

Book Series
of
jago{coding}
Learn & Share {Your Code}

www.jagocoding.com

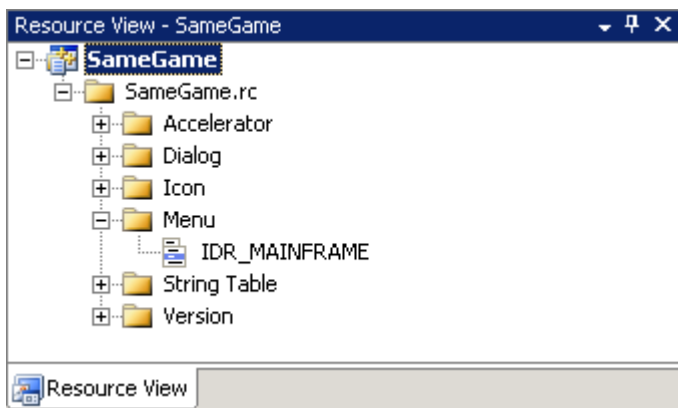
Membuat Game Mencocokkan Objek dengan C++ (Part 4)

Oleh: Sendy PK

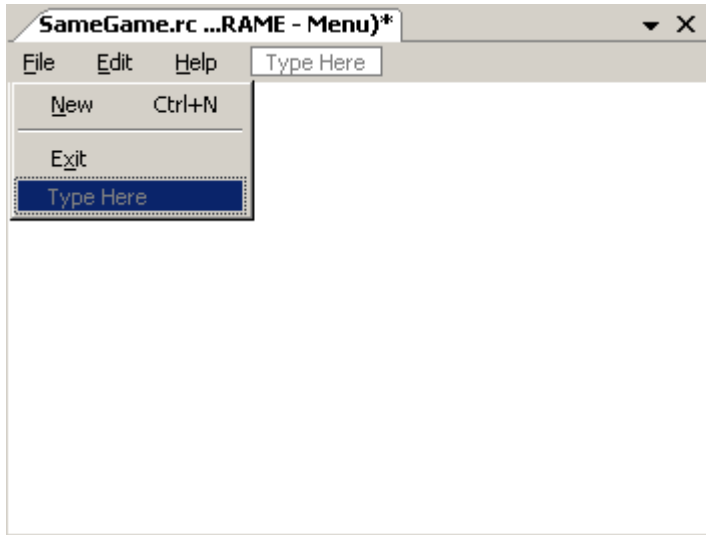
Pada tutorial kali ini kita akan fokus pada menambahkan customizability ke game melalui menu, termasuk menambahkan tingkat kesulitan. Untuk melakukan hal ini, kita akan memperbarui menu yang disertakan dengan MFC Application Wizard, menambahkan perintah baru dan mengatur event handler untuk mereka. Setel...

Pada tutorial kali ini kita akan fokus pada menambahkan customizability ke game melalui menu, termasuk menambahkan tingkat kesulitan. Untuk melakukan hal ini, kita akan memperbarui menu yang disertakan dengan MFC Application Wizard, menambahkan perintah baru dan mengatur event handler untuk mereka. Setelah beberapa latihan, game dengan hanya tiga warna cukup mudah untuk memecahkan sehingga dalam artikel ini kita akan menambahkan tingkat baru kesulitan. Kita dapat melakukan ini dengan hanya menambahkan lebih banyak warna.

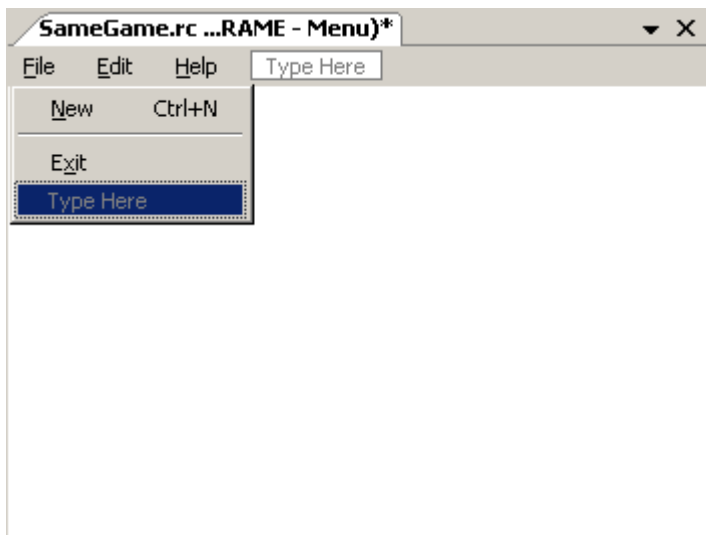
Tambahkan pilihan menu baru Anda dapat memunculkan view sumber daya dari Menu View di Visual Studio dan kemudian, other menu-> Resource View atau shortcut keyboard Ctrl + Shift + E. Lalu akan tampil window seperti gambar di bawah ini.



