

게임프로그래밍

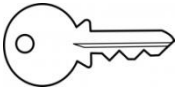
---

# 기하 2

박종승

---

Dept. of CSE, Incheon Nat. Univ.  
jong@inu.ac.kr  
<http://ecl.inu.ac.kr>



# 경로 기하

- 경로 기하(path geometry)
  - 원호, 곡선, 직선 등의 조각들로 구성된 복잡한 기하모양을 표현
  - ID2D1PathGeometry로 표현
    - ID2D1Geometry를 상속하였음
  - 생성: ID2D1Factory::CreatePathGeometry 함수를 호출
- 경로 기하의 내용을 작성하는 방법
  - ID2D1PathGeometry::Open 함수를 호출하여 ID2D1GeometrySink 를 얻기
  - ID2D1GeometrySink의 AddXXX 함수들을 사용하여 조각들을 추가
- ID2D1GeometrySink 사용하기
  - 조각들로 경로 기하를 표현하는 기능을 제공함
  - 조각: 라인, 원호, 베지어 곡선, 이차 베지어 곡선

# 기하 싱크를 사용하여 경로 기하 작성하기

---

- 기하 싱크(geometry sink)란?
  - 한 기하 싱크는 1개 이상의 피겨(figure)로 구성됨
  - 한 피겨는 1개 이상의 조각(segment)로 구성됨
    - 조각: 라인, 곡선, 원호
- 피겨 만들기 절차: 기하싱크의 함수들을 호출!
  - 1. 피겨를 시작: `BeginFigure` 함수 호출
  - 2. 조각을 추가: `AddXXX` 함수 호출
  - 3. 피겨를 완성: `EndFigure` 함수 호출
  - 다른 피겨를 또 만들고 싶으면 위의 단계를 반복
  - 모든 피겨를 완성한 후에: `Close` 함수 호출
- 참고
  - `ID2D1GeometrySink`는 `ID2D1SimplifiedGeometrySink`를 상속하였음
    - `ID2D1GeometrySink`는 일부 함수들(`AddXXX` 5개)을 제공
    - 그 외의 함수들(7개)은 `ID2D1SimplifiedGeometrySink`가 제공

# 피겨 만들기

---

- 피겨 만들기
  - 조각들을 순차적으로 연결하여 그려서 피겨를 완성함
  - 각 조각은 항상 현재점(current point)에서 시작하여 그림
  - 피겨의 출발점(즉 최초의 현재점)은 `BeginFigure` 함수의 인자로 지정함
  - 각 `AddXXX` 함수는 현재점에서 시작하는 한 조각을 피겨에 추가함
  - 각 `AddXXX` 함수가 실행된 후에는 현재점이 조각의 마지막 위치로 바뀜
- `BeginFigure`
  - 인자1: `D2D1_POINT_2F startPoint` : 새 피겨의 시작점
  - 인자2: `D2D1_FIGURE_BEGIN figureBegin` : 피겨가 채워진(filled) 것인지 속이 빈(hollow) 것인지의 여부
    - `D2D1_FIGURE_BEGIN_{FILLED,HOLLOW}`
- `EndFigure`
  - 인자1: `figureEnd` : 현재 피겨가 닫힌 모양인지를 명시함
    - 닫힌 모양이라면 현재점에서 `BeginFigure`으로 명시한 시작점으로 선분을 연결한 후에 피겨를 종료함
    - `D2D1_FIGURE_END_{OPEN,CLOSED}`

# AddXXX 함수들

---

- 선분
  - **AddLine** : 현재점에서 끝점으로 가는 선분을 추가함
    - 인자1: D2D1\_POINT\_2F point //선분의 끝점
  - **AddLines** : 선분들을 추가함
    - 인자1: D2D1\_POINT\_2F \*points //현재점에서 배열의 첫 번째 점으로, // 배열의 첫 번째 점에서 두 번째 점으로, ... 등으로 선분들을 추가함
    - 인자2: UINT pointsCount //배열에 있는 점들의 개수

