UX/UI Design.

Sempre bom pesquisar interface do usuário de um jogo parecido com o que você vai criar. Se baseei e inove em cima, não tente inventar a roda.

Existe 2 sites:

* [Game UI Database | Welcome](https://www.gameuidatabase.com/)
* [Interface In Game | Collection of video games UI | Screenshots and videos](https://interfaceingame.com/)

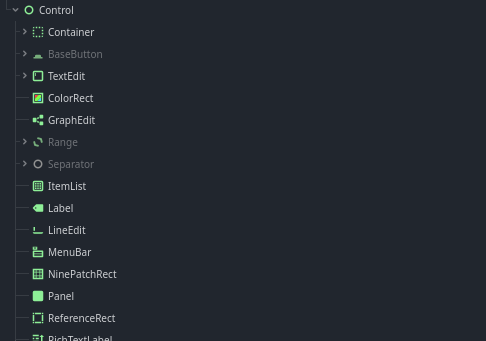
Nesses 2 sites eles coletam fotos, vídeos e etc de interface especifica. Ex: Gamer Over.

**Vamos por a mão na massa:**

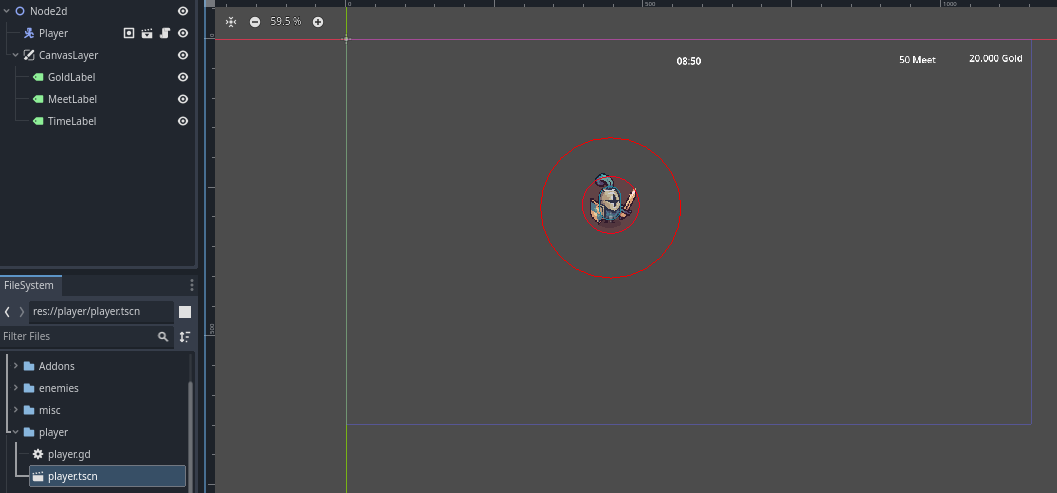
1ª Criamos uma cena chamada teste\_ui para desenvolver todas nossas ui nela.

2ª Criamos um node chamado Canvaslayer, criamos isso pois os elementos da interface do usuário não pertencem ao mundo (pelo menos nem todos eles), eles pertencem a outra camada do jogo. O canvaslayer permite desenhar coisas na tela de forma independente em relação ao node2d.

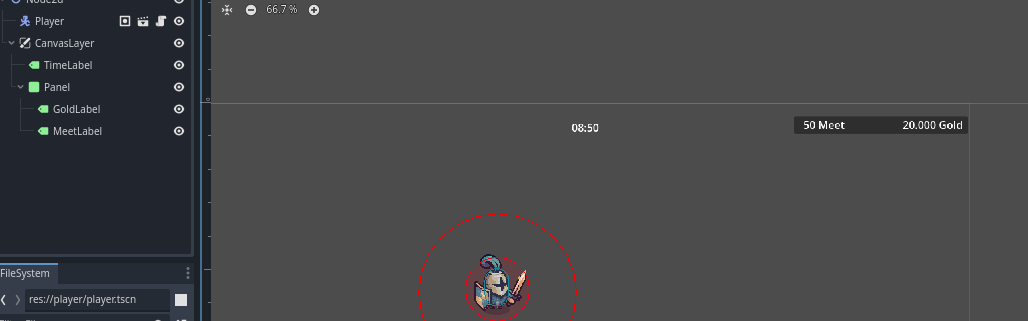
Se perceber, tudo relacionado a interface do usuário é “verde” dentro do grupo Control.