Buka menu pada menu editor dengan mengklik dua kali pada pilihan IDR_MAINFRAME di bawah Menu. Menu Editor akan memungkinkan kita untuk menambahkan, menghapus and mengedit pilihan menu. Klik pada menu file dan akan drop down menu. Hapus semua pilihan kecuali untuk "New" dan "Exit" seperti yang ditunjukkan pada gambar di bawah. Lakukan ini dengan mengklik pilihan dan menekan tombol delete.



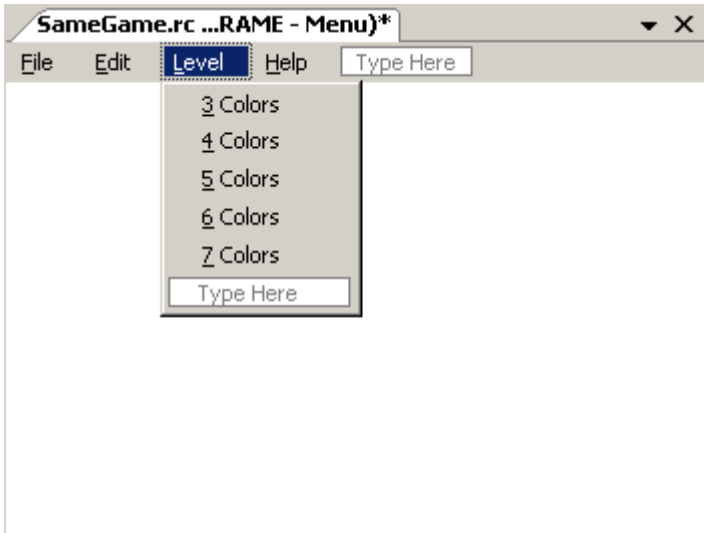
Berikutnya kita akan mengedit menu. Pada tutorial yang terakhir kita akan membahas cara membuat tumpukan Undo / Redo untuk game jadi mari kita tambahkan opsi redo ke menu edit dan hapus semua yang lain kecuali untuk undo. Setelah kita hapus opsi yang tidak diinginkan, klik pada area "Type Here" di menu Edit dan ketik seperti gambar di bawah ini.



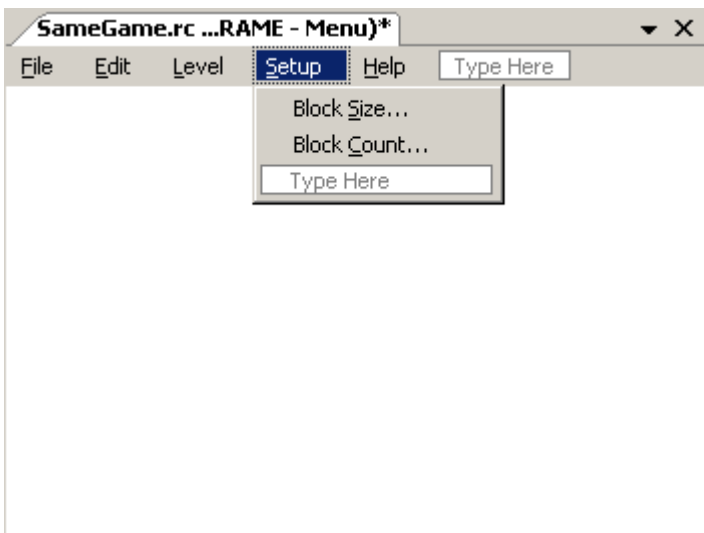
Berikut penjelasan masing-masing bagian dari string "& Redo \ tCtrl + Y". Pertama sinyal ampersand ke menu apa yang hot-key untuk pilihan menu tersebut. Hot-key adalah karakter setelah ampersand, R dalam kasus ini. Ini akan memungkinkan kita untuk menekan Alt + E untuk membuka menu Edit kemudian tekan R untuk memilih opsi Redo. Karakter tab di menu menempatkan tab antara nama pilihan menu dan tombol akselerator. Lihatlah pilihan Undo untuk melihat tab dalam tindakan. Kunci accelerator adalah seperangkat key stroke yang akan menyelesaikan tugas yang sama dengan mengklik pilihan menu. Kita akan menggunakan Windows standar Ctrl + Y untuk redo. Kita akan menambahkan event handler dan kode mendukung dalam tutorial selanjutnya.

