

Titel: Commandsets Untermenüs (z. B. In Festungen)

Typ: Coding

Programm: Final Big

Ich begrüße euch zu einem neuem Stylize-Tutorial, ich werde euch jetzt das Erstellen von Untermenüs in Commandsets erklären, und auch wie man solche erweitert und dabei eventuelle Fehler vermeidet.

Die Commandbuttons und ein eigenes Commandset bespreche ich auf Seite 2 und 3, ebenfalls auf Seite 3 etwas Bugfixing-Hilfe.

Dazu benötigen wir Final Big und extrahieren damit aus der ini.big die Dateien: „Commandset.ini“ und „Commandbutton.ini“, bzw. editieren diese in einer bereits vorhandenen Mod-Big.

Hier zuerst ein Beispiel einer Commandset mit Untermenüs, das Commandset der Elben-Festung (in rot ein paar Anmerkungen von mir):

```

CommandSet ElvenFortressCommandSet
    InitialVisible = 6 ;Diese nette Zeile gibt an, dass im Hauptmenü die
Einträge 1 - 6 angezeigt werden, alle anderen werden nicht angezeigt
    //Main Menu
    1     = Command_ConstructElvenPorter  ;normaler Unit-Baubefehl
    2     = Command_SelectRevivablesElvenFortress  ;Button zum Hero-Menu
    3     = Command_SelectUpgradesElvenFortress  ;Button zum Upgrade-Menu
    4     = Command_PurchaseNewEagle  ;Normaler Unit Baubefehl
    6     = Command_Sell

        //Improvements  ; Hier sieht man das Untermenü für die Upgrades
    7     = Command_PurchaseUpgradeElvenFortressEnchantedAnvil
    8     = Command_PurchaseUpgradeElvenFortressBlessedMist
    9     = Command_PurchaseUpgradeElvenFortressCrystalMoat
    10    = Command_PurchaseUpgradeElvenFortressMysticFountains
    11    = Command_PurchaseUpgradeElvenFortressEncasingVines
    12    = Command_PurchaseUpgradeElvenFortressEaglesNest
    13    = Command_RadialBack

    //Heroes  ;Hier sieht man das Untermenü für die Helden
    14    = Command_RingHeroReviveSlot
    15    = Command_CreateAHeroReviveSlot
    16    = Command_GenericReviveSlot1
    17    = Command_GenericReviveSlot2
    18    = Command_GenericReviveSlot3
    19    = Command_GenericReviveSlot4
    20    = Command_GenericReviveSlot5
    21    = Command_GenericReviveSlot6
    22    = Command_GenericReviveSlot7
    23    = Command_RadialBack

End

```

Die Untermenüs werden im jeweiligem Commandbutton genauer definiert, also deren genaue „Reichweite“.

Wir bauen aus Einfachkeitsgründen ein neues und kleineres Commandset. Ich nenne es jetzt einmal „TestCommandset“.

Ich baue 2 Untermenüs und 2 Buttons ins Hauptmenü und zusätzlich jeweils 3 Buttons in jedes Untermenü.

```
Commandset Testcommandset
```

```
    InitialVisible = 4
```

```
    1 = Command_Bauarbeiter
```

```
    2 = Command_Untermenu1
```

```
    3 = Command_Untermenu2
```

```
    4 = Command_NochnButtonFürFestungsfähigkeit
```

```
    ;Untermenü 1
```

```
    5 = Command_HobbitHeld
```

```
    6 = Command_Menschenheld
```

```
    7 = Command_RadialBack
```

```
    ;Untermenü 2
```

```
    8 = Command_Unitbau
```

```
    9 = Command_Upgrade
```

```
    10 = Command_RadialBack
```

```
End
```

Zur allgemeinen Erklärung:

„Command_RadialBack“ sollte in jedem Untermenü vorhanden sein, da dies der Button ist der ein Menü nach oben geht, also hier ins Hauptmenü zurück.

Ich habe es mir zudem angewöhnt das ich „Command_...“ vor jeden Button schreibe, einfach deshalb weils schöner aussieht.

Die Abtrennung der Untermenüs mit „;Untermenü NR“ ist nicht notwendig macht aber Sinn wenn man im Team arbeitet.

