데굴데굴 플로토타입

\*게임방식

볼을 가지고 있는 크레인이 크레인은 마우스위치를 따라서 좌우로 움직인다.

마우스를 클릭하면 크레인이 볼을 떨어뜨린다.

볼은 9가지 색깔 중에 랜덤으로 색이 정해진다.

볼은 제일 아래쪽에 도착하거나, 쌓인 볼에 부딪히면 떨어지는 것이 멈춘다.

바구니에 쌓인 볼은 아래의 오른쪽과 아래의 왼쪽 중 비어있는 곳으로 이동한다.

바구니에 볼이 쌓인다.

바구니에 같은 색 볼이 3개 이상 연결되면 볼이 사라진다.

\*게임소스 UML

-크래인



-공통함수

Init() : 변수 초기화 함수

Load(): 그림파일을 불러오는 함수

Update(): 경과시간만큼 좌표나 데이터 처리하는 함수

Draw(): 이미지를 출력하는 함수.(현재 크레인이 들고 있는 볼만 그려진다.)

Destroy(): 변수 해제 함수

-고유함수

SetBallX(): 크레인을 좌우로 움직인다.(크레인이 들고 있는 볼이 좌우로 움직인다.)

SetBallColorRandom(): 크레인이 들고 있는 볼을 랜덤으로 색을 정한다.

CloneBall(): 크레인의 볼을 바구니에 담고, 새로운 볼을 만든다.

DeleteBall(): 게임 종료 시 크레인의 볼을 해제 한다.

-컬러



GetIndexColor(): 0에서 9까지 숫자를 입력하면 해당하는 컬러가 나온다.

흰색, 검은색, 빨강, 주황, 노랑, 초록, 파랑, 남색, 보라색 9가지 색상이다.

GetRandomColor(): 9가지 색상 중 랜덤으로 한가지 색이 나온다.

-바구니



m\_Size: 바구니의 사이즈(바구니의 사이즈를 볼의 지름으로 나눠서

볼이 최대로 들어갈 가로 세로 크기를 정한다.)

m\_pNewBall: 크래인에서 떨어지고 있는 볼, 바닥에 도착하거나 볼끼리 충돌하면, 사라진다.

m\_ppBall: 바구니에 있는 볼의 이차원 배열의 포인터, 세로크기와 가로크기만큼 볼의 정보가 저장되어 있다. 볼이 충돌하고 나면 m\_ppBall에 정보를 저장한다.

m\_array\_ball\_w, m\_array\_ball\_h: 바구니의 사이즈안에 들어갈 수 있는 볼의 가로와 세로 개수

m\_vecSameBalls: 같은 컬러의 볼이 연속될 경우 삭제하기 위한 임시 벡터 리스트

<공통함수는 생략>

[[1]](#endnote-1)

1. SetBall(): 크래인에서 가져온 볼을 바구니에 설정한다.

   Collision(): 바닥과 볼간의 충돌처리를 한다.

   GetBall(): 배열의 볼을 가져온다.(캡슐화를 위해 함수로 만들었음)

   DropBallAll(): 위에 붙은 볼들을 아래로 떨어뜨린다.

   RemoveColorSameBallAll(): 3개 이상 같은 색이 인접해 있으면, 삭제한다.

   m\_image\_file\_name: 그림파일의 이미지이다.

   m\_x, m\_y: 볼의 x,y좌표이다.

   m\_radius: 볼의 반지름이다.

   m\_color: 볼의 색깔이다.

   m\_bSettle: 충돌되어서 가장가까운 볼의 자리로 이동되었다.

   m\_bExist: 볼이 존재하는 가? 볼이 존재하지 않으면, Update(), Draw()와 다른 처리를 수행하지 않는다. 처리도 하지 않고, 보이지 않게 된다.

   CollisionBall(): 볼과 바구니의 볼간에 충돌이 일어났는가?

   CollisioinEndLine():볼과 가장 아래쪽 선과 충돌이 일어났는가?

   SettleBall(): 볼이 정해진 위치에 자리잡게 한다. (사용되지 않음), SandBox::Update()에서 처리

   SettleLine(): 볼이 가장 아래쪽에 자리잡게 한다.(사용되지 않음) , SandBox::Update()에서 처리

   <공통함수, 세팅함수 생략>

   -메인함수

   case WM\_MOUSEMOVE:

   {

   mouse\_x = LOWORD(lParam);

   mouse\_y = HIWORD(lParam);

   g\_Crain.SetBallX((float)mouse\_x);

   }

   마우스 이동에 따라 크래인의 볼이 움직인다.

   case WM\_LBUTTONUP:

   {

   g\_bLButtonDown = false;

   mouse\_x=LOWORD(lParam);

   mouse\_y=HIWORD(lParam);

   ImBall\* pBall = g\_Crain.CloneBall();

   TheSandBox->SetBall(pBall);

   g\_Crain.SetBallColorRandom();

   }

   마우스를 눌렸다 떼면, 크래인의 볼이 바구니에 전달되고, 크래인의 볼은 새로운 컬러로 설정된다.

