MENU KIT 1.2

Documentação

Esse pacote foi desenvolvido por Wilgner Fábio

Obrigado a meu amigo Paulo Camacan por fazer o script UITransition

Contato

wilgner.fabio@gmail.com

Obrigado por comprar

Sumário

Por favor deixar uma review na Assets Store

Isso dá um feedback para mim sobre o que adicionar/melhorar e outros.

Obrigado.

MMK (Main Menu Kit)

Introdução

O Menu Kit é um pacote que contém um menu completo para seu jogo, é simples, plano e fácil de personalizar ao seu gosto, contem 4 menus prontos, **play**, **options**, **credits** e **exit**. Todos funcionando perfeitamente e com animações incluídas, além de uma caixa de diálogo, idiomas, transições, animações que podem ser feitas facilmente com alguns cliques. Opções funcionais e é salva em arquivo **json**.

Características

- Cena de demonstração
- Fácil de usar e personalizar
- Inclui configurações de opções (VIDEO, AUDIO and GAME)
- Controle de Idiomas (EN, PT_BR)
- Input customizado (Sugestão enviado no meu Email)
- Pause Menu (RESUME, OPTIONS and EXIT) (Sugestão no vídeo de apresentação do Main Menu Kit)
- Player Simples
- GUI plana e bonita
- 2 Planos de fundos
- Transições de animações (Fade In, Fade Out, Swap In, Swap Out, Custom)
- Salvar em arquivo JSON
- Cursor personalizável
- 2 tipos de play (Iniciar o jogo direto e modos de jogos, level select em desenvolvimento)
- Animações no Menu Inicial

- Animações de escala, direções ou ambas na tela de créditos e em modos de jogo
- Estrutura organizada
- Todos códigos em C# e comentados em inglês (em breve português do Brasil)
- Prefabs prontos para serem usados
- Editor Customizável
- Organização dos menus e modos de acordo com a posição no inspetor
- Documentação em PDF
- Melhorias no inspetor e códigos
- Trabalha com a Unity +2017

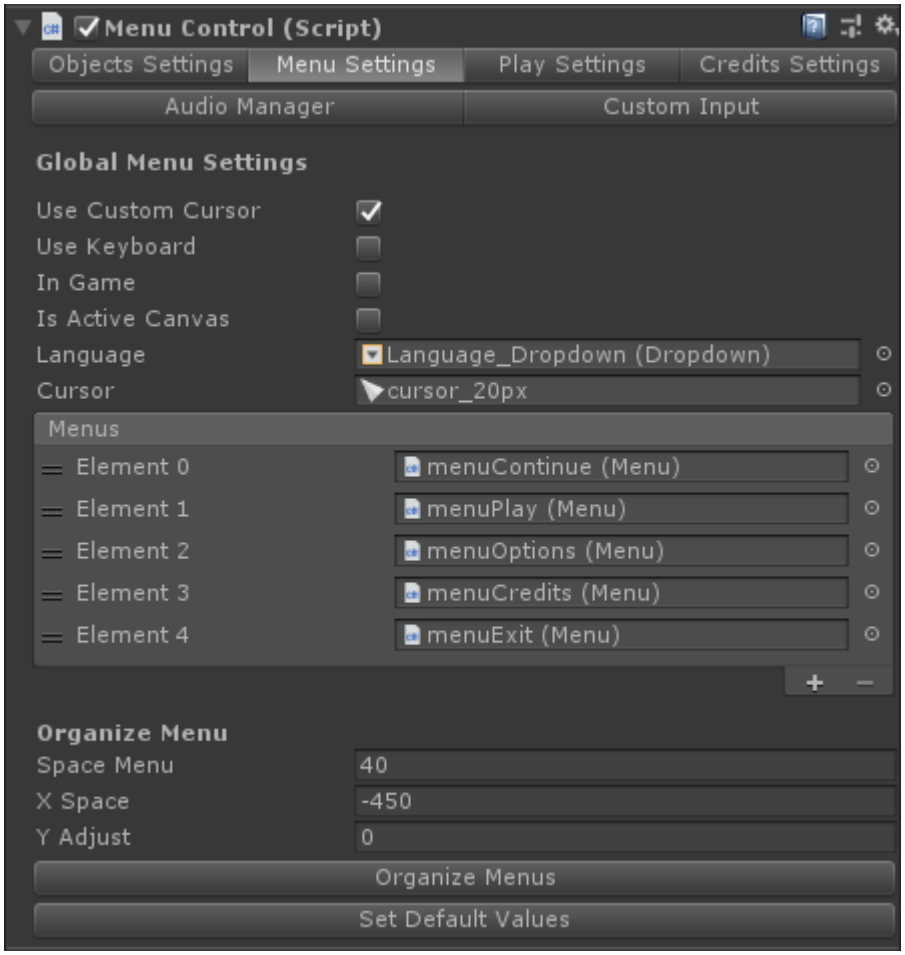
O que fazer primeiro?

Antes de importa o Menu Kit eu recomendo você fazer um backup em seu projeto e deve criar uma nova scene para o menu ou criar um projeto novo (recomendado) para seguir essa documentação, realizar modificações profundas e não acabar prejudicando outro projeto.

Após isso, você vai no menu acima, do lado da tab Windows tem Wilgner's Studio > MenuKit > Main Menu ou pressione CTRL+SHIFT+M com isso você já tem um menu funcionando perfeitamente.

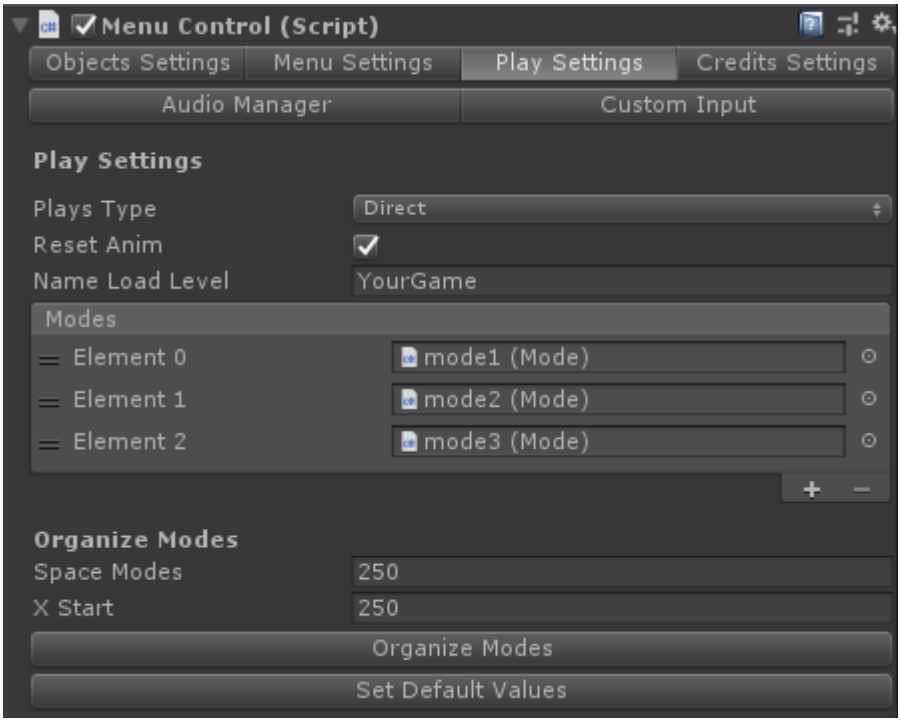
Recomendo você sempre pegar os prefabs padrões (como citado acima) e fazer alterações para que não precise fazer do 0.

