K.. / Type / to search

Code Issues Pull requests Actions Projects Security Insights

Commit 63a1877

Browse files

KalElTano authored 47 minutes ago Verified

KET Mod – Fett263 Prop File (27 April 2025)

KET modified version of Fett263 prop file from ProffieOS v7.15 release

main 1 parent 258720e commit 63a1877

Filter files...

saber_fett263_buttons.h

1 file changed +195 -0 lines changed

Search within code

saber_fett263_buttons.h +195

```
@@ -698,6 +698,12 @@ CUSTOM SOUNDS SUPPORTED (add to font to enable):
698 698  #ifndef PROPS_SABER_FETT263_BUTTONS_H
699 699  #define PROPS_SABER_FETT263_BUTTONS_H
700 700
    701 + // BEGIN >> Jamie Customization (Auto-Quote/Define)
    702 + #ifndef AUTO_QUOTE_INTERVAL
    703 + #define AUTO_QUOTE_INTERVAL 0  // Default is disabled
    704 + #endif
    705 + // END >> Jamie Customization (Auto-Quote/Define)
    706 +
701 707  #ifndef MOTION_TIMEOUT
702 708  #define MOTION_TIMEOUT 60 * 15 * 1000
703 709  #endif
```

```
@@ -952,6 +958,9 @@ EFFECT(blstend); // for End Multi-Blast
952  958    EFFECT(push); // for Force Push gesture in Battle Mode
953  959    EFFECT(quote); // quote on force effect
954  960    EFFECT(tr);
     961  + // BEGIN >> Jamie Customization (Auto-Quote/Effect)
     962  + EFFECT(autoquote);  // For playing quotes from font/quotes/ directory
     963  + // END >> Jamie Customization (Auto-Quote/Effect)
955  964    EFFECT2(trloop, trloop);
956  965    #ifdef FETT263_USE_SETTINGS_MENU
957  966    EFFECT(medit); // Edit Mode
@@ -1798,7 +1807,11 @@ SaberFett263Buttons() : PropBase() {}
1798 1807
1799 1808      void Setup() override {
1800 1809        RestoreGestureState();
     1810  +           // BEGIN >> Jamie Customization (Auto-Quote - link to checking for folder 'quotes')
     1811  +           // Check if current font has a quotes folder
     1812  +     CheckQuoteFolder();
1801 1813      }
     1814  +           // END >> Jamie Customization (Auto-Quote - link to checking for folder 'quotes')
1802 1815
1803 1816      bool chdir(const char* dir) override {
1804 1817        bool ret = PropBase::chdir(dir);
@@ -1811,6 +1824,121 @@ SaberFett263Buttons() : PropBase() {}
1811 1824        return ret;
1812 1825      }
1813 1826
     1827  +   // BEGIN >> Jamie Customization (Auto-Quote/Methods)
     1828  + void CheckQuoteFolder() {
     1829  +   // Check if current font has a quotes folder
     1830  +   LOCK_SD(true);
     1831  +   auto_quotes_enabled_ = false;
     1832  +
     1833  +   // Free previous quote order array if it exists
```

```
1834 +   if (quote_order_) {
1835 +     free(quote_order_);
1836 +     quote_order_ = nullptr;
1837 +   }
1838 +
1839 +   num_quotes_ = 0;
1840 +   current_quote_index_ = 0;
1841 +
1842 +   for (const char* dir = current_directory; dir; dir =
       next_current_directory(dir)) {
1843 +     PathHelper path(dir, "quotes");
1844 +     if (LSFS::Exists(path)) {
1845 +       // Found quotes folder in this font
1846 +       auto_quotes_enabled_ = true;
1847 +
1848 +       // Count quotes in the folder
1849 +       LSFS::Iterator it(path);
1850 +       for (; it; ++it) {
1851 +         if (endswith(".wav", it.name())) {
1852 +           num_quotes_++;
1853 +         }
1854 +       }
1855 +
1856 +       if (num_quotes_ > 0) {
1857 +         // Create and initialize the random order array
1858 +         quote_order_ = (int*)malloc(num_quotes_ * sizeof(int));
1859 +         if (quote_order_) {
1860 +           // Initialize array with sequential numbers and then shuffle
1861 +           RandomizeQuotes();
1862 +         }
1863 +       }
1864 +
1865 +       break;
1866 +     }
1867 +   }
1868 +   LOCK_SD(false);
1869 + }
```

```
1870 +
1871 + // Add this method to randomize the quote order
1872 + void RandomizeQuotes() {
1873 +   if (!quote_order_ || num_quotes_ <= 1) return;
1874 +
1875 +   // Initialize with sequential numbers
1876 +   for (int i = 0; i < num_quotes_; i++) {