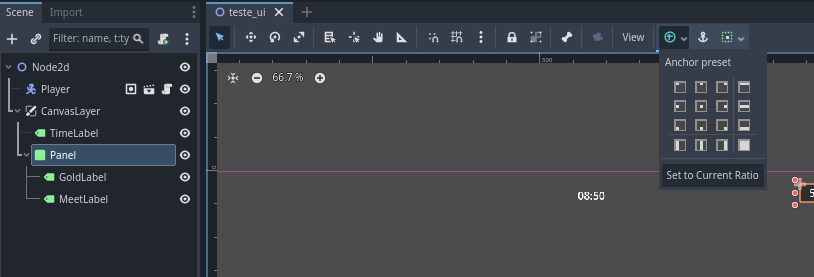


Fiz vários labels e ancorei na direita da tela.



Inclusive eu posso criar um “Panel” e colocar os textos nele:

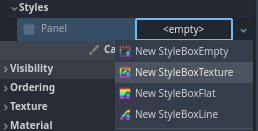




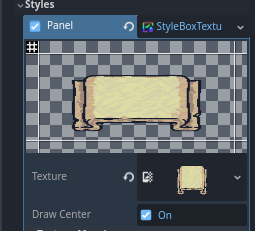
Com a ancora no canto, sempre vai estar na tela na direita.

**Textura na interface do usuário:**

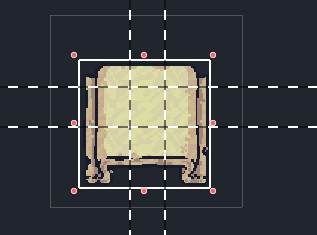
Vamos anexar a textura que a gente quer em um panel. Para isso, criamos um Panel e iremos em “Theme Overrides”



Fomos no pacote e em UI escolhemos uma imagem e arrastramos para a texture

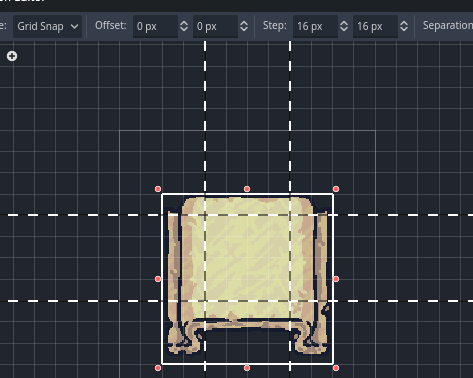


Só que fica feios só estender o elemento. Então no StyleBox tem uma configuração que se chama “Sub-Region” e nela tem um botão “edit region” ao clicar nela da para cortar o elemento em varias partes



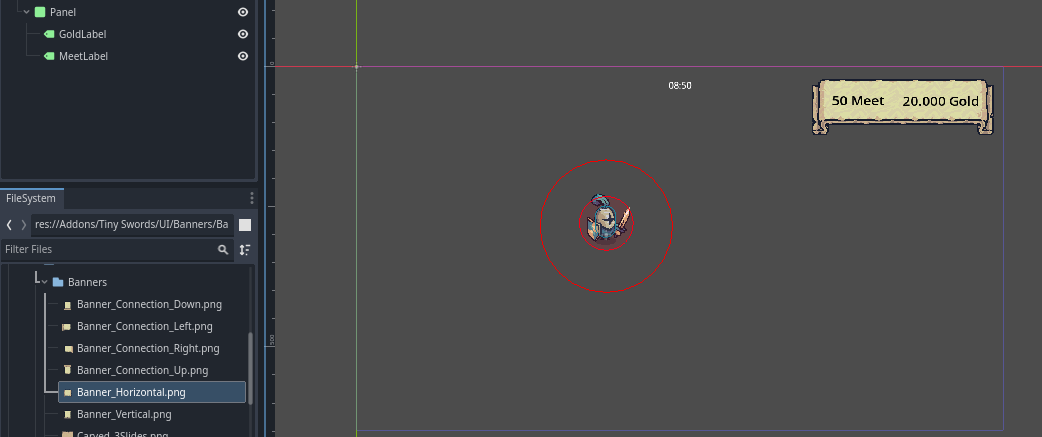
Delimitando qual parte pode ser esticada