Configurando o Menu Control



(Em ordem)

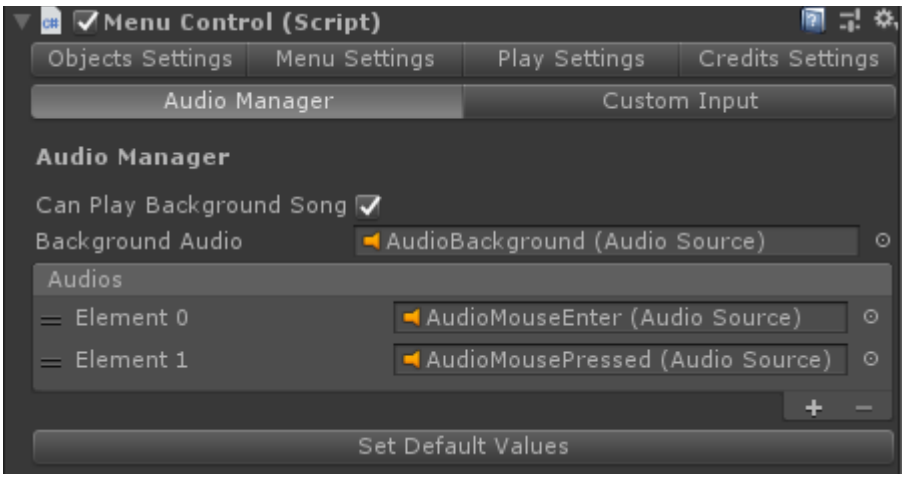
1. Utilizar um cursor customizado
2. Ao marcar essa opção pode movimentar no menu através das teclas (em desenvolvimento, funciona somente no menu principal)
3. Marque caso esse seja o menu de pause
4. Dropdaw para modificar o idioma
5. Imagem mostrada no cursor caso a 1 esteja ativada (recomendo 15-20px tamanho)
6. Menus, você pode organizar como você quer e depois clicar em “Organize Menus”



(Em ordem)

1. Tipo de plays, pode ser direto(jogo inicia diretamente), Modes (caso tenha modo de jogo, exemplo: singleplayer, training, casual)
2. Reinicia a animação modes toda vez que ir para esse menu
3. Nome da cena a ser carregado
4. Assim como o menu, você pode organizar o modos e clicar em “Organize Modes”

1. Tipo de animação (Direction: Como um filme; Scale: Pode ser estática ou escalar o tamanho)
2. Posição em que o crédito acabara
3. Direção que ele vai se mover (ao modificar aqui é importante alterar o item 2 também)
4. Tamanho inicial
5. Tamanho Máximo
6. Velocidade (Init Scale to Max Scale)
7. Velocidade em que os créditos se movem (50 unit unity)



1. Áudio de fundo pode tocar ao iniciar o jogo/menu
2. Áudio ao ser tocado caso a 1º esteja marcada

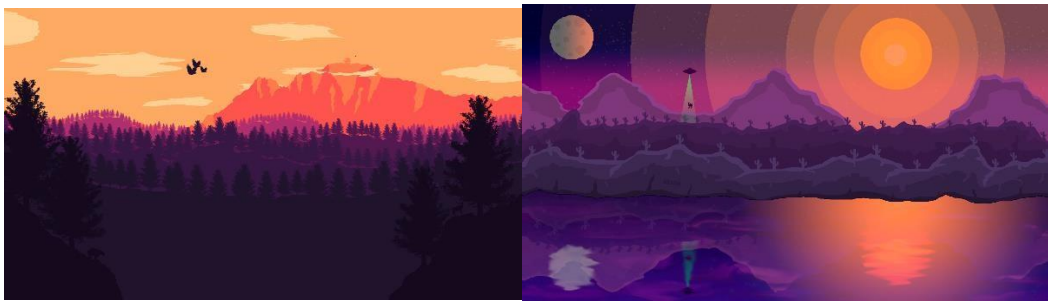
3. Lista de Áudios que o menu utilizara (pode adicionar um para troca de tela por exemplo)

Entrada customizada dos controles (básica)

Configurando o plano de fundo

Primeiramente vamos alterar o plano de fundo, é nada mais nada menos que uma image, o caminho dele é: MenuKit > Canvas > MenuKit_Canvas > backgroundImage

Selecione o Source Image no inspetor e selecione a imagem desejada, eu criei dois planos de fundo plano:



Você pode excluir/desativar a imagem de fundo para usar um cenário (como em PUBG, Overwatch).

Configurações Básicas

Configurações básicas é onde está o título e versão todas screens (exceto a de créditos) contem essa mesmo

configurações, basta acessa-la e mudar no inspetor ou adicionar uma imagem ao invés do text.

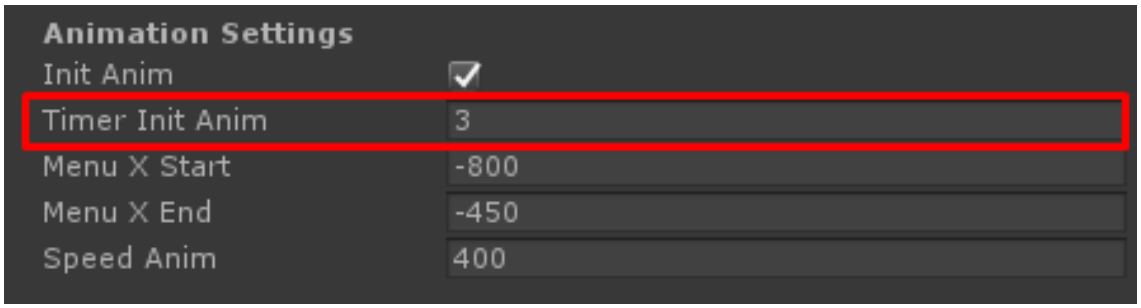
Optei por repetir os elementos para ter um maior controle caso você queira animalos ou fazer outras configurações.

Adicionando um menu

Você deseja um novo menu? Sem problemas, para facilitar o processo você pode ir por aba Wilgner's Studio ou botão direito na hierarquia e adicionar um New Menu ou pegar o prefab e arrastar até MenuKit > Canvas > MenuKit_Canvas > MainScreen > Menus, você também pode copiar e colar um menu pronto.

Agora no MenuControl clica em mais, adicione (arrasta) o novo menu criado e clique em "Organize Menus".

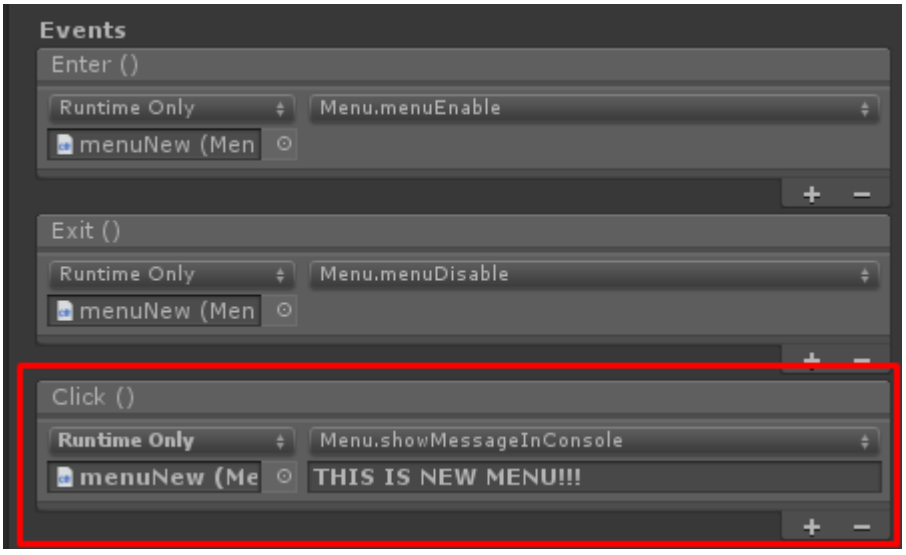
Nesse exemplo de menu estou utilizando a animação inicial de 0.5 de diferença, então no novo menu de vez ser 2.5 você altera para 3 e assim por diante e após isso você pode altera no inspetor o TEXT.