1877 +     quote_order_[i] = i;
1878 +   }
1879 +
1880 +   // Seed the random number generator
1881 +   srand(millis());
1882 +
1883 +   // Fisher-Yates shuffle
1884 +   for (int i = num_quotes_ - 1; i > 0; i--) {
1885 +     // Generate random index
1886 +     int j = rand() % (i + 1);
1887 +
1888 +     // Swap elements
1889 +     int temp = quote_order_[i];
1890 +     quote_order_[i] = quote_order_[j];
1891 +     quote_order_[j] = temp;
1892 +   }
1893 +
1894 +   current_quote_index_ = 0;
1895 + }
1896 +
1897 + void PlayAutoQuote() {
1898 +   if (!auto_quotes_enabled_ || !SaberBase::IsOn() || num_quotes_ <= 0 ||
      !quote_order_) return;
1899 +
1900 +   // Select current quote index from our randomized array
1901 +   int quote_num = quote_order_[current_quote_index_];
1902 +
1903 +   // Play the selected quote (adding 1 because quote files typically start
      at 01.wav)
1904 +   SFX_autoquote.Select(quote_num + 1);
```

```
1905 +   hybrid_font.PlayCommon(&SFX_autoquote);
1906 +
1907 +   // Advance to next quote for next time
1908 +   current_quote_index_++;
1909 +
1910 +   // If we've gone through all quotes, re-randomize for next cycle
1911 +   if (current_quote_index_ >= num_quotes_) {
1912 +     RandomizeQuotes();
1913 +   }
1914 + }
1915 +
1916 + bool CheckAutoQuote() {
1917 +   // Don't play auto quotes if:
1918 +   // - Feature is disabled
1919 +   // - Blade is off
1920 +   // - In menu mode
1921 +   // - In lockup
1922 +   // - In color change mode
1923 +   if (AUTO_QUOTE_INTERVAL == 0 ||
1924 +       !auto_quotes_enabled_ ||
1925 +       !SaberBase::IsOn() ||
1926 +       menu_ ||
1927 +       SaberBase::Lockup() ||
1928 +       SaberBase::GetColorChangeMode() != SaberBase::COLOR_CHANGE_MODE_NONE)
     {
1929 +     return false;
1930 +   }
1931 +
1932 +   uint32_t now = millis();
1933 +   if (now - last_auto_quote_time_ > AUTO_QUOTE_INTERVAL) {
1934 +     last_auto_quote_time_ = now;
1935 +     return true;
1936 +   }
1937 +
1938 +   return false;
1939 + }
```

```
                1940 +   // END >> Jamie Customization (Auto-Quote/Methods)
                1941 +
1814            1942      // Check Event "Delays" for Edit Mode for Ignition/Retraction/Preon
                     Settings Previews and Choreography Save
1815            1943      void CheckEvent() {
1816            1944        if (next_event_ && !wav_player->isPlaying()) {
                          @@ -1964,6 +2092,12 @@ SaberFett263Buttons() : PropBase() {}
1964            2092    #endif
1965            2093        EditColor();
1966            2094        FreeWavplayerIfPossible();
                2095 + // BEGIN >> Jamie Customization (Auto-Quote/Loop)
                2096 + // Check if it's time to play an auto quote
                2097 + if (CheckAutoQuote()) {
                2098 +   PlayAutoQuote();
                2099 + }
                2100 + // END >> Jamie Customization (Auto-Quote/Loop)
1967            2101        if (SaberBase::IsOn()) {
1968            2102          DetectSwing();
1969            2103    #ifdef FETT263_SAVE_CHOREOGRAPHY
                          @@ -2164,6 +2298,10 @@ SaberFett263Buttons() : PropBase() {}
2164            2298        SaveState(preset);
2165            2299    #endif
2166            2300        SetPreset(preset, true);
                2301 +         // BEGIN >> Jamie Customization (Auto-Quote/SelectPreset)
                2302 +           // Check if new preset's font has a quotes folder
                2303 +   CheckQuoteFolder();
                2304 +         // END >> Jamie Customization (Auto-Quote/SelectPreset)
2167            2305      }
2168            2306
2169            2307      void DetectMenuTurn() {
                          @@ -4567,6 +4705,37 @@ SaberFett263Buttons() : PropBase() {}
4567            4705
4568            4706      bool Parse(const char *cmd, const char* arg) override {
4569            4707        if (PropBase::Parse(cmd, arg)) return true;