   -GDI+

   전역변수

   HDC g\_hdc;

   HDC g\_memdc;

   Bitmap\* pMemBitmap = NULL;

   GdiplusStartupInput gdiplusStartupInput;

   ULONG\_PTR gdiplusToken;

   초기화

   GdiplusStartup(&gdiplusToken, &gdiplusStartupInput, NULL);

   g\_hdc = GetDC(g\_hWnd);

   g\_memdc = CreateCompatibleDC(g\_hdc);

   RECT rt;

   GetClientRect(g\_hWnd,&rt);

   pMemBitmap = new Bitmap(rt.right - rt.left, rt.bottom - rt.top);

   해제

   delete pMemBitmap;

   ReleaseDC(g\_hWnd,g\_memdc);

   ReleaseDC(g\_hWnd, g\_hdc);

   GdiplusShutdown(gdiplusToken);

   그리기(더블버퍼링)

   Graphics g\_hdcGraphics(g\_hdc);

   Graphics g\_memGraphics(pMemBitmap);

   g\_memGraphics.Clear(Color::AliceBlue);

   TheSandBox->Draw(g\_memGraphics);

   g\_Crain.Draw(g\_memGraphics);

   //g\_Ball.Draw(g\_memGraphics);

   g\_hdcGraphics.DrawImage(pMemBitmap,0,0);

   게임루프

   Init();

   while(true)

   {

   float fCurTime = GetTickCount() \* 0.001f;

   float fElapsedTime = fCurTime - fLastTime;

   fLastTime = fCurTime;

   while(PeekMessage(&Message,hWnd,0,0,PM\_REMOVE))

   {

   TranslateMessage(&Message);

   DispatchMessage(&Message);

   }

   if(Message.message == WM\_QUIT)

   break;

   Input();

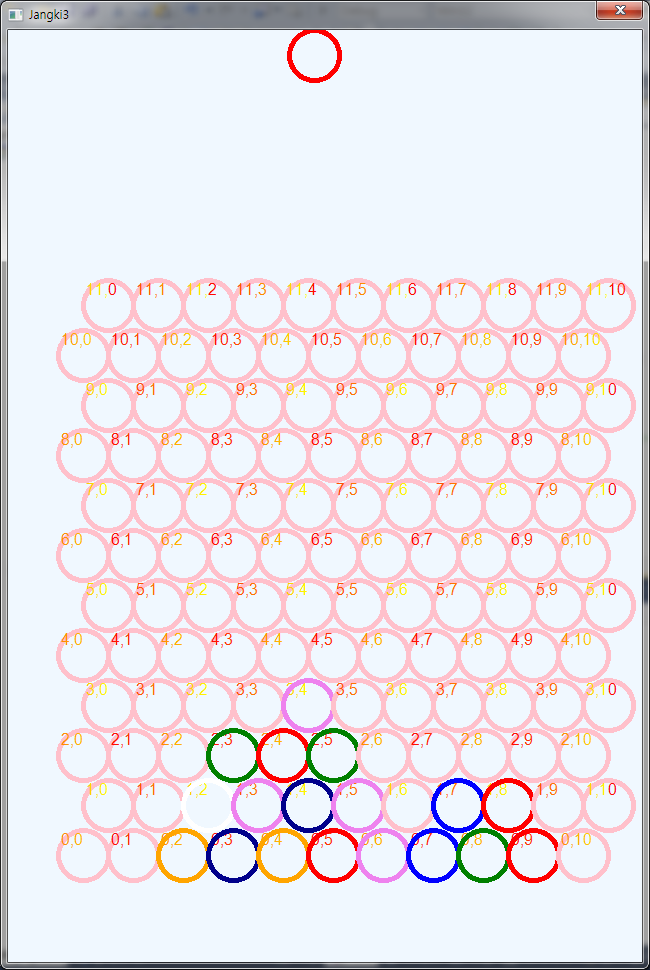
   Update(fElapsedTime);

   Display();

   }

   Destroy();

   실행화면

    [↑](#endnote-ref-1)