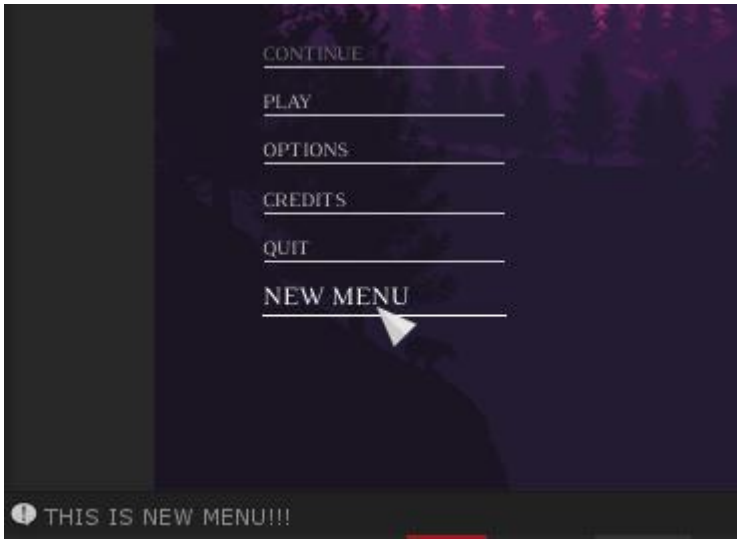
Simples né? Já temos um novo menu:

Por enquanto ele faz a ação do prefab ou o do menu copiado, para alterar procuramos o Events no próprio Menu.cs, os nomes dos eventos já sugestivo a ação que ele fara.

Por exemplo vou adicionar uma mensagem no console, no evento Click() arraste o Menu.cs(acima do inspetor) pra dentro do click e vamos em No Function > Menu > showMessageInConsole e digitamos algum texto.



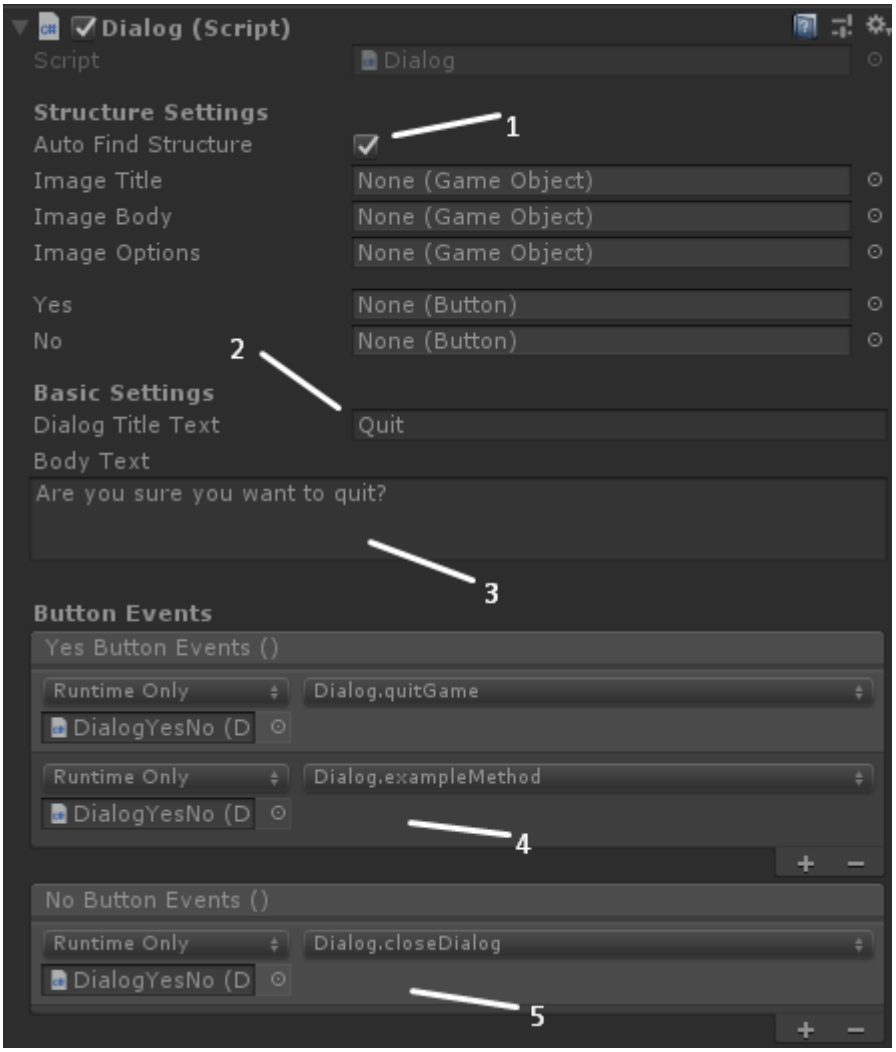
Agora ao executarmos o projeto e clicar no nosso novo menu:



Muito simples. Parabéns, você já criou um item de menu novo.

Criando uma Caixa de Dialogo

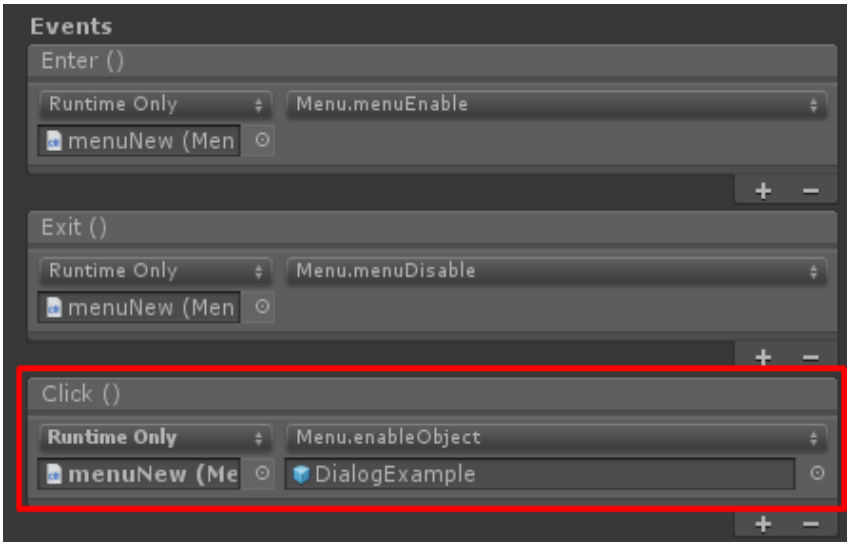
Para criar um caixa de diálogo com botões sim e não, pegue o prefab e arrasta-o para a scene ou use a tab customizada, algumas configurações importantes:



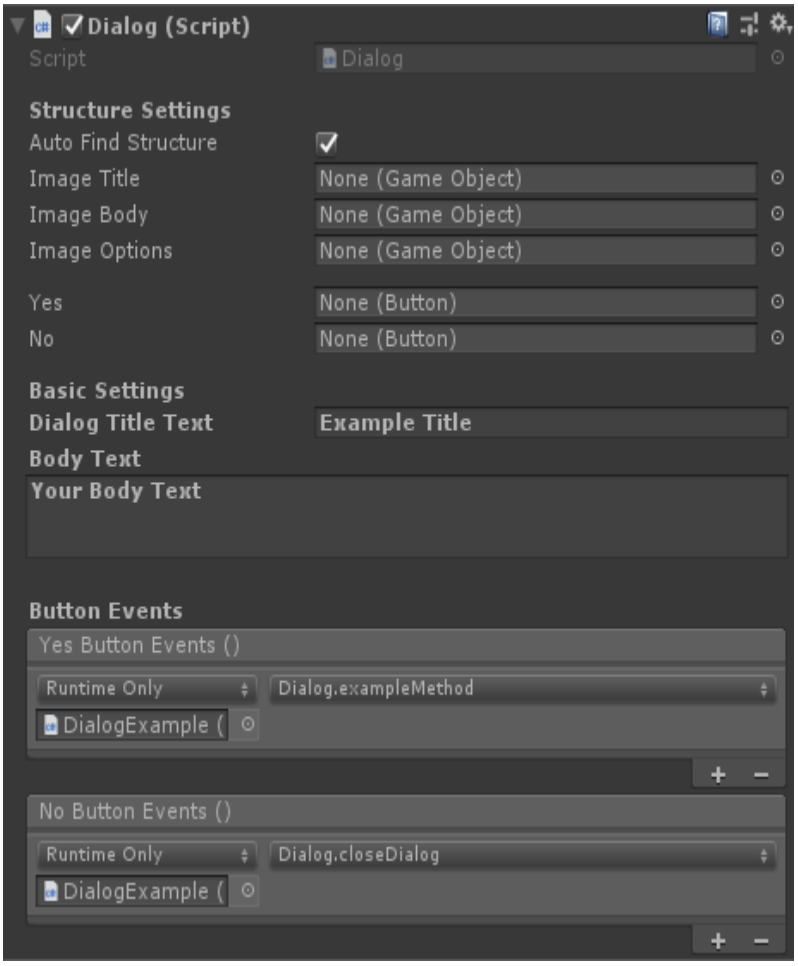
1. Essa opção encontra automaticamente toda a hierarquia necessária para o funcionando do Dialog. Você pode desativa-la e colocar manualmente.
2. Esse é o título da Caixa de Dialogo
3. O corpo de texto / mensagem
4. Todos eventos que ocorrem ao clicar no botão yes
5. Todos eventos que ocorrem ao clicar no botão no

Após copiar a caixa de diálogo já existente vou alterar o título e a mensagem e excluir o evento quitGame.

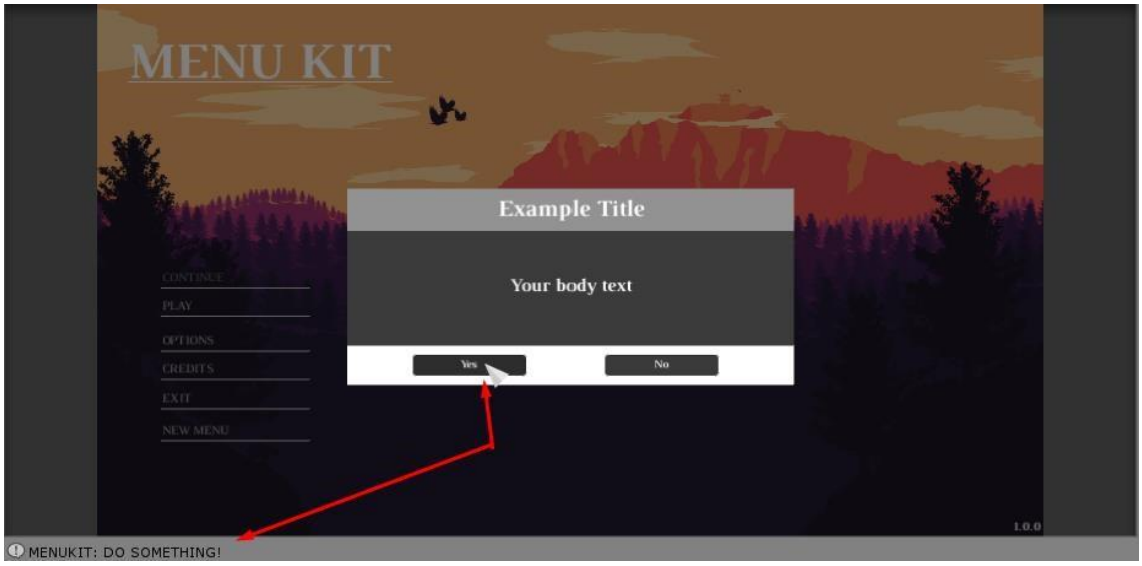
Lembra do nosso menu criado vamos alterar o evento de click, vou adicionar ele mesmo no event e selecionar a função Menu > EnableObject e vou definir o nosso Dialog novo, ficando assim:



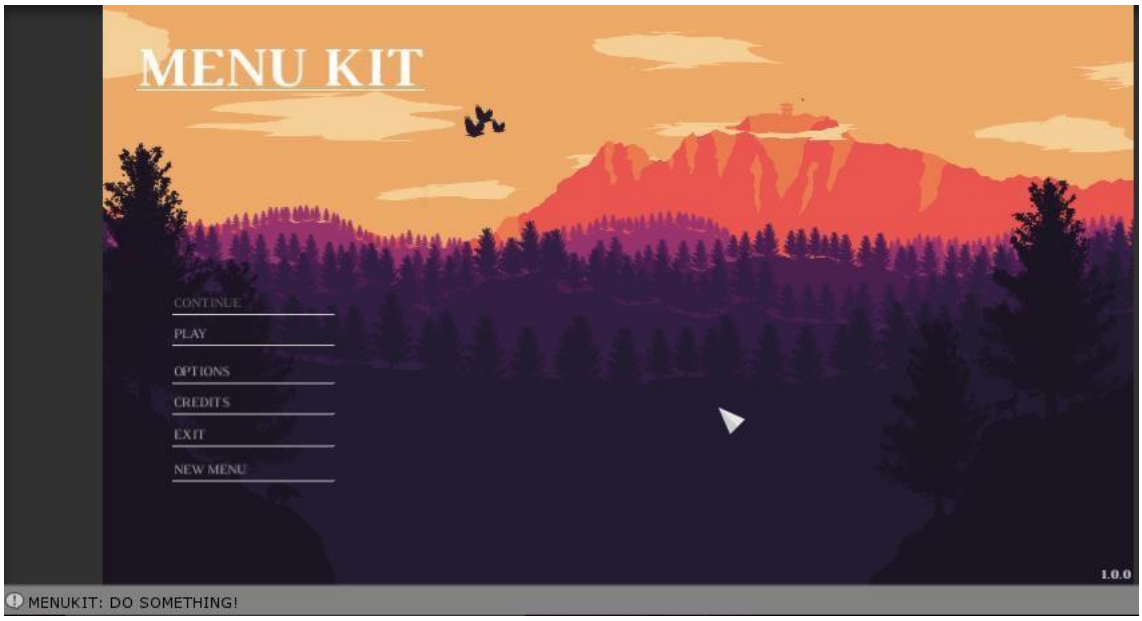
A nossa nova caixa de diálogo está assim:



O método exampleMethod mostra apenas uma mensagem no console, após tudo configurado você pode iniciar o seu projeto, ao clicar no menu novo veja o que acontece: Gif(Pastas)



Ao clicar no yes temos uma mensagem no console, clicando em no:



A caixa de diálogo fecha.

