

Zadaća 1

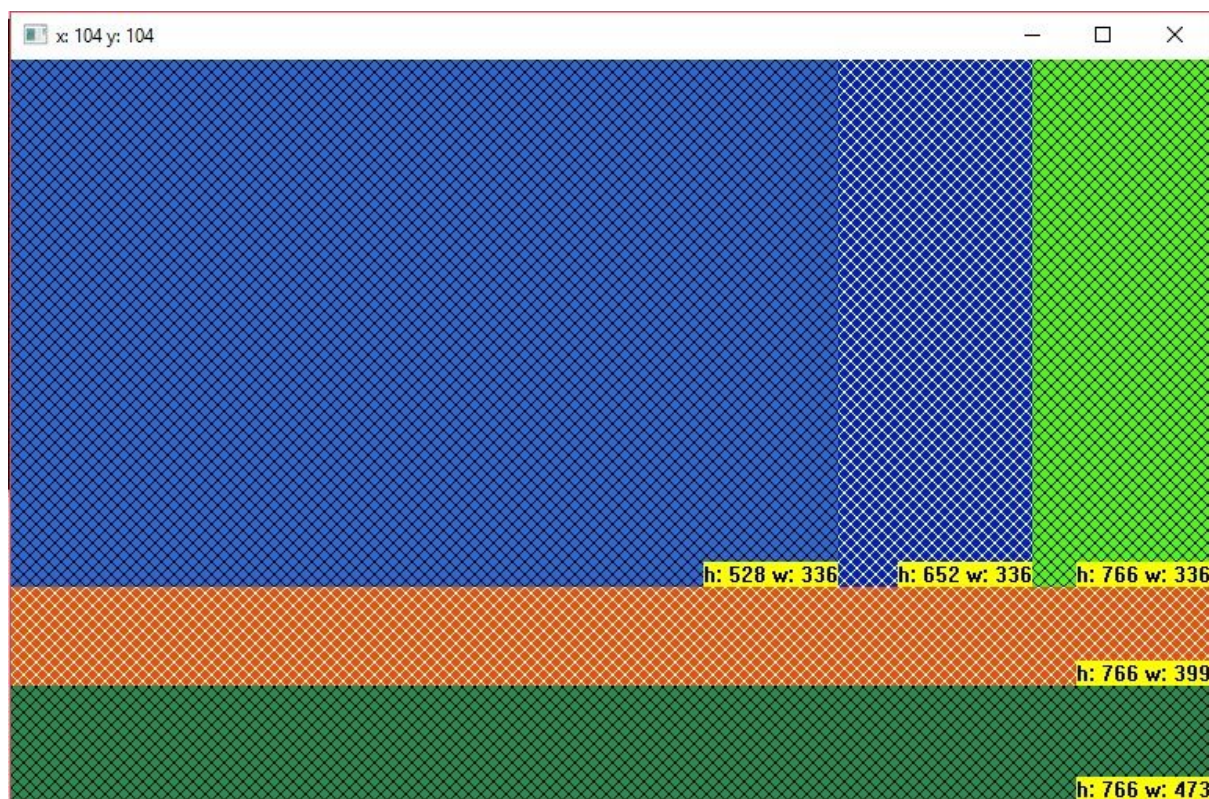
Zadatak 1

Na laboratorijskim vježbama je urađen primjer sa procesiranjem WM_PAINT poruke i sa validiranjem i bojenjem invalidnih područja različitom bojom.

Modifikovati primjer sa vježbi tako da su u svakom invalidnom području date trenutne vrijednosti visine i širine klijentskog područja. Pozadina invalidnog područja je neka random boja. Boja rešetke se naizmjenično mijenja (crna/bijela), a pozadina teksta je žute boje.

Tekst je prikazan u donjem desnom uglu svakog invalidnog područja. Početne dimenzije prozora postaviti na (300,200). U title bar – u prozora, na svaku promjenu pozicije prozora, trebaju biti ispisane trenutne koordinate prozora u formatu: **x: vrijednost, y: vrijednost**.