```

```
        4708 +         // BEGIN >> Jamie Customization (Auto-Quote/Parse)
        4709 +         if (!strcmp(cmd, "toggle_auto_quotes")) {
        4710 +     if (auto_quotes_enabled_) {
        4711 +       auto_quotes_enabled_ = false;
        4712 +       STDOUT.println("Auto Quotes Disabled");
        4713 +     } else {
        4714 +       CheckQuoteFolder();
        4715 +       if (auto_quotes_enabled_) {
        4716 +         STDOUT.println("Auto Quotes Enabled");
        4717 +       } else {
        4718 +         STDOUT.println("No quotes folder found in current font");
        4719 +       }
        4720 +     }
        4721 +         if (!strcmp(cmd, "randomize_quotes")) {
        4722 +           if (auto_quotes_enabled_ && quote_order_) {
        4723 +         RandomizeQuotes();
        4724 +         STDOUT.println("Quotes order randomized");
        4725 +           } else {
        4726 +       STDOUT.println("Auto Quotes not enabled or no quotes found");
        4727 +         }
        4728 +   return true;
        4729 + }
        4730 +     return true;
        4731 +   }
        4732 +
        4733 +   if (!strcmp(cmd, "get_auto_quotes_status")) {
        4734 +     STDOUT.print("Auto Quotes: ");
        4735 +     STDOUT.println(auto_quotes_enabled_ ? "Enabled" : "Disabled");
        4736 +     return true;
        4737 +   }
        4738 +   // END >> Jamie Customization (Auto-Quote/Parse)
4570    4739         if (!strcmp(cmd, "list_current_tracks")) {
4571    4740           // Tracks must be in: font/tracks/*.wav
4572    4741           LOCK_SD(true);
```

```
@@ -4822,6 +4991,24 @@ SaberFett263Buttons() : PropBase() {}
```

```
4822  4991
4823  4992       bool Event2(enum BUTTON button, EVENT event, uint32_t modifiers) override
                 {
4824  4993         switch (EVENTID(button, event, modifiers)) {
      4994   +
      4995   + // BEGIN >> Jamie Customization (Auto-Quote/Event2)
      4996   +           case EVENTID(BUTTON_POWER, EVENT_FOURTH_HELD_LONG, MODE_ON):
      4997   +       if (!auto_quotes_enabled_) {
      4998   +         CheckQuoteFolder();
      4999   +         if (auto_quotes_enabled_) {
      5000   +           hybrid_font.PlayCommon(&SFX_quote);
      5001   +           // Reset timer to avoid immediate quote after enabling
      5002   +           last_auto_quote_time_ = millis();
      5003   +         } else {
      5004   +           beeper.Beep(0.5, 2000);
      5005   +         }
      5006   +       } else {
      5007   +         auto_quotes_enabled_ = false;
      5008   +         beeper.Beep(0.5, 3000);
      5009   +       }
      5010   +       return true;
      5011   + // END >> Jamie Customization (Auto-Quote/Event2)
4825  5012
4826  5013     #ifdef BLADE_DETECT_PIN
4827  5014           case EVENTID(BUTTON_BLADE_DETECT, EVENT_LATCH_ON, MODE_ANY_BUTTON |
                 MODE_ON):
      ↓↑   @@ -6429,6 +6616,14 @@ SaberFett263Buttons() : PropBase() {}
6429  6616       bool auto_lockup_on_ = false; // Battle Mode Lockup active
6430  6617       bool auto_melt_on_ = false; // Battle Mode Melt/Drag active
6431  6618       bool battle_mode_ = false; // Battle Mode active
      6619   + // BEGIN >> Jamie Customization (Auto-
                 Quote/Class:SaberFett263Buttons/private/bool)
      6620   +   bool auto_quotes_enabled_ = false;
      6621   +   uint32_t last_auto_quote_time_ = 0;
      6622   +   // Variables for random quotes
      6623   +   int num_quotes_ = 0;
```

```
      6624 +   int* quote_order_ = nullptr;
      6625 +   int current_quote_index_ = 0;
      6626 + // END >> Jamie Customization (Auto-
           +   Quote/Class:SaberFett263Buttons/private/bool)
6432  6627     #ifdef FETT263_SPIN_MODE
6433  6628       bool spin_mode_ = false;
6434  6629     #endif
```

Comments 0

Lock conversation

Comment

Unsubscribe You're receiving notifications because you're subscribed to this thread.