Bem simples também, com isso terminamos o passeio na tela principal. Você pode adicionar o tanto de funções que quiser no Events.

Modificando a Tela de Créditos

Para começar ao mexer em outra tela eu recomendo você desativar as outras, nesse caso desativei a

MainScreen(que estava ativa) no inspetor e assim fica mais limpo/fácil/prático de trabalhar, vamos lá.

A CreditsScreen se encontra MenuKit > Canvas > MenuKit_Canvas > CreditsScreen, para alterar o título vamos objeto title_c e alteramos o TEXT, coloquei como “MENU KIT2”.

Na hierarquia temos Subtitle_structure1 e 2, são dois exemplos muito utilizado nos créditos, fique à vontade para alterar, no meu caso vou usar eles de exemplo e vou desativar o objeto wilgner_credits.

Vou fazer uma peque alteração na animação dele, por padrão vai vir a animação direção, vou posicionar a CreditsScreen nova no centro:

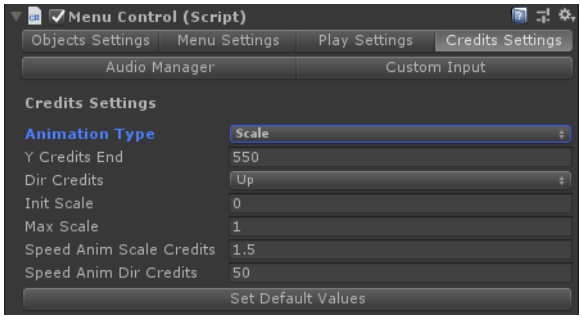
Dessa forma e agora vamos no MenuControl, em

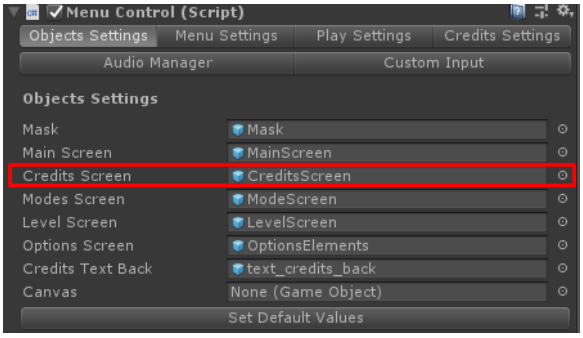
Objects Settings vou colocar a minha nova

CreditsScreen, vou desativar o Anim Direction Credits e ativar Anim Credits Scale ficando dessa forma:

Você pode definir a

Escala Inicial, Escala Máxima e a velocidade da escala para fazer a animação de escala (Já vem configurada).





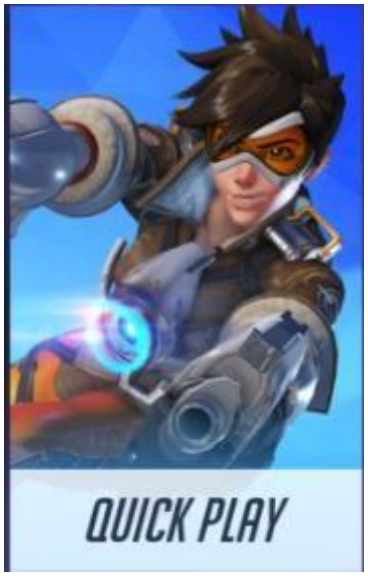
Você pode iniciar o projeto ou desativar a CreditsScreen no inspetor (todas telas são desativadas ao iniciar, menos a MainScreen).

Simples.

Modificando a ModeScreen

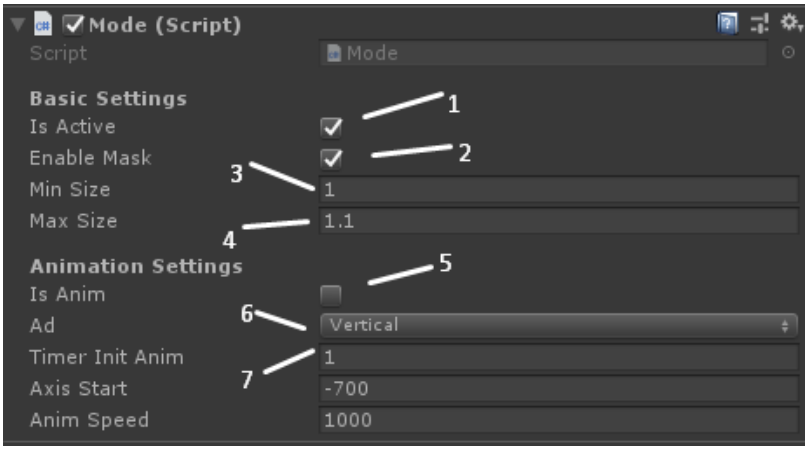
Novamente, desativa as outras telas e ativa apenas a ModeScreen para trabalhar melhor sem as outras atrapalharem. ModeScreen se encontra em MenuKit > Canvas > MeniKit_Canvas > ModeScreen

Novamente temos a BasicSettings, você pode alterar o título e a versão. Vou copiar a mode3(nomear mode4) e posicionar como eu quero, não vou alterar nenhuma informação nela, vamos para a Modes, em background_mode podemos alterar a imagem de fundo do nosso fundo, vou deixar padrão, um bom exemplo é do jogo Overwatch, podemos ver:



Para alterar o título é simples, vamos no title_mode_text e mudamos o componente TEXT no inspetor a mesma coisa no description_text, o objeto maskDisable é caso o modo está indisponivel a maskDisable é ativada e deixa o modo escuro.

Lembre-se de verificar se todas as mascara estão na posição do modo correto que elas pertencem.



1. Caso ativa significa que o modo está ativo e pode ser interativo.
2. Ativar a máscara é caso ele esteja desativado vai haver um escurecimento no modo.
3. Escala padrão
4. Escala quando o mouse estiver em cima
5. Animação inicial
6. Direção da animação
7. Tempo para iniciar a animação (estou usando 0.5 de diferença de cada modo)

Vou alterar a animação do mode4 para 1.5. No EventTrigger em Pointer Click vou alterar o playmode3 para playmode4.