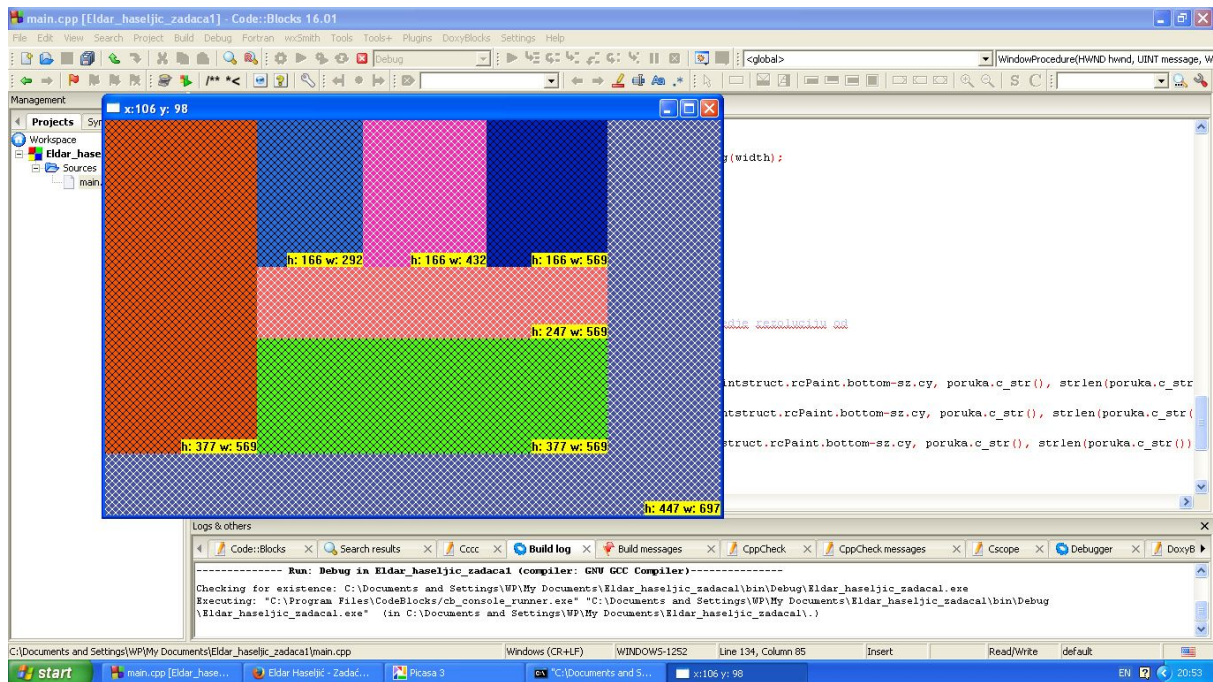
Rezultat izvršenja programa nakon nekoliko operacija povećavanja prozora, treba da bude sličan donjoj slici:



U nastavku donjeg teksta, na odgovarajuća mjesta je potrebno ubaciti sliku dobijenog prozora, definiciju window klase, poziv funkcije *CreateWindow* te definiciju window procedure za odgovarajući prozor.

Pri popunjavanju dokumenta za kod koristiti font Courier New.

Slika prozora (screenshot)



Definicija window klase

```
WNDCLASSEX wincl;          /* Data structure for the windowclass */

/* The Window structure */
wincl.hInstance = hThisInstance;
wincl.lpszClassName = szClassName;
wincl.lpfnWndProc = WindowProcedure;          /* This function is called
by windows */
wincl.style = CS_DBLCLKS;                  /* Catch double-clicks */
wincl.cbSize = sizeof (WNDCLASSEX);

/* Use default icon and mouse-pointer */
wincl.hIcon = LoadIcon (NULL, IDI_APPLICATION);
wincl.hIconSm = LoadIcon (NULL, IDI_APPLICATION);
wincl.hCursor = LoadCursor (NULL, IDC_ARROW);
wincl.lpszMenuName = NULL;                  /* No menu */
wincl.cbClsExtra = 0;          /* No extra bytes after the window class */
wincl.cbWndExtra = 0;          /* structure or the window instance */

/* Use Windows's default colour as the background of the window */
```

```

        wincl.hbrBackground = (HBRUSH) COLOR_BACKGROUND;
/* Register the window class, and if it fails quit the program */
        if (!RegisterClassEx (&wincl))
            return 0;

```

Poziv funkcije CreateWindow

```

/* The class is registered, let's create the program*/
hwnd = CreateWindowEx (
    0,                      /* Extended possibilities for variation */
    szClassName,           /* Classname */
    _T("Code::Blocks Template Windows App"), /* Title Text */
    WS_OVERLAPPEDWINDOW, /* default window */
    104,                   /* Windows decides the position */
    104,                   /* where the window ends up on the screen */
    300,                   /* The programs width */
    200,                   /* and height in pixels */
    HWND_DESKTOP,         /* The window is a child-window to desktop */
    NULL,                 /* No menu */
    hThisInstance,        /* Program Instance handler */
    NULL                  /* No Window Creation data */
);

```

Definicija window procedure

```

/* This function is called by the Windows function DispatchMessage() */
LRESULT CALLBACK WindowProcedure (HWND hwnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    HDC hdc;
    static int width = LOWORD(lParam), height = HIWORD(lParam), i = 0;
    PAINTSTRUCT paintstruct;
    static SIZE sz;
    HBRUSH brush;
    RECT rect;
    string poruka, title;
    poruka += "h: " + to_string(height) + " w: " + to_string(width);

    switch (message) /* handle the messages */
    {
        case WM_CREATE:
            //printf("WM_CREATE");
            hdc = GetDC(hwnd);
            GetTextExtentPoint32(hdc, poruka.c_str(),
                                strlen(poruka.c_str()), &sz);
            ReleaseDC(hwnd, hdc);
            break;

```

```

case WM_MOVE:
    //printf("WM_MOVE");
    GetWindowRect(hwnd, &rect);
    title += "x:" + to_string(rect.left) + " y: " + to_string(rect.top);
    SetWindowText(hwnd, title.c_str());
break;

case WM_SIZE:
    //printf("WM_SIZE");
    height = HIWORD(lParam);
    width = LOWORD(lParam);
    poruka += "h: " + to_string(height) + " w: " + to_string(width);
break;

case WM_PAINT:
    //printf("WM_PAINT");
    hdc = BeginPaint(hwnd, &paintstruct);
    SetBkColor(hdc, RGB(rand() % 255, rand() % 255, rand() % 255));
    i = i % 2;
    brush = CreateHatchBrush(HS_DIAGCROSS, colors[i]);
    FillRect(hdc, &paintstruct.rcPaint, brush);
    SetBkColor(hdc, RGB(255, 255, 0));

    if(width < 1000 && height < 100)
        TextOut(hdc, paintstruct.rcPaint.right -
sz.cx + 16, paintstruct.rcPaint.bottom - sz.cy, poruka.c_str(),
strlen(poruka.c_str()));
    else if(width < 1000 || height < 100)
        TextOut(hdc, paintstruct.rcPaint.right -
sz.cx + 8, paintstruct.rcPaint.bottom - sz.cy, poruka.c_str(),
strlen(poruka.c_str()));
    else
        TextOut(hdc, paintstruct.rcPaint.right -
sz.cx, paintstruct.rcPaint.bottom - sz.cy, poruka.c_str(),
strlen(poruka.c_str()));
    i++;
    EndPaint(hwnd, &paintstruct);
break;

case WM_DESTROY:
    //printf("WM_DESTROY");
    PostQuitMessage(0);    /* send a WM_QUIT to the message queue */
break;

default:    /* for messages that we don't deal with */
    return DefWindowProc(hwnd, message, wParam, lParam);
}
return 0;
}

```

Kada završite sa radom, kliknite na odgovarajuće dugme na interfejsu kako biste vratili Vaš rad na ocjenu. Uz popunjen dokument dostaviti zip-ovane projekte iz Codeblocks-a pri čemu je zip file imenovan na sljedeći način:

ime_prezime_zadaca1.zip