Jetzt bauen wir einen Commandbutton für „Command_Untermenu1“

```
CommandButton Command_Untermenu1
```

```
    Command = PUSH_VISIBLE_COMMAND_RANGE
```

```
    TextLabel = CONTROLBAR:NameDesUntermenüButtons ;"Heldenmenü"
```

```
    ButtonImage = BildnameFürDenButton
```

```
    ButtonBorderStyle = SYSTEM
```

```
    DescriptLabel = CONTROLBAR:BeschreibungDesUntermenüButtons
```

```
    Radial = Yes ;Seeeehr sinnvoll, sonst haben wir das Menü
```

```
bei der Minimap unten drin...
```

```
    CommandRangeStart = 4 ;ACHTUNG!!!! Hier beginnt das zählen nicht wie bei
```

„InitialVisible“ bei 1, sondern bei 0. Da wir bei 5 mit dem Menü anfangen müssen wir also 4 hineinschreiben.

```
    CommandRangeCount = 3 ; Hier geben wir an wie viele Buttons von dem 5ten
```

aus in dem Menü sind, hier zählen wir den 5ten mit, daher 3

```
End
```

Nochmals zur Erklärung:

Die entscheidenden Zeilen sind „CommandRangeStart“ und „CommandRangeCount“.

CommandRangeStart:

Diese Zeile gibt an wo das Menü startet, allerdings nicht von 0 gezählt, sondern von 1. Also in meinem Beispiel Button 5 müsste wie oben so aussehen:

```
CommandRangeStart = 4
```

Man kann also einfach so rechnen:

„Nr. des Startbuttons in Commandset“ - 1 = „CommandRangeStart“

CommandRangeCount:

Diese Zeile gibt an wie weit das Untermenü geht. Der Startbutton wird dabei mitgezählt, also wenn ich bis Button 7 gehen möchte heißt das für mich:

CommandRangeCount = 3

Also als Rechnung:

„Anzahl der Buttons im Untermenü“ - „CommandRangeStart“ = „CommandRangeCount“

Das Untermenü 2 hätte also die Angaben:

CommandRangeStart = 7

CommandRangeCount = 3

Bei puren Heldenmenüs wie in den Festungen ist es sinnvoll dies an das Ende der Commandset zu setzen. Denn so kann man sie leicht erweitern indem man einfach in die Playertemplate einen neuen Helden schreibt.

Die GeneriReviveSlots können ja beliebig viele gesetzt werden (Achtung: es gibt Standardmäßig nur 13 GenericReviveSlots in der Commandbutton.ini, es können aber noch einige erweitert werden sofern Bedarf besteht).

Sollte man in der CommandRangeCount Zeile über die Anzahl der Commandbuttons in der Commandset hinaus kommen, ist das nicht weiter schlimm, da das vom Programm ignoriert wird (Achtung, dies ist nicht weiter schlimm beim letzten Untermenü, doch beim ersten wäre dann auch der Inhalt des Zweiten zu sehen)

Bekannte Fehler und Probleme:

In einem Untermenü wird ein Button angezeigt der dort gar nicht hingehört, was kann der Fehler sein?

Das ist einfach zu lösen:

Ist der Button an erster Stelle, so ist der Wert in der CommandRangeStart-Zeile um 1 zu niedrig, befindet er sich dagegen an Letzter Stelle, so ist (wer hätte es gedacht^^) die CommandRangeCount-Zeile um eins zu hoch....

Nachdem ich einen Button benutzt habe der nach Benutzung verschwindet (z. B. Lvl. 3 Upgrade-Button in der Kaserne), warum ist das so, und was kann ich dagegen tun?

Also dies ist nicht wirklich ein Fehler den man beheben kann, dies tritt auf wenn das Gebäude angewählt bleibt, während das Upgrade beendet wird.

Das sollte nicht mehr sein, wenn das Gebäude einmal abgewählt und dann erneut angewählt wird. Verhindern kann man diesen kurzzeitigen Bug allerdings leider nicht.

Mein Hauptmenü beinhaltet zu viele / zu wenige Buttons, wo liegt das Problem?

Das ist ganz einfach die InitialVisible-Zeile, bei einem Button zu viel? -> die Zahl um eins senken. Zu wenige Buttons? -> Die Zahl höher stellen :-P