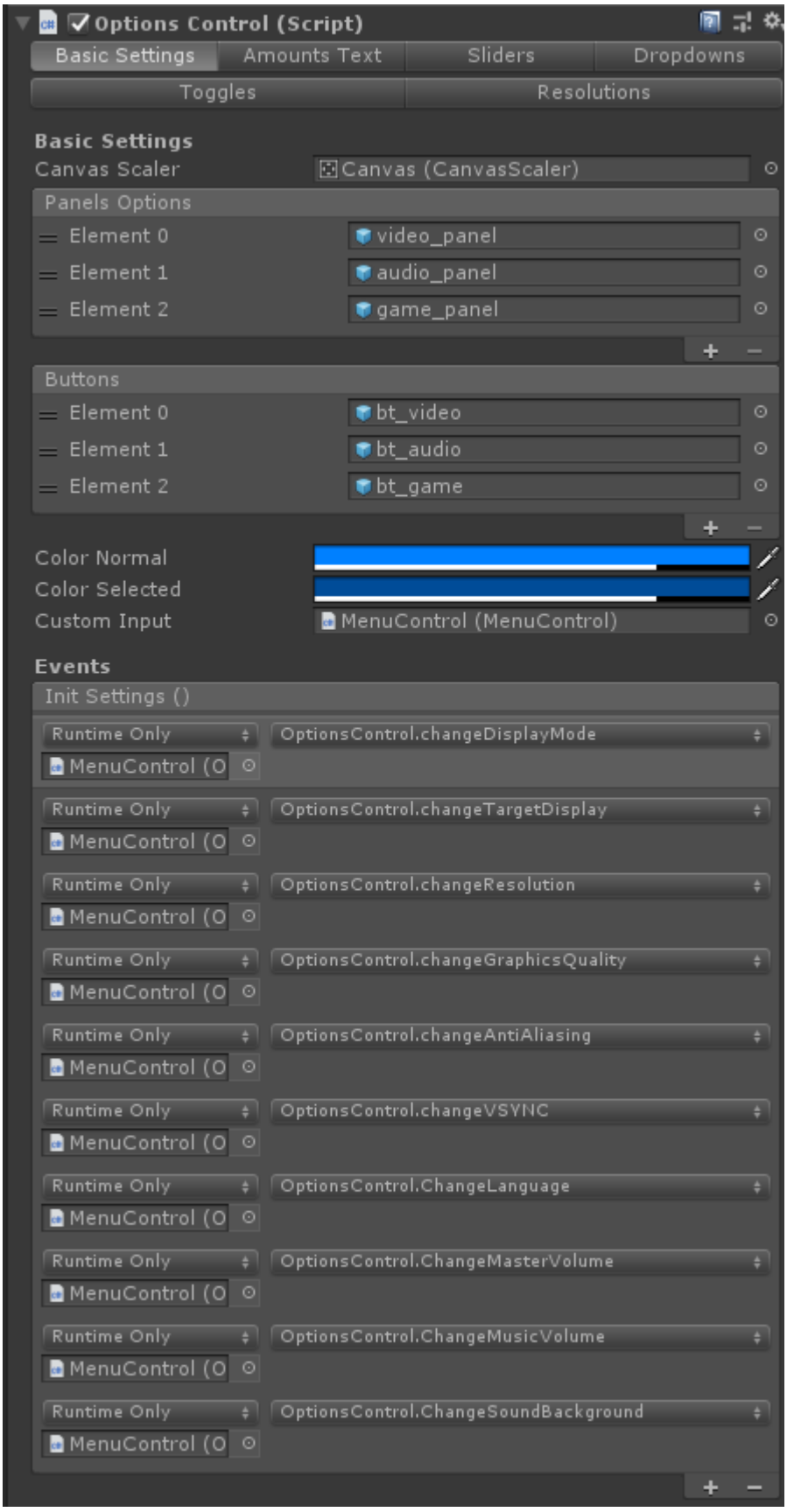
Vou no mode2 e alterar o isActive para falso.

Vamos configuar o MenuControl:

Caso você tenha outra tela de modos e não esteja alterando a do prefab não esqueça de definir a nova no Objects Settings assim como fizemos com a Credits Screen.
Em Play Settings desativa a No Modes.

Pronto, teste o projeto e veja o que acontece:

Tela de Opções



Canvas Scaler: Canvas scaler definido no objeto Canvas

Panels Options: São todos painéis de menu opções (video, audio and game) e **Buttons** são os botões desses respectivos painéis.

Color nomal: É a cor padrão (sem estar selecionado) do botão

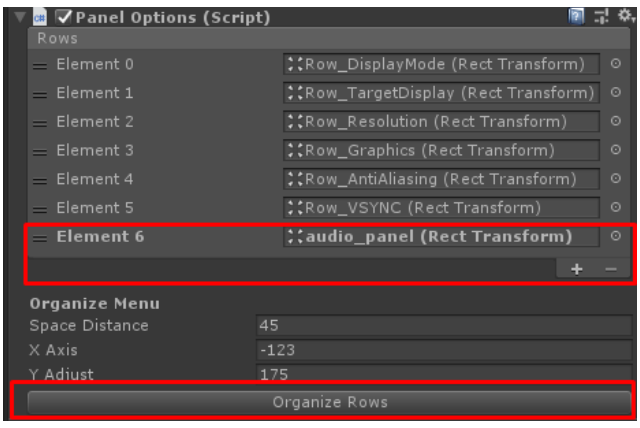
Color Selected: É a cor de quando o painel do respectivo botão está selecionado.

Events: Todos eventos iniciados junto com o menu, por exemplo **changeResolution**, esse evento altera a resolução do menu de acordo com o **json** ao iniciar o menu. Você pode adicionar ou remover ações para ser executado no início.
(Sugestão enviada no meu email)

Agora vamos criar um toggles(checkbox) de exemplo e aprender a salvar no JSON.

Você pode copiar um Toggle pronto ou criar a partir da aba.

Você pode organizar no Painel adicionando um novo elemento e clicando em “Organize Rows”, isso vale para todos os painéis



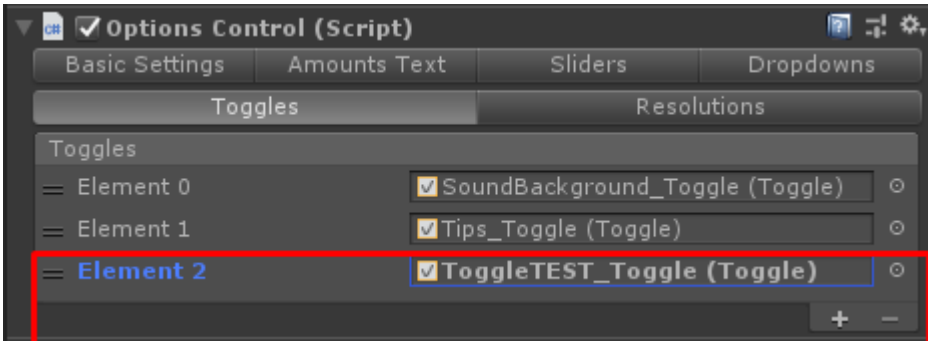
Lembre-se de alterar os nomes para ficar organizado, vamos abrir o Script > GameConfig, isso é a classe onde o json é criado/salvo/armazenado, em //VIDEO abaixo do VSYNC vamos criar uma nova variável:

public bool toggleTest;

Tipo bool porque é esse o valor que o toggle armazena.

```
public class GameConfig {

    // VIDEO \\
    public int displayMode;
    public int targetDisplay;
    public int resulationId;
    public int graphicsQuality;
    public int antialiasing;
    public int vsync;
    public bool toggleTest;
```

Agora vamos no objeto MenuControl (OptionsControl.cs) em toggles vamos adicionar o toggle que criamos:



Agora vamos no Script > OptionsConfig e vamos no método saveGameConfig. Em //Graphics vamos digitar: gameConfig.toggleTest = toggles [2].isOn;

O 2 é a posição dele no vetor, no caso 2 (Element 2), agora ele já está salvando o valor no arquivo JSON, agora vamos fazer ele ao iniciar de novo o jogo carregar a opção selecionada. Vamos no método setValues em //Graphics vamos digitar:

toggles [2].isOn = _gameConfig.toggleTest;

Com isso, o valor dele é definido de acordo com o valor no nosso JSON.
Para mudar algo no jogo ao clicar em Aplicar você deve criar um método de alteração, trabalhar o valor do toggle(exemplo) dentro e chamar o método no changeSettingsGame().

Com apenas isso você já tem uma opção nova na OptionsScreen funcionando e salvando, carregando apartir de um arquivo JSON.

Você pode recuperar esses valores em outras cenas lendo o arquivo json (exemplo na outra scene, YOURGAME).

Controle de Idiomas

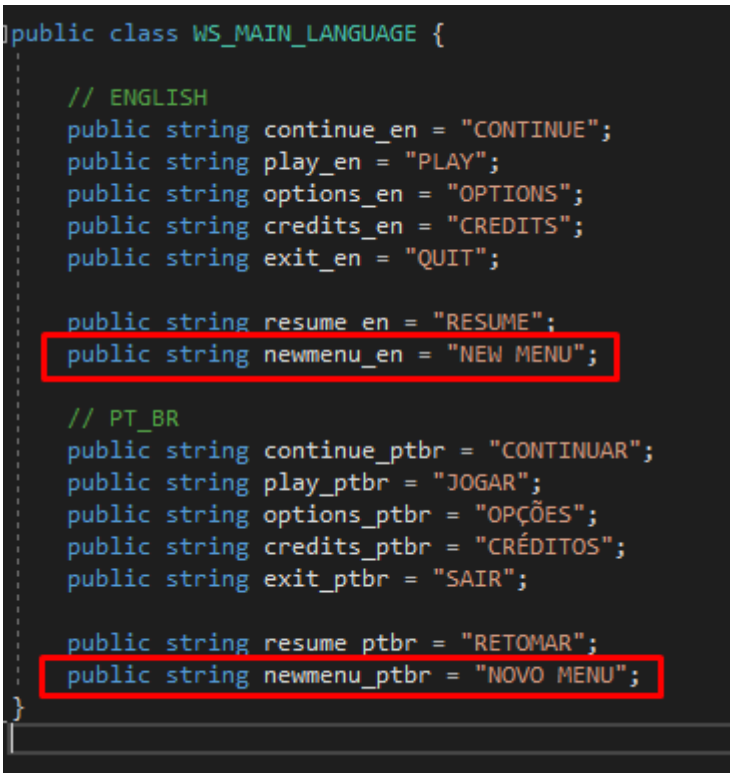
A nova versão do MMK permite múltiplos idiomas.
Vamos começar. Anteriormente ensinei a criar um novo item de menu, agora vamos traduzir.
Primeiro os item de menus são referenciados Menu Settings(MenuControl.cs) e não no LanguageControl.cs.
Nós já referenciamos ele na criação.

Vamos na pasta **Scripts** > **Json Class** > **Languages** > **WS_MENUKIT**
WS_MAIN_LANGUAGE.cs > Esse arquivo contém as traduções dos item de menu (CONTINUE, PLAY, OPTIONS, etc).
WS_OPTIONS_LANGUAGE.cs > Esse arquivo contém as traduções para todos elementos presentes na tela de opções

Ainda estou desenvolvendo, mas o fundamental já está pronto, por exemplo falta traduções dentro dos dropdowns.

Como criamos um item de menu vamos utilizar o **WS_MAIN_LANGUAGE.cs** como explicado acima.

Após abri-lo vamos inserir duas variáveis para representar o texto em inglês e em português (exemplo, você pode adicionar outro idioma), vai ficar assim:



```
public class WS_MAIN_LANGUAGE {

    // ENGLISH
    public string continue_en = "CONTINUE";
    public string play_en = "PLAY";
    public string options_en = "OPTIONS";
    public string credits_en = "CREDITS";
    public string exit_en = "QUIT";

    public string resume_en = "RESUME";
    public string newmenu_en = "NEW MENU";

    // PT_BR
    public string continue_ptbr = "CONTINUAR";
    public string play_ptbr = "JOGAR";
    public string options_ptbr = "OPÇÕES";
    public string credits_ptbr = "CRÉDITOS";
    public string exit_ptbr = "SAIR";

    public string resume_ptbr = "RETOMAR";
    public string newmenu_ptbr = "NOVO MENU";
}
```

Após isso salve e vamos abrir o LanguageControl (pasta scripts ou pode abrir a partir do objeto MenuControl com o botão direito e edit).

Esse script serve para verificar qual idioma está selecionado e a partir disso ele atualiza o texto conforme o nosso banco de linguagens (Os dois scripts que contém os idiomas).

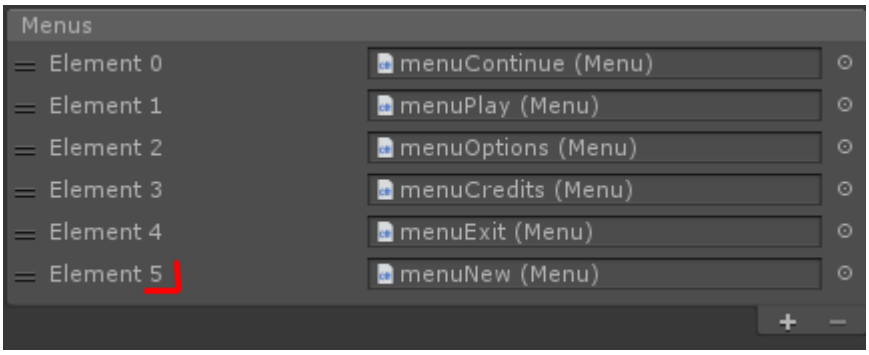
Vamos encontrar duas funções:

SetMainLanguage: Essa função serve para atualizar os textos na tela principal.

SetOptionsLanguage: Atualiza os textos das opções

Como esse novo item de menu está no menu principal e não o de pause vamos colocar dentro do IF inGame (ou isPause).

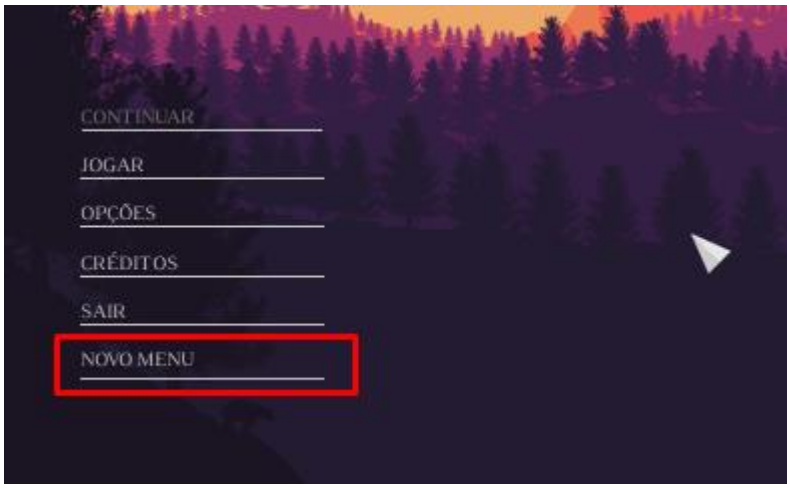
Vamos referenciar o index do menu novo que criamos (no meu caso é o 5)