# AddXXX 함수들'

---

- 원호

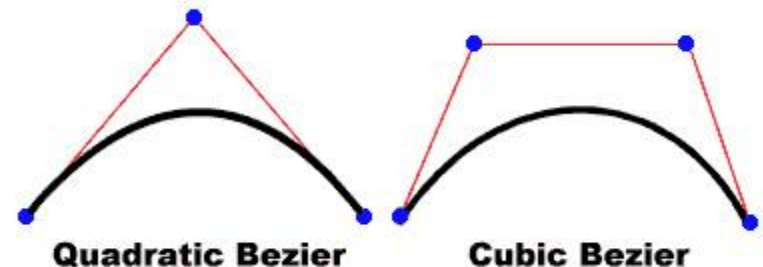
- AddArc : 한 원호를 추가함

- 인자1: D2D1\_ARC\_SEGMENT arc //추가할 원호 조각 (편리함수: ArcSegment)
      - D2D1\_POINT\_2F point; //원호의 끝점
      - D2D1\_SIZE\_F size; //원호의 x/y- 반지름
      - FLOAT rotationAngle; //타원이 시계방향으로 얼마나 회전되었는지의 각도. 0이면 좌우대칭을 이루는 타원의 위쪽 또는 아래쪽 절반에 해당함.
      - D2D1\_SWEEP\_DIRECTION sweepDirection; //호가 진행하는 방향을 명시함
        - » D2D1\_SWEEP\_DIRECTION\_{COUNTER\_CLOCKWISE,CLOCKWISE}
      - D2D1\_ARC\_SIZE arcSize; //원호가 180도 이하/이상임을 명시함
        - » D2D1\_ARC\_SIZE\_{SMALL,LARGE}

# AddXXX 함수들"

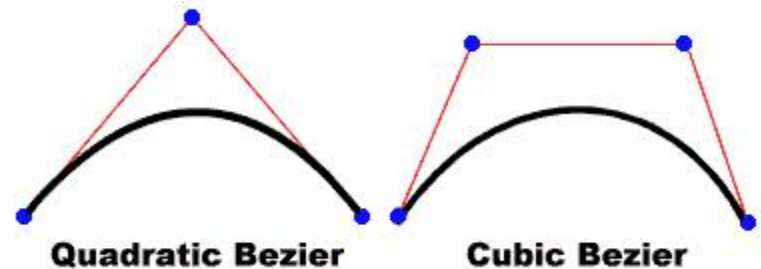
- 2차곡선
  - **AddQuadraticBezier** : 현재점에서 끝점까지 연결하는 2차Bezier곡선을 추가함
    - 인자1: D2D1\_QUADRATIC\_BEZIER\_SEGMENT bezier //추가할 2차Bezier곡선 조각 (편리함수: QuadraticBezierSegment )
      - D2D1\_POINT\_2F point1; //2차Bezier곡선의 제어점
      - D2D1\_POINT\_2F point2; //2차Bezier곡선의 끝점
  - **AddQuadraticBeziers** : 여러 2차Bezier곡선들을 추가함
    - 인자1: D2D1\_QUADRATIC\_BEZIER\_SEGMENT \*beziers  
//D2D1\_QUADRATIC\_BEZIER\_SEGMENT의 배열
    - 인자2: UINT bezierCount //2차Bezier곡선들의 개수

※ 제어점: 시작점과 끝점을 연결하는 선분을 당기는 자석과 유사함. 선분을 제어점 방향으로 당겨서 곡선을 만듦. 곡선이 제어점을 지나가지는 않음!

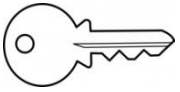


# AddXXX 함수들'''

- 3차곡선
  - **AddBezier** : 현재점에서 끝점까지 연결하는 3차Bezier곡선을 추가함
    - 인자1: D2D1\_BEZIER\_SEGMENT bezier //추가할 3차Bezier곡선 조각 (편리함수: BezierSegment )
      - D2D1\_POINT\_2F point1; //첫 번째 제어점
      - D2D1\_POINT\_2F point2; //두 번째 제어점
      - D2D1\_POINT\_2F point3; //끝점
  - **AddBeziers** : 여러 3차Bezier곡선들을 추가함
    - 인자1: D2D1\_BEZIER\_SEGMENT \*beziers  
//D2D1\_BEZIER\_SEGMENT의 배열
    - 인자2: UINT beziersCount //3차Bezier곡선들의 개수