Mas podemos fazer com que o tamanho fique exatamente correto. No snap mode colocar Grid Snap e configurar



Agora mexendo na configuração do stylebox chamada “Axis Stretch” e mudar para tile fit

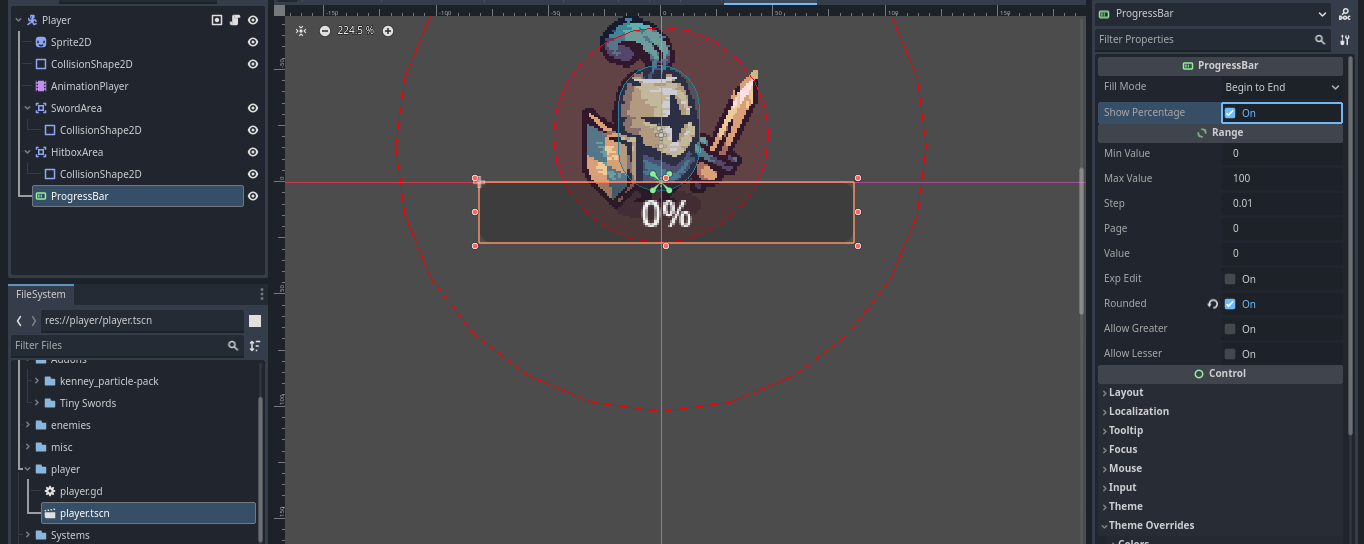




**Agora iremos criar a vida que fica embaixo do personagem:**

Na cena do player iremos criar um node chamado “ProgressBar”.

Com ela podemos definir um valor mínimo e um valor máximo, valor atual e etc.

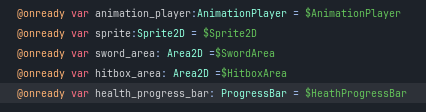


E personalizar no Theme Overides.

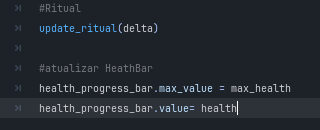


Agora programando ela:

No script do player:

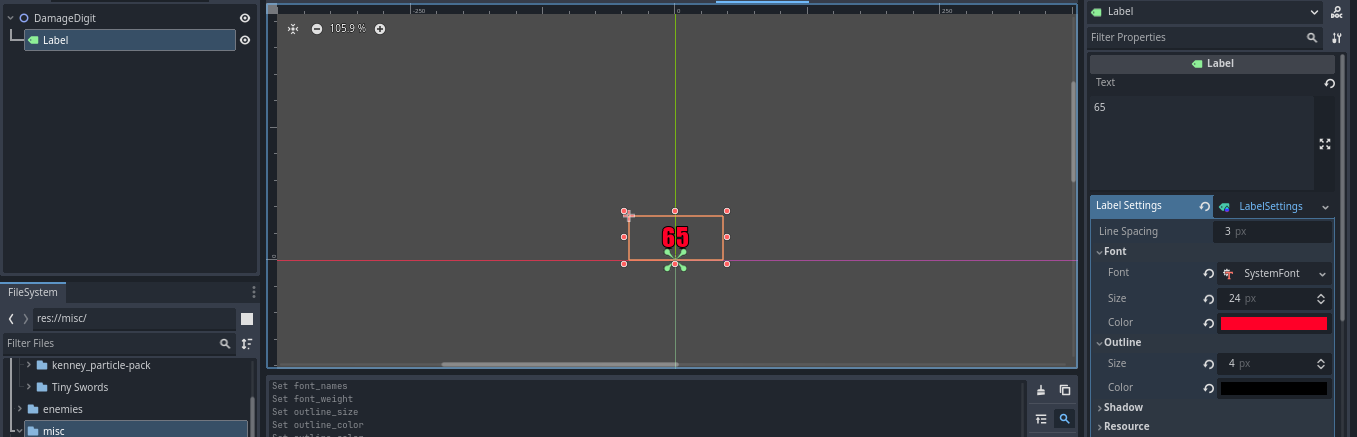


E na função process:



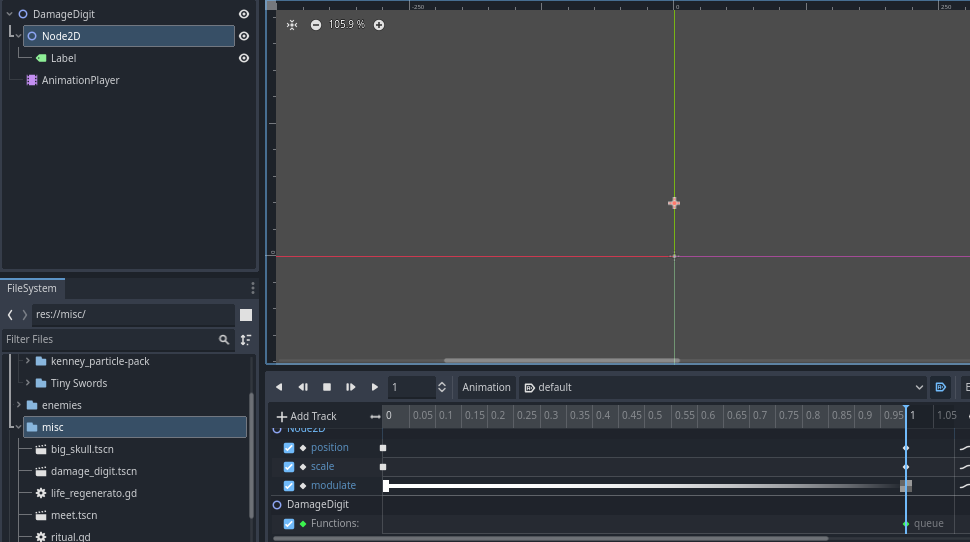
**Programar agora para mostrar quanto de dano demos no inimigo:**

Criamos uma cena nova em misc e dentro dela vamos criar um label:



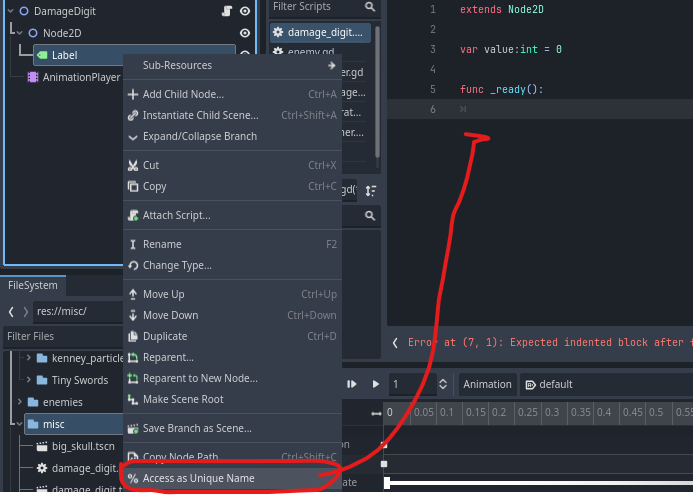
Personalizamos do jeito que a gente quer e praticamente ta pronto. Agora é só incorporar no script.