Referenciar o script em que a tradução desse menu e a variável que criamos acima para referenciar ele, ficando assim:

```
// MAIN MENU
public void SetMainLanguage(){
    if (_optionsControl != null && _optionsControl._gameConfig != null && ws_main_language != nul
        if (_optionsControl._gameConfig.language == 0) { // ENGLISH
            //Debug.Log ("Change To English!");
            if (_menuControl.inGame == false)
            {
                main_titles[0].text = ws_main_language.play_en;
                main_titles[1].text = ws_main_language.options_en;
                _menuControl.menu[0].SetText(ws_main_language.continue_en);
                _menuControl.menu[1].SetText(ws_main_language.play_en);
                _menuControl.menu[2].SetText(ws_main_language.options_en);
                _menuControl.menu[3].SetText(ws_main_language.credits_en);
                _menuControl.menu[4].SetText(ws_main_language.exit_en);

                _menuControl.menu[5].SetText(ws_main_language.newmenu_en);
            }
            else
            {
                _menuControl.menu[0].SetText(ws_main_language.resume_en);
                _menuControl.menu[1].SetText(ws_main_language.options_en);
                _menuControl.menu[2].SetText(ws_main_language.exit_en);
            }
        } else if (_optionsControl._gameConfig.language == 1) { // PTBR
            //Debug.Log ("Change To PTBR");
            if (_menuControl.inGame == false) // not Pause Menu
            {
                main_titles[0].text = ws_main_language.play_ptbr;
                main_titles[1].text = ws_main_language.options_ptbr;
                _menuControl.menu[0].SetText(ws_main_language.continue_ptbr);
                _menuControl.menu[1].SetText(ws_main_language.play_ptbr);
                _menuControl.menu[2].SetText(ws_main_language.options_ptbr);
                _menuControl.menu[3].SetText(ws_main_language.credits_ptbr);
                _menuControl.menu[4].SetText(ws_main_language.exit_ptbr);

                _menuControl.menu[5].SetText(ws_main_language.newmenu_ptbr);
            }
            else // In Pause Menu
            {
                _menuControl.menu[0].SetText(ws_main_language.resume_ptbr);
                _menuControl.menu[1].SetText(ws_main_language.options_ptbr);
                _menuControl.menu[2].SetText(ws_main_language.exit_ptbr);
            }

        }
    }
}// END
```

Após salvar já está pronto (teste).



Vamos traduzir agora o toggle que criamos. Seguimos quase que os mesmos passos.

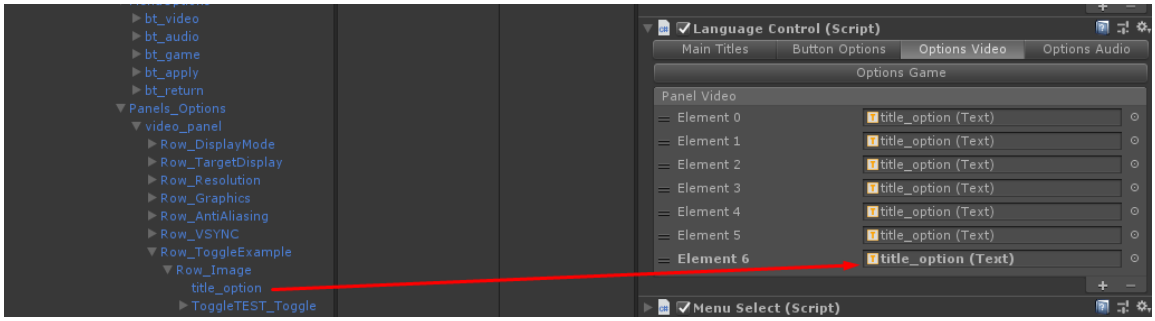
Agora vamos usar o: **WS_OPTIONS_LANGUAGE.cs**

O toggle se encontra no painel vídeo, então:

```
public class WS_OPTIONS_LANGUAGE {

    //************************************************** ENGLISH **
    public string video_en = "VIDEO";
    public string audio_en = "AUDIO";
    public string game_en = "GAME";
    public string apply_en = "APPLY";
    public string return_en = "RETURN";

        // VIDEO
    public string displayMode_en = "DISPLAY MODE";
    public string targetDisplay_en = "TARGET DISPLAY";
    public string resolution_en = "RESOLUTION";
    public string graphicsQuality_en = "GRAPHICS QUALITY";
    public string antialiasing_en = "ANTIALIASING";
    public string vsync_en = "VSYNC";
    public string toggleExample_en = "Toggle Example";
```

```
//************************************************** PT_BR ****
public string video_ptbr = "VÍDEO";
public string audio_ptbr = "SOM";
public string game_ptbr = "JOGO";
public string apply_ptbr = "APLICAR";
public string return_ptbr = "VOLTAR";

    // VIDEO
public string displayMode_ptbr = "MODO DE EXIBIÇÃO";
public string targetDisplay_ptbr = "MONITOR";
public string resolution_ptbr = "RESOLUÇÃO";
public string graphicsQuality_ptbr = "QUALIDADE GRÁFICAS";
public string antialiasing_ptbr = "ANTI-SERRILHAMENTO";
public string vsync_ptbr = "VSYNC";
public string toggleExample_ptbr = "Examplo de Alternância";

// AUDIO
```

Já temos o banco de idiomas pronto, agora precisamos referenciar o texto do toggle para o MMK saber que ele existe e para podemos utilizar nos códigos.



Vamos referenciar a nossa variável nova (idiomas) para atualizar o texto corretamente:

LanguageControl.cs e o método **SetOptionsLanguage()** dessa vez.

Lembrando que o index dele é o 6 como a imagem acima, então:

```
// OPTIONS LANGUAGE
public void SetOptionsLanguage(){
    if (_optionsControl != null && _optionsControl._gameConfig != null && ws_main_language != null) {
        if (_optionsControl._gameConfig.language == 0) { // ENGLISH
            // BUTTONS
            buttons_options[0].text = ws_options_language.video_en;
            buttons_options[1].text = ws_options_language.audio_en;
            buttons_options[2].text = ws_options_language.game_en;
            buttons_options[3].text = ws_options_language.apply_en;
            buttons_options[4].text = ws_options_language.return_en;

            // VIDEO
            options_video[0].text = ws_options_language.displayMode_en;
            options_video[1].text = ws_options_language.targetDisplay_en;
            options_video[2].text = ws_options_language.resolution_en;
            options_video[3].text = ws_options_language.graphicsQuality_en;
            options_video[4].text = ws_options_language.antialiasing_en;
            options_video[5].text = ws_options_language.vsync_en;

            options_video[6].text = ws_options_language.toggleExample_en;

            // AUDIO
```

E agora o outro idioma (no caso pt_br):

```
        } else if (_optionsControl._gameConfig.language == 1) { // PTBR
            // BUTTONS
            buttons_options[0].text = ws_options_language.video_ptbr;
            buttons_options[1].text = ws_options_language.audio_ptbr;
            buttons_options[2].text = ws_options_language.game_ptbr;
            buttons_options[3].text = ws_options_language.apply_ptbr;
            buttons_options[4].text = ws_options_language.return_ptbr;

            // VIDEO
            options_video[0].text = ws_options_language.displayMode_ptbr;
            options_video[1].text = ws_options_language.targetDisplay_ptbr;
            options_video[2].text = ws_options_language.resolution_ptbr;
            options_video[3].text = ws_options_language.graphicsQuality_ptbr;
            options_video[4].text = ws_options_language.antialiasing_ptbr;
            options_video[5].text = ws_options_language.vsync_ptbr;

            options_video[6].text = ws_options_language.toggleExample_ptbr;

            // AUDIO
```

Você pode referenciar o **GameConfig.language** no script **OptionsControl** para poder traduzir outras partes do seu jogo e para novos idiomas.

Por favor deixar uma review na Assets Store
Isso dá um feedback para mim sobre o que adicionar/melhorar e outros.
Obrigado.

F.A.Q

As perguntas mais frequentes no youtube, fórum, email ou na Unity Assets Store vão aparecer aqui.

Credits & Contact

Esse pacote foi desenvolvido por Wilgner Fábio e é apenas um pacote de algo maior.

Agradecimentos especiais a Paulo Camacan por se oferecer e fazer o script UITransition e a vocês por comprarem e me dar suporte <3.

Luthier Reg pela Fonte.

wilgner.fabio@gmail.com

ou

wilgner.fabio2@gmail.com

https://forum.unity.com/threads/released-main-menu-kit-simple-flat-and-complete.518549/