

Tutorial Classe I_QuestionController1

Inicialização Default Da Classe QuestionController1

Através do Framework, e tendo ele já instalado no Unity, vai ser utilizada a classe I_QuestionController1 com sua inicialização padrão, que vai permitir criar automaticamente uma pergunta (com elementos padrões de exemplo), na qual as alternativas são textos 2D tipo UI tradicional. Para isso, devem-se seguir os seguintes passos:

Nota: Preparar cena final chamada Scene1,

- 1) Criar um objeto vazio do Unity (GameObject/Create Empty), ver figura 1, que será usado como container para as classes que permitirão criar a pergunta (classes I_QuestionController1 e QuestionController1E).



Figura 1: Criação de um objeto vazio do Unity.

- 2) Selecionar o objeto vazio, e arrastar nele a classe I_QuestionController1 (contida na pasta MainScripts). A cena deveria ficar como na figura 2.

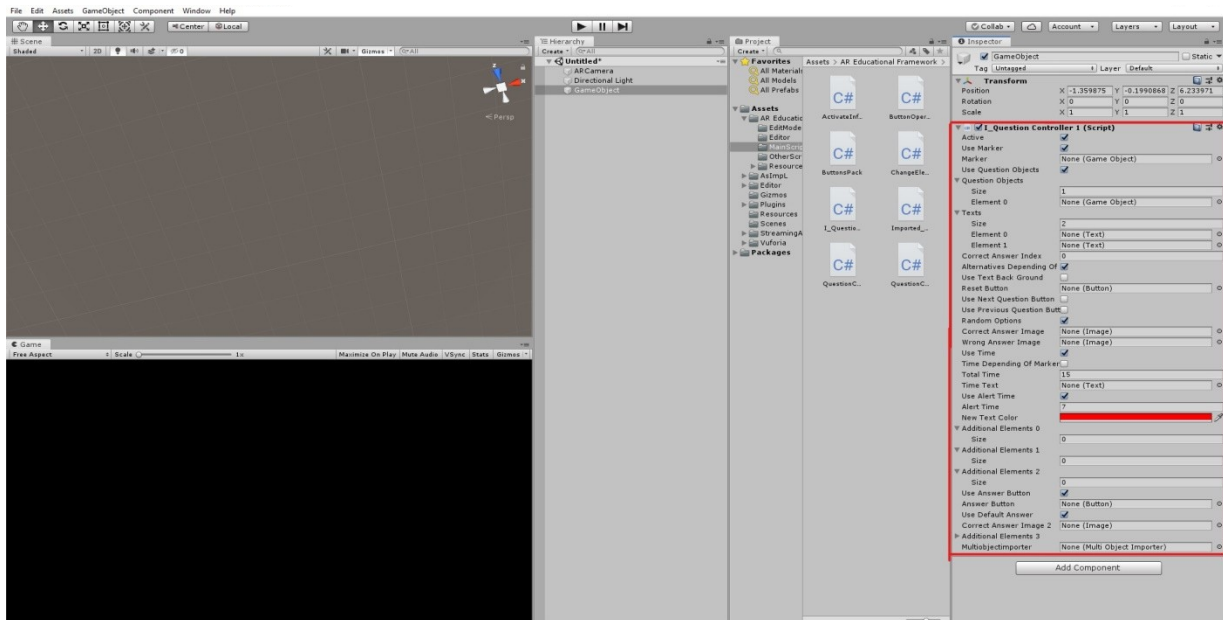


Figura 2: Classe I_QuestionController1 arrastada no objeto vazio.

- 3) Selecionar o objeto vazio, e arrastar nele a classe I_QuestionController1E (contida na pasta EditMode) para a criação automática dos elementos default, após isso recomenda-se fazer click no marcador criado ImageTarget para fazer ele visível na cena.

A cena deveria ficar como na figura 3 tendo na parte direita da tela dois textos usados como alternativas de resposta, no centro duas imagens de feedback de resposta certa e errada com botões para ver resposta certa e repetir pergunta, no corner superior direito um texto para mostrar o tempo, e finalmente na parte superior da tela dois textos adicionais para pergunta e resposta respetivamente.

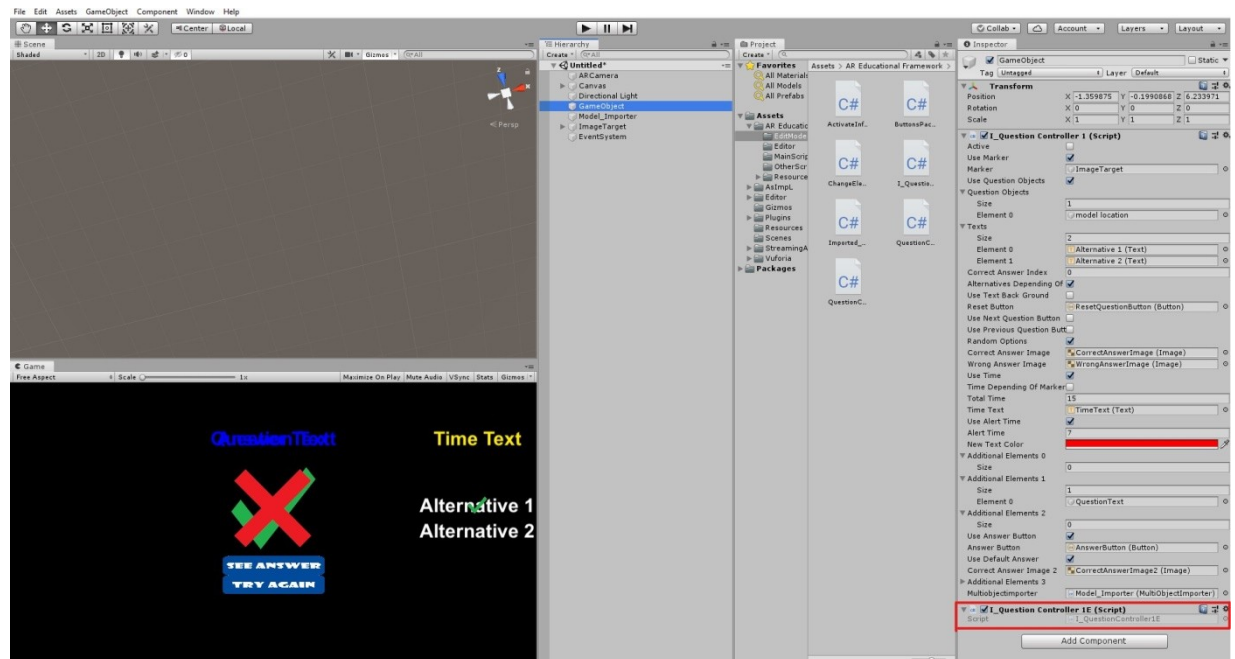


Figura 3: Classe I_QuestionController1E arrastada no objeto vazio.

- 4) Após ser feita a inicialização padrão recomenda-se remover a classe I_QuestionController1E.
- 5) Criado todos os objectos, é necessario a criação dos arquivos para lietura na pasta Assets/StreamingAssets. Primeiro crie um arquivo .txt de nome igual ao gmeobject que contém a classe I_QuestionController1 (no caso, o nome foi alterado para Question1, então o arquivo chama Question1.txt)

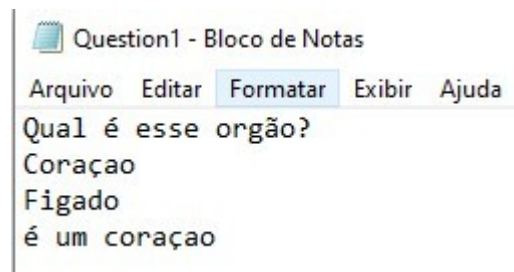
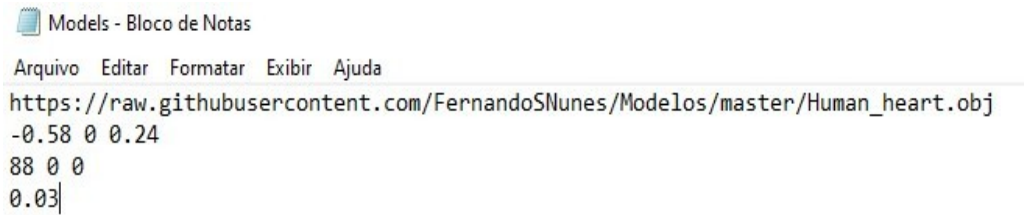


Figura 4: Arquivo Question1.txt criado com as informações que serem exibidas quando ele estiver ativo. As informações devem estar sempre nessa ordem sendo: Pergunta, alternativa 1, alternativa 2 (o numero de alternativas é especificado no unity, bem como qual a correta que por padrão é a primeira) e por fim o texto exibido quando o usuario errar a questão, sendo cada um em uma linha.

- 6) Depois, criar o arquivo models.txt com o endereço dos modelos a serem utilizados. Os modelos devem estar salvos em alguma plataforma em que se consiga acessar o raw deles, como o github e enviar esse link.



Figur

a 5: Arquivo models que contém as informações do modelo a ser carregado. Elas devem seguir essa ordem: Endereço do modelo, as 3 coordenadas sobre a posição em que o modelo deve estar separadas por um espaço, em seguida a rotação que o modelo irá receber também divididas por um espaço e por fim a escala que ele terá. **Esses dados podem ser obtidos em um programa para testar os modelos.** No caso de múltiplos modelos a serem carregados, os novos devem ser colocados nas linhas consecutivas seguindo a mesma ordem, sem pular linhas entre eles.

Configurar Classe I_QuestionController1 Para Usar Múltiplas Perguntas

A seguir será explicado o jeito de usar múltiplas perguntas numa cena, para o exemplo, a pergunta atual ficará igual, e será criada uma nova pergunta onde o segundo texto

alternativa será configurado de acordo com um outro arquivo .txt de nome igual a ele, assim uma vez que responder a primeira pergunta será possível ir para a nova. Para isso, devem-se seguir os seguintes passos:

Nota: Uma cena final chamada Scene3,

- 1) Duplicar o objeto vazio original que possui a classe `I_QuestionController1` fazendo click direito nele e na opção Duplicate. Para melhor compreensão os objetos vazios são renomeados como `Question1` e `Question2` respectivamente, a cena deveria ficar como na figura 6. Sendo que a primeira pergunta está especificada no model importer (ao final da importação ele irá ativar ela).

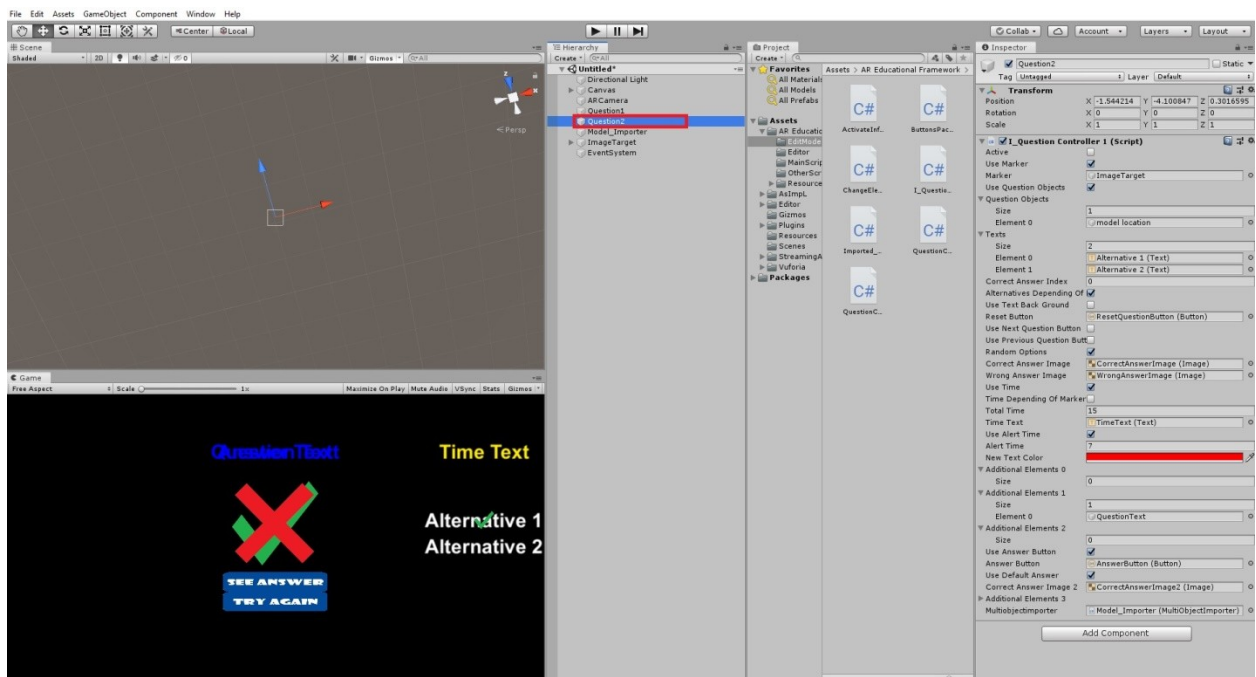


Figura 6: Novo objeto vazio criado duplicando o original. Os objetos vazios tem sido renomeados como `Question1` e `Question2` respectivamente.

- 2) Criar textos alternativas da nova pergunta, para isso, neste exemplo são duplicados os textos alternativa existentes localizados no Canvas (Alternative1 e Alternativa2), renomeado os textos novos como `Alternative1Question2` e `Alternative2Question2` respectivamente. Após isso, os textos novos são arrastados nos campos Texts do objeto vazio `Question2` como na figura 7. Nota: Com a classe `I_QuestionController1` não é suportado o reuso dos elementos usados como alternativas, e portanto deve ser feito este passo a mais criando novos textos alternativa para a nova pergunta.

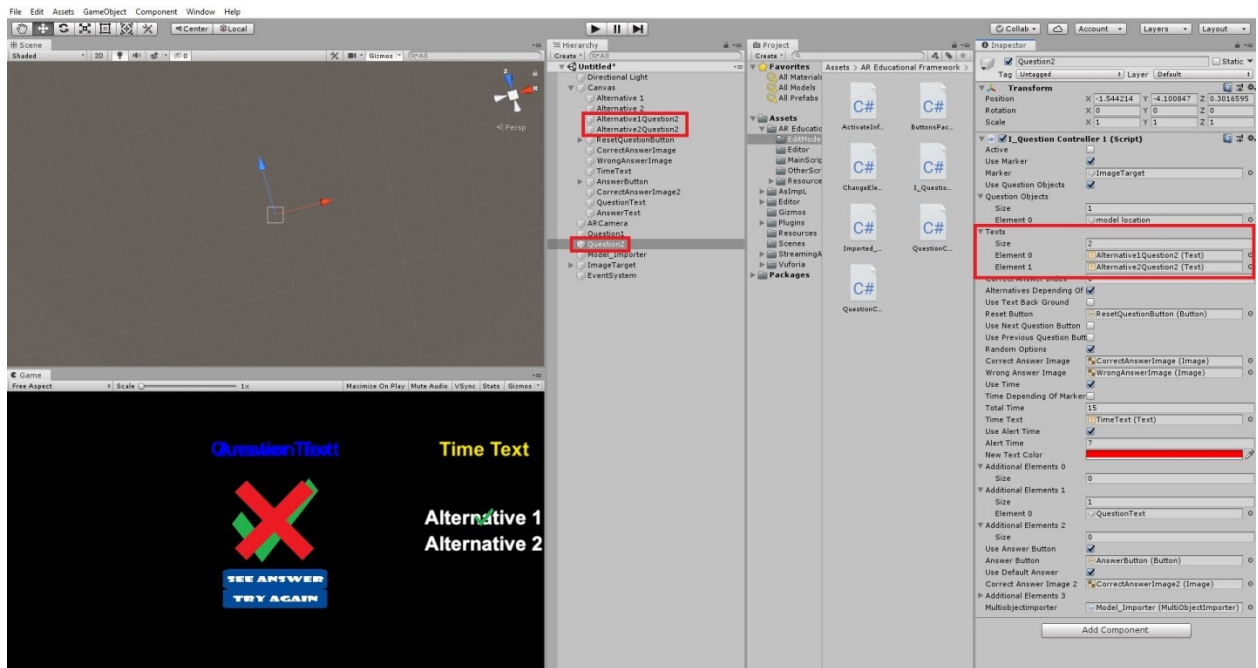


Figura 7: Novos textos de alternativa criados e arrastados nos campos Texts de Question2.

- 3) Criar o botão de próxima pergunta (GameObject/UI/Button) e posicionar ele embaixo do botão TryAgain. Para isso, neste exemplo se duplica o botão ResetQuestionButton, se posiciona o novo embaixo do original na vista da cena renomeando lá como NextQuestionButton, e se arrasta a imagem Next (localizada no caminho: AR Educational Framework/Resources/ButtonTextures), no campo SourceImage do botão, ficando como na figura 8.

