Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32
Posted by Oblivion on Mon, 14 Dec 2020 21:48:28 GMT
View Forum Message <> Reply to Message

Below code seems to allow numpad, editpad, cursor keys with modifiers on X11:

```
static struct { KeySym keysym; dword key; } tab[] = {
 { XK_ISO_Left_Tab, K_TAB|K_SHIFT },
 { XK_BackSpace,    K_BACKSPACE },
 { XK_Tab,          K_TAB       },
 { XK_Return,       K_ENTER     },
 { XK_KP_Enter,     K_ENTER     },
 { XK_Escape,       K_ESCAPE    },
 { XK_space,        K_SPACE     },
 { XK_Pause,        K_BREAK     },
 { XK_Scroll_Lock,  K_SCROLL    },
 { XK_Home,         K_HOME      },
 { XK_End,          K_END       },
 { XK_Left,         K_LEFT      },
 { XK_Up,           K_UP        },
 { XK_Right,        K_RIGHT     },
 { XK_Down,         K_DOWN      },
 { XK_Prior,        K_PRIOR     },
 { XK_Next,         K_NEXT      },
 { XK_Page_Up,      K_PAGEUP    },
 { XK_Page_Down,    K_PAGEDOWN  },
 { XK_Insert,       K_INSERT    },
 { XK_KP_Space,     K_SPACE     },
 { XK_KP_Tab,       K_TAB       },
 { XK_KP_Enter,     K_ENTER     },
 { XK_KP_F1,        K_F1        },
 { XK_KP_F2,        K_F2        },
 { XK_KP_F3,        K_F3        },
 { XK_KP_F4,        K_F4        },
 { XK_KP_Home,      K_HOME      },
 { XK_KP_Left,      K_LEFT      },
 { XK_KP_Up,        K_UP        },
 { XK_KP_Right,     K_RIGHT     },
 { XK_KP_Down,      K_DOWN      },
 { XK_KP_Page_Up,   K_PAGEUP    },
 { XK_KP_Page_Down, K_PAGEDOWN  },
 { XK_KP_End,       K_END       },
 { XK_KP_Begin,     K_HOME      },
 { XK_KP_Insert,    K_INSERT    },
 { XK_KP_Delete,    K_DELETE    },
 { XK_KP_Multiply,  K_MULTIPLY  },
```

```
  { XK_KP_Add,       K_ADD       },
  { XK_KP_Separator, K_SEPARATOR },
  { XK_KP_Subtract,  K_SUBTRACT  },
  { XK_KP_Decimal,   K_DECIMAL   },
  { XK_KP_Divide,    K_DIVIDE    },
 };
 for(int i = 0; i < __countof(tab); i++)
  if(tab[i].keysym == keysym) {
   DispatchKey(KEYtoK(tab[i].key)|up, count);
   return;
  }
 if(GetShift() && chr == 0) {
  static dword k[] = { 41, 33, 64, 35, 36, 37, 94, 38, 42, 40 };
  for(int i = 0; i < 10; i++)
   if(keysym == k[i]) {
    DispatchKey(KEYtoK(i + K_0)|up, count);
    return;
   }
 }
#ifndef PLATFORM_OSX11
 if(GetCtrl() || GetAlt()) { // fix Ctrl+Shift+1 etc...
  keysym = decode((int)event->xkey.keycode, 0xa, '1', 0xb, '2', 0xc, '3', 0xd, '4',
                  0xe, '5', 0xf, '6', 0x10, '7', 0x11, '8', 0x12, '9', 0x13, '0',
                  keysym);
 }
#endif
 // DLOG("keysym: " << keysym << " " << (char)keysym);
 keysym &= 0xFFFF;
 if(keysym >= '0' && keysym <= '9' && (chr == 0 || GetCtrl() || GetAlt())) {
  DispatchKey(KEYtoK(keysym - '0' + K_0)|up, count);
  return;
 }
 if(chr >= 1 && chr < 32) {
  DispatchKey(KEYtoK(chr - 1 + K_CTRL_A)|up, count);
  return;
 }
 if(keysym >= 0xff80 && keysym <= 0xffb9 && chr) {
  DispatchKey(KEYtoK(chr)|up, count);
  return;
 }
 if(keysym >= 0xff00 && chr < 128 ||
    (GetCtrl() || GetAlt()) && keysym >= 0x20 && keysym < 0x7f) {
  if(keysym >= 'a' && keysym <= 'z')
   keysym = keysym - 'a' + 'A';
  DispatchKey(KEYtoK(keysym|K_DELTA)|up, count);
  return;
 }
```

```
 if((chr == 32 || chr == 9 || chr == 13) && !pressed)
  DispatchKey(chr|K_KEYUP, count);
 if(chr && pressed) {
  DispatchKey(chr, count);
  for(int ii = 1; ii < wtext.GetLength(); ii++)
   DispatchKey(wtext[ii], count);
 }
}
break;
```

Best regards,
Oblivion