# 복잡한 그림 생성하기 예제

01.FourPathGeometries

- 복잡한 그림
  - 4개의 경로 기하 객체를 사용
    - 왼쪽 산, 오른쪽 산, 해(불꽃포함), 강



# 1: 왼쪽 산의 경로 기하 만들기

- 1: 왼쪽 산의 경로 기하 만들기

- 경로 기하 생성

- CreatePathGeometry를 호출하여 ID2D1PathGeometry를 생성

- Open 함수를 호출하여 ID2D1PathGeometry로부터 기하 싱크 얻기

- BeginFigure 를 호출

- 피겨를 fill하도록 D2D1\_FIGURE\_BEGIN\_FILLED 명시

- AddLines 를 호출

- D2D1\_POINT\_2F 타입의 점들의 배열을 명시

```
m_pD2DFactory->CreatePathGeometry(&m_pLeftMountainGeometry);
```

```
ID2D1GeometrySink *pSink = NULL;
```

```
m_pLeftMountainGeometry->Open(&pSink);
```

```
pSink->SetFillMode(D2D1_FILL_MODE_WINDING); //이후에 별도로 설명함
```

```
pSink->BeginFigure( D2D1::Point2F(346,255), D2D1_FIGURE_BEGIN_FILLED );
```

```
D2D1_POINT_2F points[5] = { D2D1::Point2F(267, 177), D2D1::Point2F(236, 192),  
                             D2D1::Point2F(212, 160), D2D1::Point2F(156, 255), D2D1::Point2F(346, 255), };
```

```
pSink->AddLines(points, ARRAYSIZE(points));
```

```
pSink->EndFigure(D2D1_FIGURE_END_CLOSED);
```

```
pSink->Close();
```

```
SafeRelease(&pSink);
```



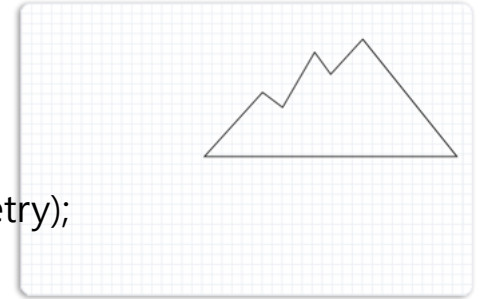
## 2: 오른쪽 산의 경로 기하 만들기

- 2: 오른쪽 산의 경로 기하 만들기
  - 이전과 유사

```
m_pD2DFactory->CreatePathGeometry(&m_pRightMountainGeometry);
ID2D1GeometrySink *pSink = NULL;
m_pRightMountainGeometry->Open(&pSink);

pSink->SetFillMode(D2D1_FILL_MODE_WINDING);
pSink->BeginFigure( D2D1::Point2F(575,263), D2D1_FIGURE_BEGIN_FILLED );
D2D1_POINT_2F points[] = { D2D1::Point2F(481, 146), D2D1::Point2F(449, 181),
                           D2D1::Point2F(433, 159), D2D1::Point2F(401, 214), D2D1::Point2F(381, 199),
                           D2D1::Point2F(323, 263), D2D1::Point2F(575, 263) };
pSink->AddLines(points, ARRAYSIZE(points));
pSink->EndFigure(D2D1_FIGURE_END_CLOSED);

pSink->Close();
SafeRelease(&pSink);
```



# 3: 해의 경로 기하 만들기

- 3: 해의 경로 기하 만들기 – Arc, Bezier

```
m_pD2DFactory->CreatePathGeometry(&m_pSunGeometry);
ID2D1GeometrySink *pSink = NULL;
m_pSunGeometry->Open(&pSink);

pSink->SetFillMode(D2D1_FILL_MODE_WINDING);
pSink->BeginFigure( D2D1::Point2F(270, 255), D2D1_FIGURE_BEGIN_FILLED );
pSink->AddArc(D2D1::ArcSegment(D2D1::Point2F(440, 255), D2D1::SizeF(85, 85),0.0f,
    D2D1_SWEEP_DIRECTION_CLOCKWISE, D2D1_ARC_SIZE_SMALL ));
pSink->EndFigure(D2D1_FIGURE_END_CLOSED);

pSink->BeginFigure( D2D1::Point2F(299, 182), D2D1_FIGURE_BEGIN_HOLLOW );
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(299, 182), D2D1::Point2F(294, 176), D2D1::Point2F(285, 178) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(276, 179), D2D1::Point2F(272, 173), D2D1::Point2F(272, 173) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

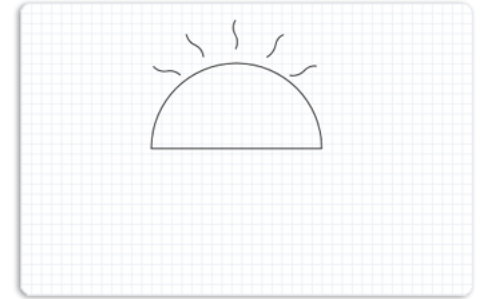
pSink->BeginFigure(D2D1::Point2F(354, 156),D2D1_FIGURE_BEGIN_HOLLOW);
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(354, 156), D2D1::Point2F(358, 149), D2D1::Point2F(354, 142) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(349, 134), D2D1::Point2F(354, 127), D2D1::Point2F(354, 127) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

pSink->BeginFigure(D2D1::Point2F(322,164),D2D1_FIGURE_BEGIN_HOLLOW);
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(322, 164), D2D1::Point2F(322, 156), D2D1::Point2F(314, 152) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(306, 149), D2D1::Point2F(305, 141), D2D1::Point2F(305, 141) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

pSink->BeginFigure(D2D1::Point2F(385, 164),D2D1_FIGURE_BEGIN_HOLLOW);
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(385,164), D2D1::Point2F(392,161), D2D1::Point2F(394,152) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(395,144), D2D1::Point2F(402,141), D2D1::Point2F(402,142) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

pSink->BeginFigure(D2D1::Point2F(408,182),D2D1_FIGURE_BEGIN_HOLLOW);
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(408,182), D2D1::Point2F(416,184), D2D1::Point2F(422,178) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(428,171), D2D1::Point2F(435,173), D2D1::Point2F(435,173) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

pSink->Close();
SafeRelease(&pSink);
```