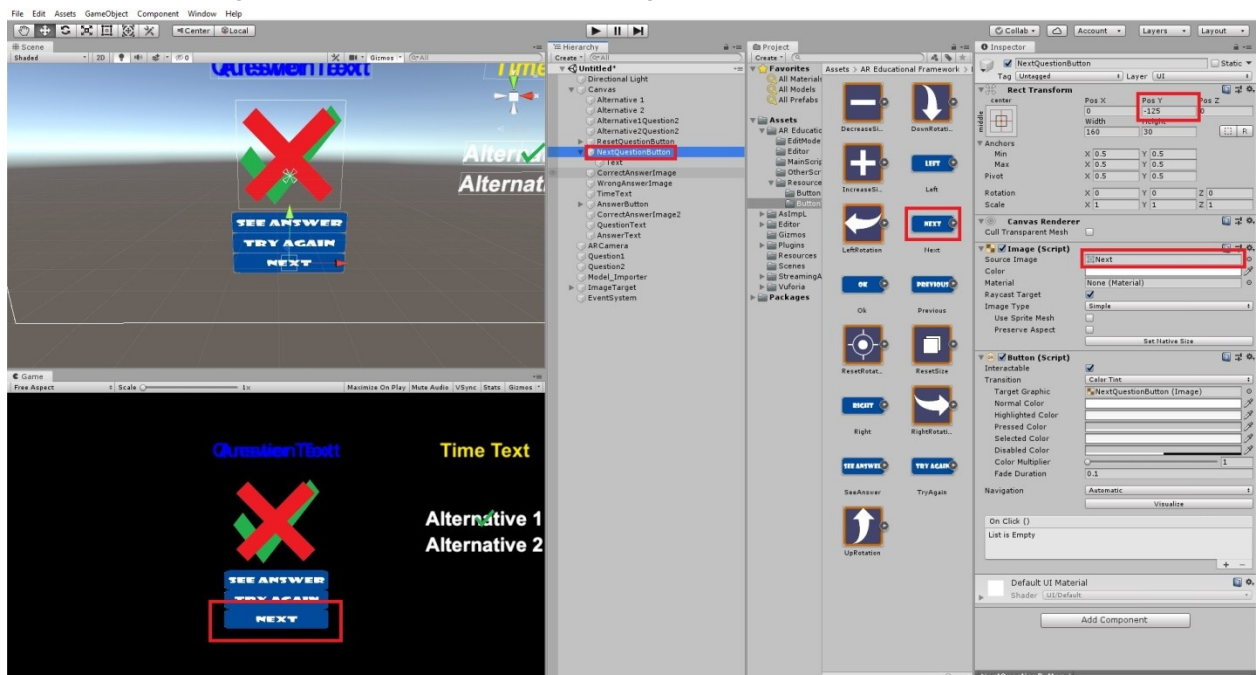


Figura 8: Novo botão NextQuestionButton da próxima pergunta já criado, e com a imagem Next arrastada no seu campo SourceImage.

- 4) Selecionar o objeto vazio Question1, ativar o campo UseNextQuestionButton para usar o botão de ir à próxima pergunta, e finalmente arrastar no campo NextQuestionButton o botão criado no passo anterior com o mesmo nome, e no

campo NextQuestion o objeto vazio Question2 que se refere à próxima pergunta, a cena deve ficar como na figura 9.

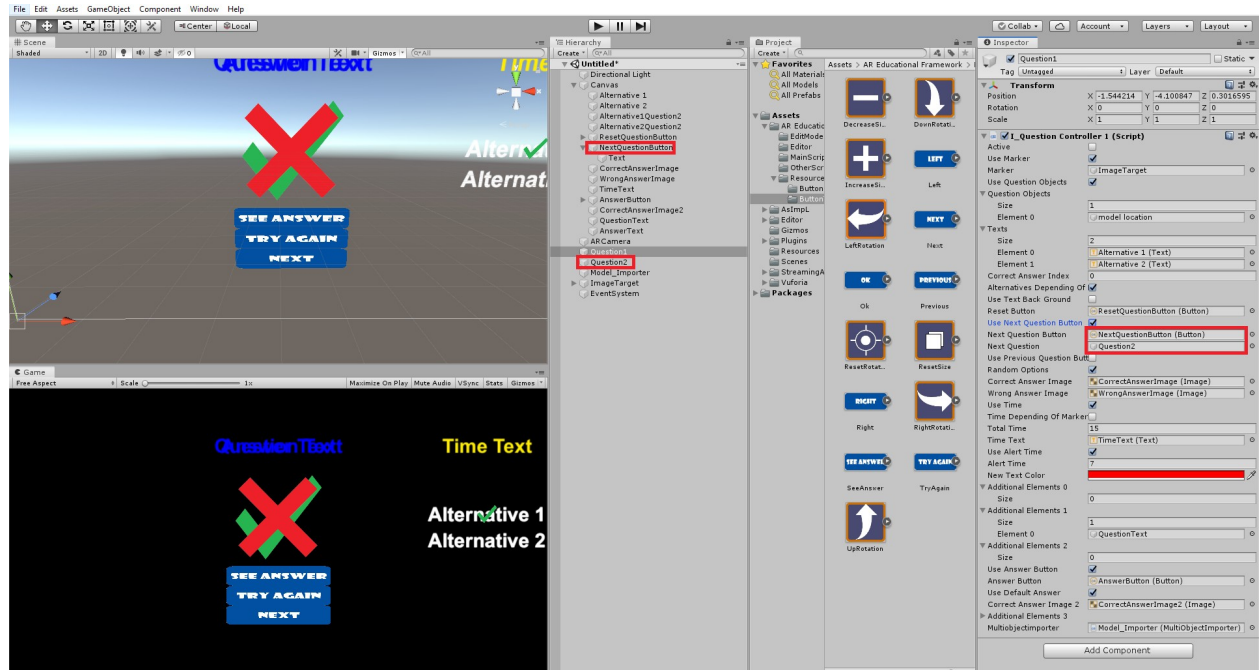


Figura 9: Campo UseNextQuestionButton ativado no objeto vazio Question1, para usar botão de ir à próxima pergunta.