Tambahkan menu baru dengan mengklik "Type Here" di atas dalam menu bar. Ketik "& Level" untuk menambahkan menu Tingkat. Setelah menekan Enter kita dapat memindahkan menu di sebelah kiri menu Help dengan hanya mengklik menu dan menyeretnya ke kiri. Sekarang kita akan tambahkan dalam pilihan menu. Kita ingin memungkinkan pengguna untuk memilih tingkat yang berbeda dari tiga sampai tujuh warna. Tambahkan opsi lima menu berikut yang terlihat di bawah. Yang pertama

berisi teks "& 3 Warna" dan seterusnya.

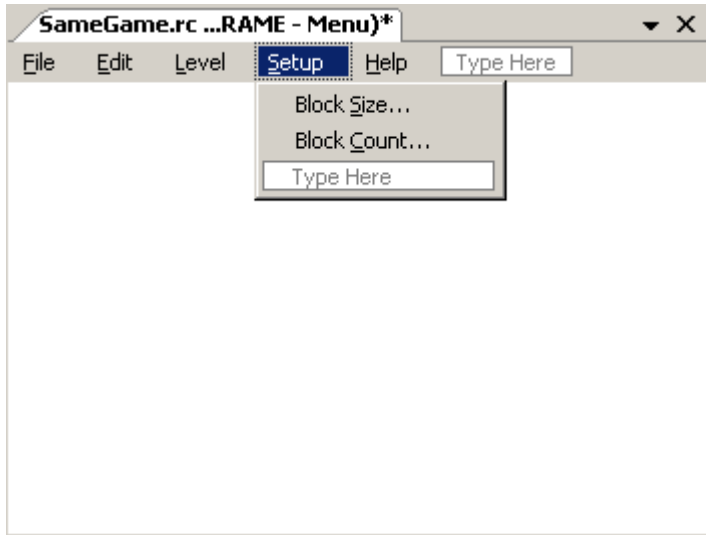


Tambahkan "& Setup" menu dan pindahkan ke kiri menu Help. Kemudian tambahkan "Block & Ukuran ..." dan "Block & Hitung ..." pilihan. Perhatikan tiga periode setelah nama pilihan, yang merupakan standar untuk sistem menu untuk menunjukkan bahwa akan ada jendela lain untuk popup.