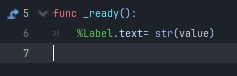
Mas vamos colocar uma animatedplayer para fazer algo diferente. E para personalizar vamos criar um node2d e jogar o label para dentro dele.



Mudamos a position, scale e modulate do node2d e colocamos uma função queue free para deletar ao final da animação.

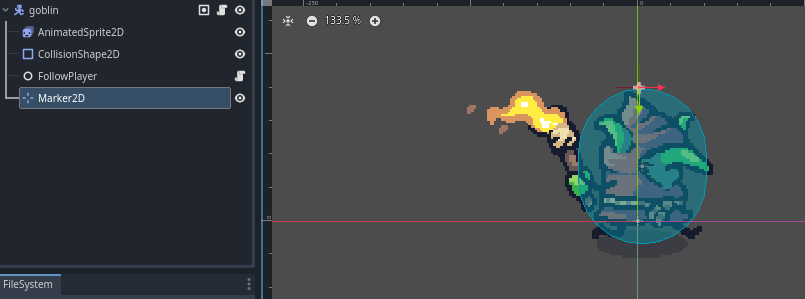
E agora? Como pegamos o dano que o inimigo toma? Criamos um script no DamageDigit.





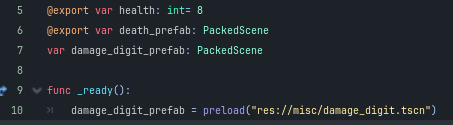
Agora em cada inimigo eu vou criar uma posição: (mark2d)

Que é um marcador de posição.



E renomeamos e copiamos e colamos em cada inimigo.

Agora no script do inimigo:



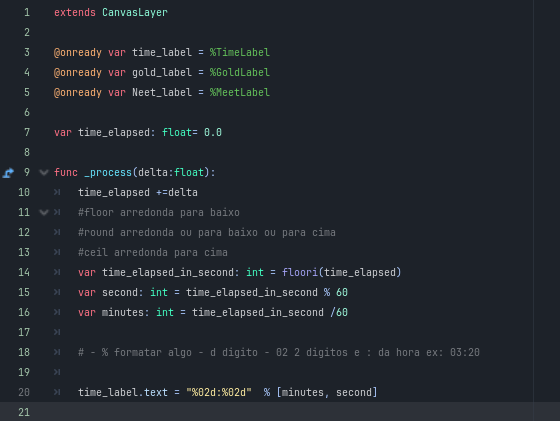
Fazemos uma função ready para sempre carregar a cena do damage\_digit

Agora criando quando toma dano:



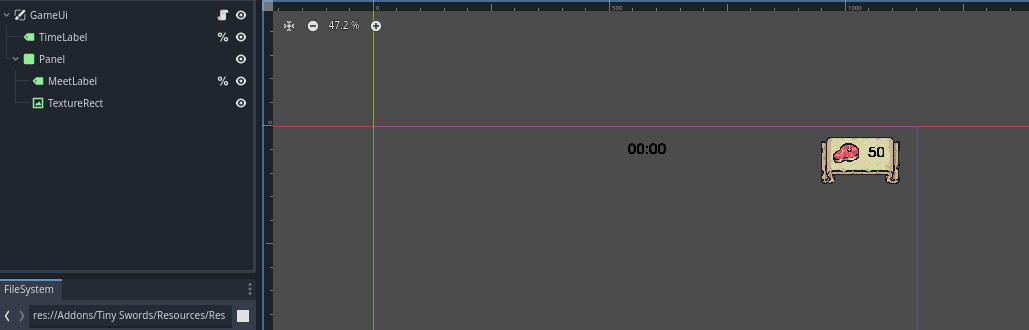
**Programando o restante da cena UI:**

Vamos criar um script e 1ª coisa que vamos fazer é mexer no tempo.



**Mexer no meat:**

Vamos colocar no lugar de um label, um TextureRect que é um bloco que tem uma imagem.



Programando ela:

Vamos usar 4 scripts: Game\_ui, game\_manager, life\_ regenerato e player.

No player vamos fazer um signal para disparar toda vez que a gente coletar uma carne:

