom.setMaterial(pantsMaterial);
leftLegGeom.setLocalTranslation(-0.15f, 0.45f, 0);
farmer rootNode.attachChild(leftLegGeom);

// Perna Direita
Geometry rightLegGeom = new Geometry("RightLeg", new Box(0.13f, 0.45f, 0.13f));
rightLegGeom.setMaterial(pantsMaterial);
rightLegGeom.setLocalTranslation(0.15f, 0.45f, 0);
farmer rootNode.attachChild(rightLegGeom);

// Ajuste final
farmer rootNode.setLocalTranslation(0, -0.9f, 0);

return farmer rootNode;
}

```

o Decisão (usar/alterar/rejeitar)

Aceitar

o Reflexão (1–3 frases sobre o que aprendeste)

Com isso entendi como fazer os villagers do meu projeto

Contexto (o que estavas a trabalhar, e porquê precisavas de ajuda)

Cria criar o hud do craft mas não sabia exatamente como o fazer

o Prompt (texto exato)

Escreve um trecho de código em **Java** para o **jMonkeyEngine** que inicialize a interface de um **Menu de Craft**.

1. Cria um título que diga 'MENU DE CRAFT' a grande
2. Cria um texto de ajuda que diga: 'C: fechar | TAB: rato | Clique: craftar'.
3. Considerando que existe uma lista de receitas (**recipes**), inicializa arrays para guardar os textos das receitas e as coordenadas dos botões (X, Y, largura e altura).

o Output (apenas a parte relevante)

```
// Título do Menu
menuTitleText = new BitmapText(guiFont);
menuTitleText.setText("MENU DE CRAFT");
menuTitleText.setSize(guiFont.getCharSet().getRenderedSize() * 1.5f);

// Texto de Instruções (Ajuda)
instructionsText = new BitmapText(guiFont);
instructionsText.setText("C: fechar | TAB: rato | Clique: craftar");

// Preparar os arrays para os botões e textos
// recipes.length é a quantidade de receitas que tens no jogo
recipeLabelTexts = new BitmapText[recipes.length];

// Arrays que definem a área de cada botão (Posição X/Y e Tamanho)
buttonXPositions = new int[recipes.length];
buttonYPositions = new int[recipes.length];
buttonWidths    = new int[recipes.length];
buttonHeights   = new int[recipes.length];

// Inicializar os textos de cada receita
for (int i = 0; i < recipes.length; i++) {
    recipeLabelTexts[i] = new BitmapText(guiFont);
}
```

o Decisão (usar/alterar/rejeitar)

aceitar

o Reflexão (1–3 frases sobre o que aprendeste)

Aprendi o que tinha de fazer para criar a interface do craft