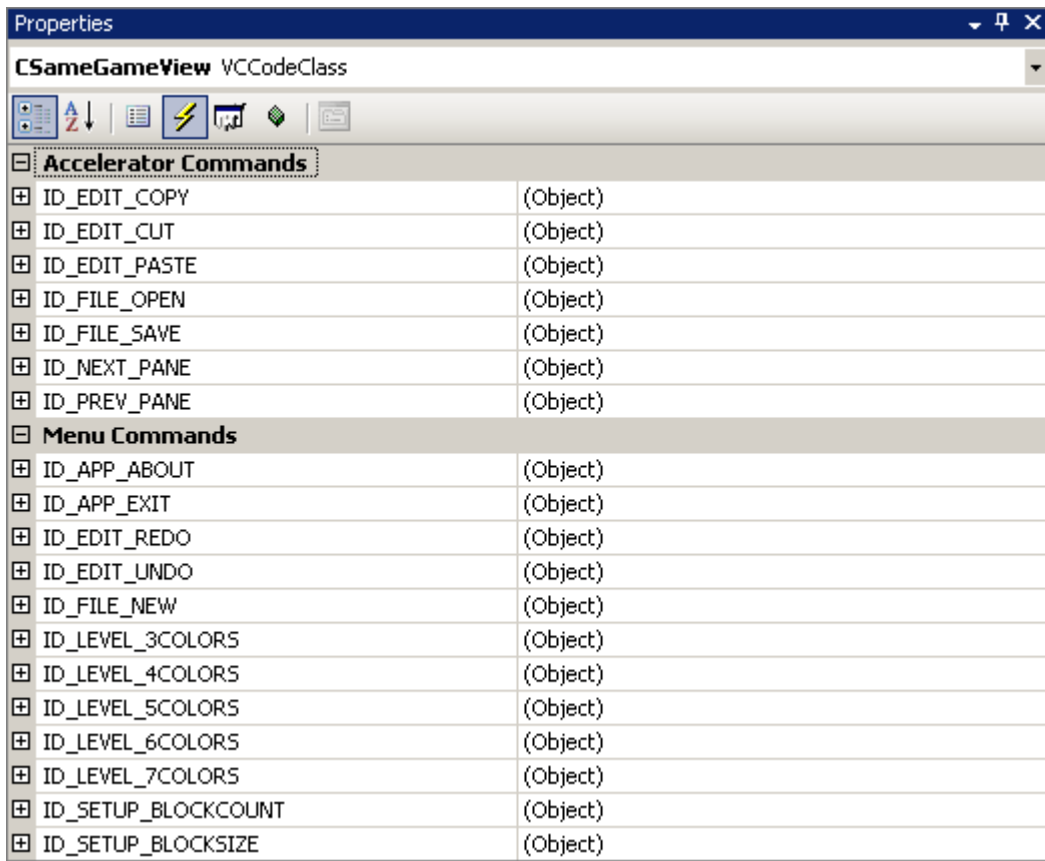


Sekarang kita akan fokus pada menu Level dan Option yang kita akan kita tambahkan. Setelah kita telah menambahkan pilihan menu kita dapat mengkompilasi dan menjalankan game dan melihat bahwa menu telah ditambahkan, tetapi belum dapat berfungsi

Ketika kita mengklik pada pilihan menu program fires event yang menunjukkan menu yang pilihan terpilih. Kita hanya perlu untuk menangkap acara dengan sebuah event handler. Hal ini sangat mirip dengan penanganan event yang kita bahas dan digunakan dalam artikel sebelumnya. Satu-satunya perbedaan adalah di mana kita menemukan aktivitas untuk pilihan menu. Kita mulai dari jendela properti dari CSameGameView dan klik pada "Events" tombol (bolt). Setelah kita klik akan tampil seperti pada gambar berikut ini.



Kita akan melihat pada bagian Menu Command Section daftar semua menu item Entri dalam format ID_ <MENU> _ <OPTION> mana <MENU> <OPTION> adalah teks dari pilihan menu. Expand tanda plus di sebelah opsi ID_LEVEL_3COLORS dan akan terlihat dua pilihan di bawahnya, COMMAND dan UPDATE_COMMAND_UI. Pilihan COMMAND adalah event handler untuk event sebenarnya memilih opsi menu. The UPDATE_COMMAND_UI adalah event kecil yang berfungsi mengubah keadaan pilihan menu. aktifkan / nonaktifkan atau centang / tidak centang pilihan menu. Dalam menu kita, kita akan ceklis tingkat yang telah kita pilih.



Klik panah bawah di dropdown COMMAND dan pilih "Add". Kita ingin melakukan ini untuk UPDATE_COMMAND_UI juga. Pergi ke depan dan melakukannya untuk semua pilihan warna lalu

kita mulai menambahkan kode.

Kita akan mulai dengan game board, pergi ke dokumen dan selesai dengan view. Pada file header SameGameBoard.h tambahkan new data member untuk melacak jumlah warna di bawah variabel m_nRemaining.

```
/* Number of colors */  
int m_nColors;
```

Kita juga akan menambahkan dua fungsi untuk mendapatkan dan menetapkan jumlah warna. Menambahkannya ke bagian umum tepat di bawah fungsi DeleteBlocks.

```
/* Functions to get and set the number of colors */  
int GetNumColors(void) { return m_nColors; }  
void SetNumColors(int nColors)  
{ m_nColors = (nColors >= 3 && nColors <= 7) ? nColors : m_nColors; }
```