## 4: 강의 경로 기하 만들기

- 4: 강의 경로 기하 만들기 – Bezier

```
m_pD2DFactory->CreatePathGeometry(&m_pRiverGeometry);
ID2D1GeometrySink *pSink = NULL;
m_pRiverGeometry->Open(&pSink);
pSink->SetFillMode(D2D1_FILL_MODE_WINDING);
pSink->BeginFigure(D2D1::Point2F(183, 392), D2D1_FIGURE_BEGIN_FILLED);
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(238, 284),
                                         D2D1::Point2F(472, 345), D2D1::Point2F(356, 303) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(237, 261),
                                         D2D1::Point2F(333, 256), D2D1::Point2F(333, 256) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(335, 257),
                                         D2D1::Point2F(241, 261), D2D1::Point2F(411, 306) ));
pSink->AddBezier( D2D1::BezierSegment( D2D1::Point2F(574, 350),
                                         D2D1::Point2F(288, 324), D2D1::Point2F(296, 392) ));
pSink->EndFigure(D2D1_FIGURE_END_OPEN);

pSink->Close();
SafeRelease(&pSink);
```



## 5: 만든 경로 기하들을 화면에 그리기

- 5: 만든 경로 기하들을 화면에 그리기
  - 순서대로, 해, 왼쪽 산, 강, 오른쪽 산을 그림

```
m_pRenderTarget->BeginDraw();
m_pRenderTarget->SetTransform(D2D1::Matrix3x2F::Identity());
m_pRenderTarget->Clear(D2D1::ColorF(D2D1::ColorF::White));

D2D1_SIZE_F rtSize = m_pRenderTarget->GetSize();
m_pRenderTarget->FillRectangle( D2D1::RectF(0, 0, rtSize.width, rtSize.height), m_pGridPatternBitmapBrush );

m_pRenderTarget->FillGeometry(m_pSunGeometry, m_pRadialGradientBrush);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::Black, 1.f));
m_pRenderTarget->DrawGeometry(m_pSunGeometry, m_pSceneBrush, 1.f);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::OliveDrab, 1.f));
m_pRenderTarget->FillGeometry(m_pLeftMountainGeometry, m_pSceneBrush);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::Black, 1.f));
m_pRenderTarget->DrawGeometry(m_pLeftMountainGeometry, m_pSceneBrush, 1.f);

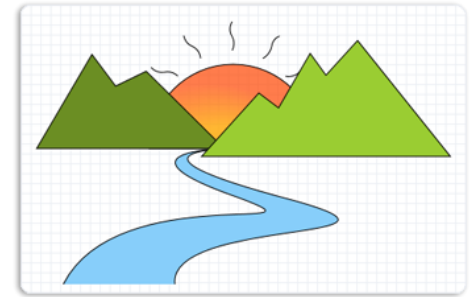
m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::LightSkyBlue, 1.f));
m_pRenderTarget->FillGeometry(m_pRiverGeometry, m_pSceneBrush);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::Black, 1.f));
m_pRenderTarget->DrawGeometry(m_pRiverGeometry, m_pSceneBrush, 1.f);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::YellowGreen, 1.f));
m_pRenderTarget->FillGeometry(m_pRightMountainGeometry, m_pSceneBrush);

m_pSceneBrush->SetColor(D2D1::ColorF(D2D1::ColorF::Black, 1.f));
m_pRenderTarget->DrawGeometry(m_pRightMountainGeometry, m_pSceneBrush, 1.f);

m_pRenderTarget->EndDraw();
```



## 예제: 기하 그리기

- 예제: 단순 기하와 경로 기하를 그리기

02.GeometriesExample

