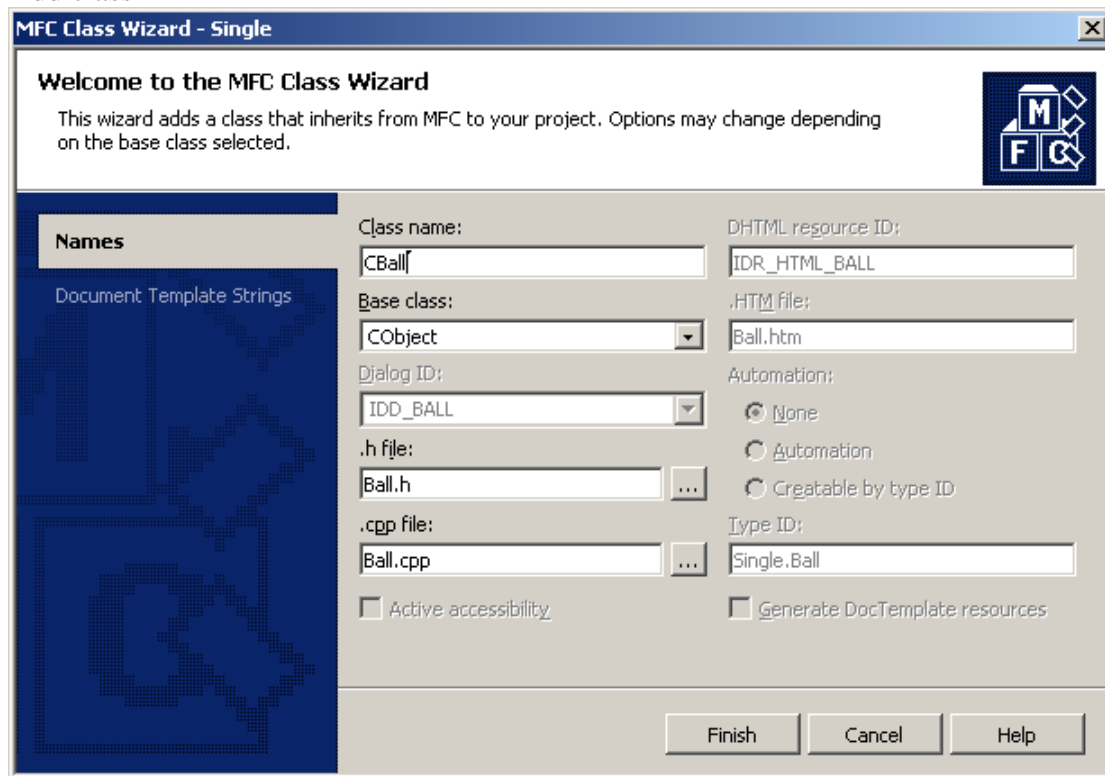


Project "SINGLE": SDI

Add class



add variables: int Radius, float CenterX, CenterY, DeltaX, DeltaY, Velocity

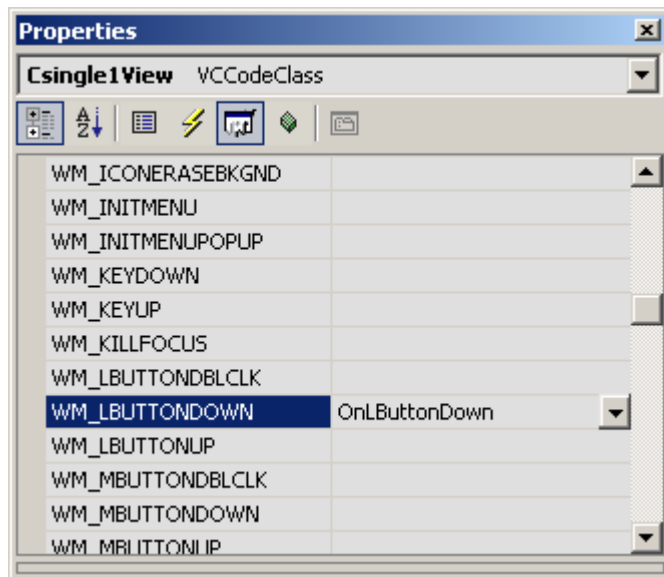
Into `class CSingleDoc` add variable:

```
public:
    COBArray arrayBall;
```

and modify destructor:

```
CSingleDoc::~CSingleDoc()
{
    for (int i = 0; i < arrayBall.GetSize(); i++)
    {
        CBall* pBall = (CBall*) arrayBall[i];
        delete pBall;
    }
}
```

Add "message handlers" "ON\_LBUTTONDOWN" :