Fungsi dari SetNumColors adalah membatasi nilai yang ditetapkan untuk angka antara tiga dan tujuh, seperti pilihan menu kita. Kita tambahkan lebih banyak warna kita perlu memperbarui m_arrColors array untuk membuatnya lebih besar

```
/* List of colors, 0 is background and 1-7 are piece colors */  
static COLORREF m_arrColors[8];
```

Sekarang dalam source file untuk game board kita akan memperbarui beberapa fungsi dan warna array. Kita perlu memperbarui konstruktor untuk menginisialisasi jumlah warna yaitu tiga.

```
COLORREF CSameGameBoard::m_arrColors[8];  
  
CSameGameBoard::CSameGameBoard(void)  
: m_arrBoard(NULL),  
  m_nColumns(15), m_nRows(15),  
  m_nHeight(35), m_nWidth(35),  
  m_nRemaining(0), m_nColors(3)  
{  
  m_arrColors[0] = RGB( 0, 0, 0);  
  m_arrColors[1] = RGB(255, 0, 0);  
  m_arrColors[2] = RGB(255,255, 64);  
  m_arrColors[3] = RGB( 0, 0,255);  
  
  m_arrColors[4] = RGB( 0,255, 0);  
  m_arrColors[5] = RGB( 0,255,255);  
  m_arrColors[6] = RGB(255, 0,128);
```

```

m_arrColors[7] = RGB( 0, 64, 0);

// Create and setup the board
SetupBoard();
}

```

Perubahan lain yang kita butuhkan untuk membuatnya yaitu di SetupBoard tersebut. Sebelumnya, kita telah memasukkan code jumlah warna untuk yaitu 3 di setup. Sekarang, kita akan mengubah nomor tiga untuk m_nColors mana kita mod nomor acak seperti di bawah ini.

```

void CSameGameBoard::SetupBoard(void)
{
    // Create the board if needed
    if(m_arrBoard == NULL)
        CreateBoard();
    // Randomly set each square to a color
    for(int row = 0; row < m_nRows; row++)
        for(int col = 0; col < m_nColumns; col++)
            m_arrBoard[row][col] = (rand() % m_nColors) + 1;
    // Set the number of spaces remaining
    m_nRemaining = m_nRows * m_nColumns;
}

```

Sekarang kita akan beralih ke dokumen. Kita akan menambahkan fungsi sehingga view dapat mengubah jumlah warna. Fungsi The GetNumColors fungsi sederhana yang dapat kita selesaikan dalam file header. Tambahkan baris kode berikut di bagian umum dari dokumen SameGameDoc.h.

```

int GetNumColors()      { return m_board.GetNumColors(); }
void SetNumColors(int nColors);

```

Pada source file kita akan menambahkan implementasi dari fungsi SetNumColors. Di sini kita tidak bisa hanya memanggil fungsi SetNumColors pada game board dengan sendirinya. Kita akan me-reset game board untuk menambahkan warna baru.

```

void CSameGameDoc::SetNumColors(int nColors)
{
    // Set the number of colors first
    m_board.SetNumColors(nColors);
    // Then reset the game board
    m_board.SetupBoard();
}

```

Berikut adalah semua yang kita butuhkan dalam dokumen. Kita belum bisa menggunakan fungsi-fungsi ini. View adalah hal terakhir yang harus diperbarui, Dengan menambahkan event handler kita akan dapat fungsi berikut dimasukkan ke dalam file header.

```
/* Functions to change the difficulty level */
afx_msg void OnLevel3colors();
afx_msg void OnLevel4colors();
afx_msg void OnLevel5colors();
afx_msg void OnLevel6colors();
afx_msg void OnLevel7colors();
/* Functions to update the menu options */
afx_msg void OnUpdateLevel3colors(CCmdUI *pCmdUI);
afx_msg void OnUpdateLevel4colors(CCmdUI *pCmdUI);
afx_msg void OnUpdateLevel5colors(CCmdUI *pCmdUI);
afx_msg void OnUpdateLevel6colors(CCmdUI *pCmdUI);
afx_msg void OnUpdateLevel7colors(CCmdUI *pCmdUI);
```

Kita akan melihat beberapa hal dalam prototipe ini yang belum pernah kita lihat sebelumnya, `afx_msg` menetapkan bahwa fungsi adalah event handler. Fungsi `OnUpdate` menggunakan pointer ke objek `CCmdUI`. Pada source file kita akan melihat beberapa kode lain yang telah ditambahkan.

```
BEGIN_MESSAGE_MAP(CSameGameView, CView)
    ON_WM_LBUTTONDOWN()
    ON_WM_ERASEBKGD()

    ON_COMMAND(ID_LEVEL_3COLORS, &CSameGameView::OnLevel3colors)
    ON_COMMAND(ID_LEVEL_4COLORS, &CSameGameView::OnLevel4colors)
    ON_COMMAND(ID_LEVEL_5COLORS, &CSameGameView::OnLevel5colors)
    ON_COMMAND(ID_LEVEL_6COLORS, &CSameGameView::OnLevel6colors)
    ON_COMMAND(ID_LEVEL_7COLORS, &CSameGameView::OnLevel7colors)
    ON_UPDATE_COMMAND_UI(ID_LEVEL_3COLORS,
        &CSameGameView::OnUpdateLevel3colors)
    ON_UPDATE_COMMAND_UI(ID_LEVEL_4COLORS,
        &CSameGameView::OnUpdateLevel4colors)
    ON_UPDATE_COMMAND_UI(ID_LEVEL_5COLORS,
        &CSameGameView::OnUpdateLevel5colors)
    ON_UPDATE_COMMAND_UI(ID_LEVEL_6COLORS,
        &CSameGameView::OnUpdateLevel6colors)
    ON_UPDATE_COMMAND_UI(ID_LEVEL_7COLORS,
        &CSameGameView::OnUpdateLevel7colors)

END_MESSAGE_MAP()
```

`MESSAGE_MAP` adalah daftar macro C++ yang mengasosiasikan event dengan sebuah event handler. Semua `OnLevel*` fungsi warna akan terlihat seperti fungsi berikut.


```

void CSameGameView::OnLevel3colors()
{
    // First get a pointer to the document
    CSameGameDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if(!pDoc)
        return;
    // Set the number of colors
    pDoc->SetNumColors(3);
    // Force the view to redraw
    Invalidate();
    UpdateWindow();
}

```

Pada semua fungsi view kita akan dapat pointer ke dokumen. Kemudian kita atur jumlah warna pada jumlah di nama fungsi, yaitu OnLevel3colors call SetNumColors (3) dan seterusnya. Terakhir kita akan force view untuk redraw. Ulangi langkah ini untuk semua event untuk pilihan menu. Dengan semua fungsi ini dilakukan, kompilasi dan menguji itu, kita akan melihat jumlah warna berubah 3-4 dan seterusnya. Kemudian kita bisa membuat kode seperti ini:

Dalam SameGameView.h header, kita tambahkan helper ini:

```

void setColorCount(int numColors);