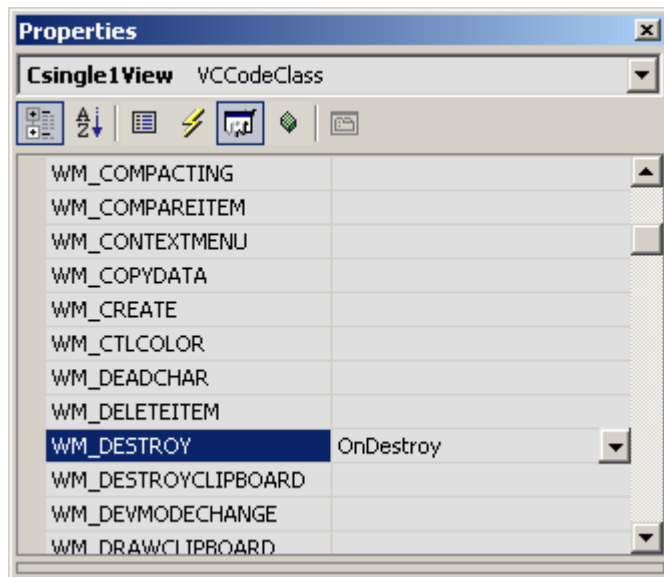
```
void Csingle1View::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CBall *pBall = new CBall();
    pBall->Radius = pBall->CenterX = pBall->CenterY = 10 +
rand()%90; // max radius 100;
    pBall->DeltaX = (float) rand() / RAND_MAX;
    pBall->DeltaY = (float) rand() / RAND_MAX;
    float DeltaLength = sqrt(BallDlg.m_fDX * BallDlg.m_fDX +
        BallDlg.m_fDY * BallDlg.m_fDY);
    pBall->DeltaX = pBall->DeltaX / DeltaLength;
    pBall->DeltaY = pBall->DeltaY / DeltaLength;
    pBall->Velocity = (float) (1+rand()%200); // max velocity
200;

    Csingle1Doc *pDoc = GetDocument();
    pDoc->arrayBall.Add(pBall);

    SetTimer(pDoc->arrayBall.GetSize(), (int) 1000 / pBall-
>Velocity, NULL);

    CView::OnLButtonDown(nFlags, point);
}
```

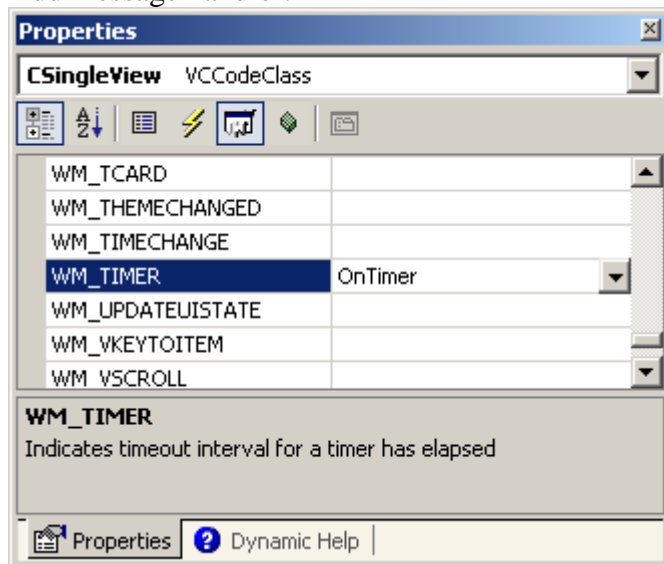
and don't forget to kill timers:



```
CSingleView::OnDestroy()
{
    CSingleDoc *pDoc = GetDocument();
    for (int i = 0; i < pDoc -> arrayBall.GetSize(); i++)
        KillTimer(i+1);
}
```

In "SingleView.cpp" add `#include "Ball.h"`.

Add message handler:



```
void CSingleView::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    CSingleDoc *pDoc = GetDocument();
    CBall *pBall = (CBall*) pDoc -> arrayBall[nIDEvent-1];

    pBall->CenterX += pBall->DeltaX;
    pBall->CenterY += pBall->DeltaY;

    CRect WinRect;
```

```

GetClientRect( &WinRect );
if ( pBall->CenterX < WinRect.left + pBall->Radius ||
    pBall->CenterX > WinRect.right - pBall->Radius)
    pBall->DeltaX *= -1;
if ( pBall->CenterY < WinRect.top + pBall->Radius ||
    pBall->CenterY > WinRect.bottom - pBall->Radius)
    pBall->DeltaY *= -1;

Invalidate();

CView::OnTimer(nIDEvent);
}

```

### Modify OnDraw method:

```

void CSingleView::OnDraw(CDC* pDC)
{
    CSingleDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    // TODO: add draw code for native data here
    for (int i = 0; i < pDoc->arrayBall.GetSize(); i++)
    {
        CBall *pBall = (CBall*) pDoc->arrayBall[i];
        pDC->Ellipse(pBall->CenterX - pBall->Radius,
                    pBall->CenterY - pBall->Radius,
                    pBall->CenterX + pBall->Radius,
                    pBall->CenterY + pBall->Radius);
    }
}

```

### Add “message handlers” "ON\_RBUTTONDOWN":

```

void Csingle1View::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    Csingle1Doc *pDoc = GetDocument();
    KillTimer(pDoc->arrayBall.GetSize());
    CBall *pBall = (CBall*) pDoc->arrayBall[pDoc->arrayBall.GetSize()-1];
    delete pBall;
    pDoc->arrayBall.RemoveAt(pDoc->arrayBall.GetSize()-1);
    Invalidate();
    CView::OnRButtonDown(nFlags, point);
}

```