```

Dalam SameGameView.cpp, kita tambahkan kode ini:

```

void CSameGameView::setColorCount(int numColors)
{
    // First get a pointer to the document
    CSameGameDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if(!pDoc)
        return;
    // Set the number of colors
    pDoc->SetNumColors(numColors);
    // Force the view to redraw
    Invalidate();
    UpdateWindow();
}

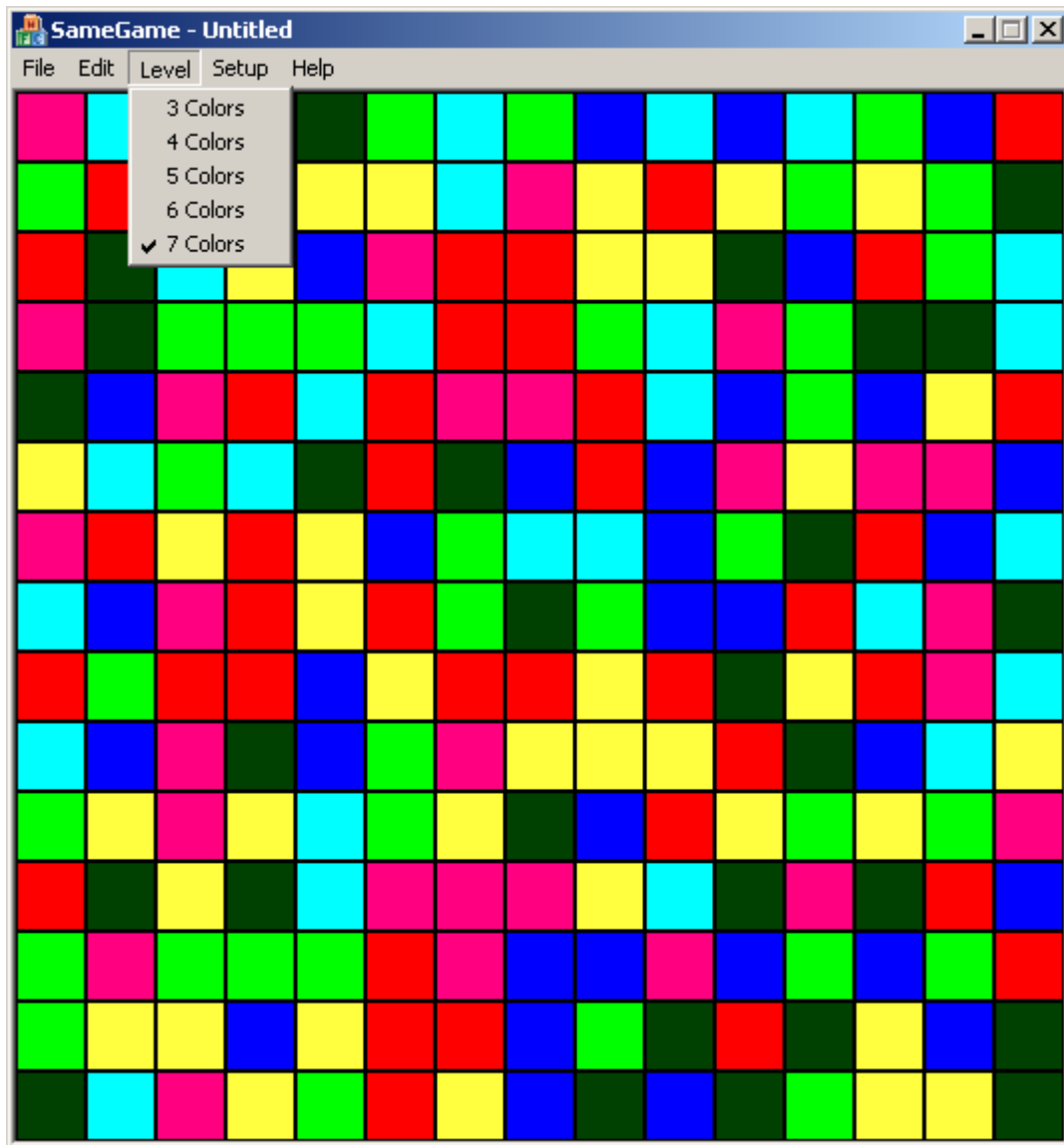
void CSameGameView::OnLevel3colors()
{
    setColorCount(3)
}

```

Sekarang kembali ke final set dari event handlers. ON_UPDATE_COMMAND_UI event dipanggil ketika menu ditarik ke bawah oleh pengguna, satu untuk setiap pilihan menu. Kita akan menggunakan fungsi SetCheck dari objek CCmdUI untuk mengatur dan diset kotak centang di sebelah tingkat. Kita mulai dengan kode boiler-piring kita untuk mendapatkan pointer ke dokumen dan kemudian mengatur cek berdasarkan jumlah warna di board.

```
void CSameGameView::OnUpdateLevel3colors(CCmdUI *pCmdUI)
{
    // First get a pointer to the document
    CSameGameDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if(!pDoc)
        return;
    // Set the check if this is the right level
    pCmdUI->SetCheck(pDoc->GetNumColors() == 3);
}
```

Fungsi ini semua sama kecuali untuk jumlah yang kita bandingkan hasilnya GetNumColors. Kelima fungsi ini disebut ketika menu Tingkat ditampilkan untuk mengatur atau unset tanda centang. Sekarang game akan terlihat seperti ini.



Pada tutorial ini kita telah membuat game kita sedikit lebih menarik dengan menambahkan empat tingkat kesulitan baru. Kita belajar tentang menu editor dan cara setup menu. Kita juga telah setup event handler untuk `ON_COMMAND` dan `ON_COMMAND_UPDATE_UI` jenis event untuk menu. Pada tutorial berikutnya kita akan menambahkan pilihan baru untuk mengatur jumlah dan ukuran blok di game board.

Tentang Penulis



Sendy PK

Saya adalah Programmer yang memiliki impian untuk menguasai dunia
kunjungi situs pribadi saya di www.spk.my.id