



# SFML-개인 프로젝트 SimCity2000 모작

기맛플밍 6기 김희정

# 게임 선정 이유

---

```
bool Utils::PointInTransformBounds(const sf::Transformable& transformable, const sf::FloatRect& localBounds, const sf::Vector2f point)
{
    sf::Transform inverse = transformable.getInverseTransform();
    return localBounds.contains(inverse.transformPoint(point));
}
```

- 아이소메트릭 뷰에서 마우스 좌표를 타일 인덱스로 매핑해보고 싶다!



# 목차

---

1. 구현범위
2. 마우스 타일 충돌 감지
3. MCV 디자인 패턴
4. 브레셴햄 알고리즘
5. 9 슬라이싱 스프라이트
6. 잘한점 못한점

# 구현범위

---

- UI 및 버튼
- 도로, 전기선 등 연결타일 건설
- 구역 지정

# 구현범위

- UI 버튼



```

108
109 void SimCityButtonBar::Update(float dt)
110 {
111     sf::Vector2f currMousePos = INPUT_MGR->GetMouseViewPos(m_ViewIndex);
112     m_HasFocus = false;
113     if (m_Bar->GetGlobalBounds().contains(currMousePos))
114     {
115         m_HasFocus = true;
116     }
117
118     for (int i = 0; i < m_ButtonCount; i++)
119     {
120         if (m_ButtonWork[i])
121         {
122             if (m_Buttons[i]->GetGlobalBounds().contains(currMousePos) && INPUT_MGR->GetMouseDown(sf::Mouse::Left))
123             {
124                 SOUND_MGR->PlaySfx("sound/SFX/Click.wav");
125                 m_Buttons[m_CurrButtonIndex]->SetColor(ColorPalette::White);
126                 m_CurrButtonIndex = i;
127                 m_Buttons[m_CurrButtonIndex]->SetColor(ColorPalette::Gray);
128                 m_ButtonFunc((Action)m_CurrButtonIndex);
129
130                 if (m_SubButtons[i])
131                 {
132                     m_SubButtons[i]->ShowSubBar(true);
133                     m_SubButtons[i]->setLocalPosition(m_Buttons[i]->getLocalPosition() + sf::Vector2f(0, 32));
134                 }
135             }
136         }
137     }
138 }

```

```

49
50 void SimCitySubButton::Update(float dt)
51 {
52     if (m_Delay) { ... }
57
58     sf::Vector2f currMousePos = INPUT_MGR->GetMouseViewPos(m_ViewIndex);
59     int currIndex = -1;
60     for (int i = 0; i < m_Buttons.size(); i++)
61     {
62         m_Buttons[i]->SetColor(ColorPalette::DarkGray);
63         if (m_Buttons[i]->GetGlobalBounds().contains(currMousePos))
64         {
65             m_Buttons[i]->SetColor(ColorPalette::Black);
66             currIndex = i;
67         }
68     }
69
70     if (INPUT_MGR->GetMouseUp(sf::Mouse::Left) || INPUT_MGR->GetMouseDown(sf::Mouse::Left))
71     {
72         if (currIndex != -1)
73         {
74             m_ButtonFunc[currIndex]();
75         }
76         ShowSubBar(false);
77     }
78 }

```

# 구현범위

- 도로, 전기선 등 연결타일 건설

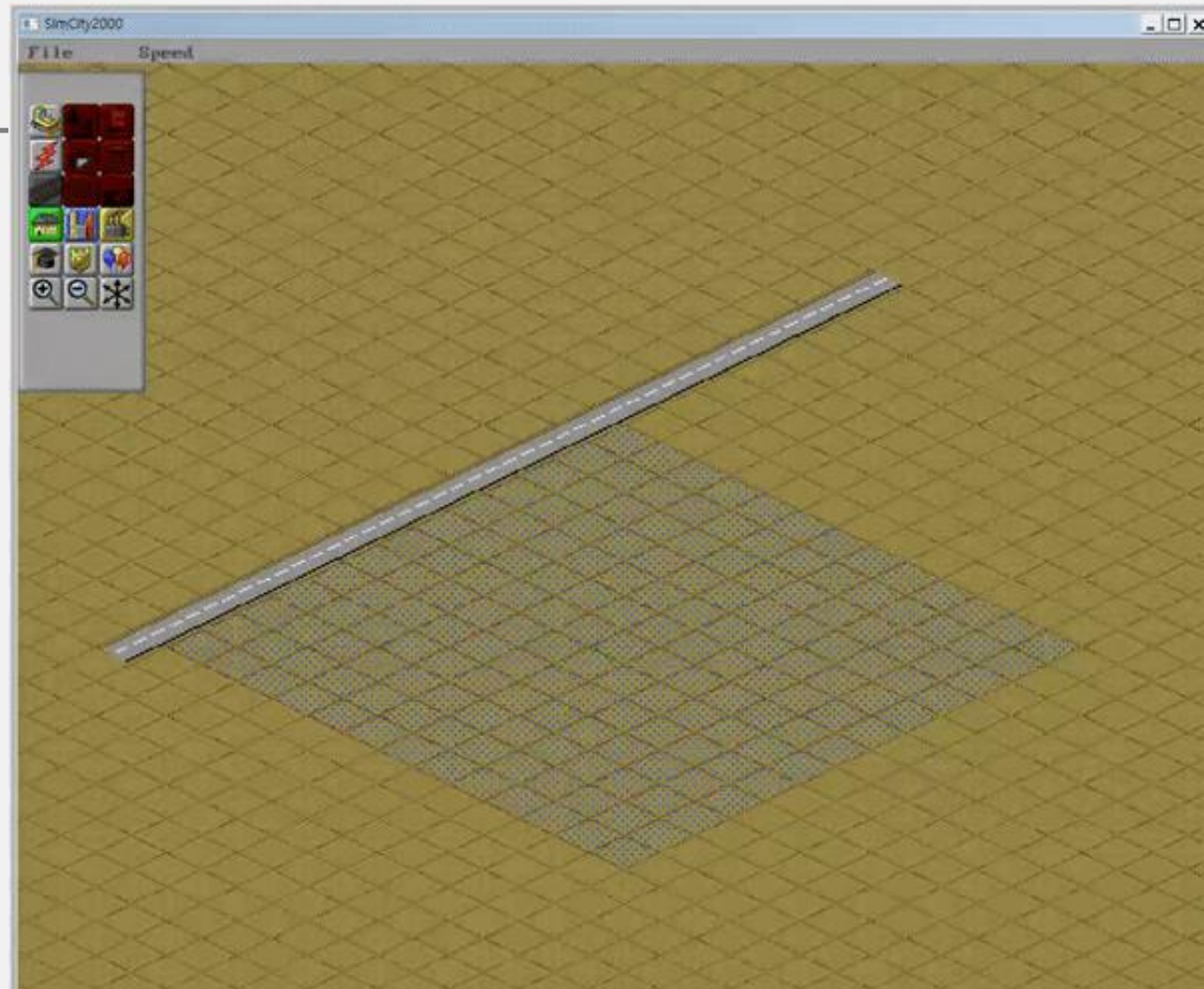




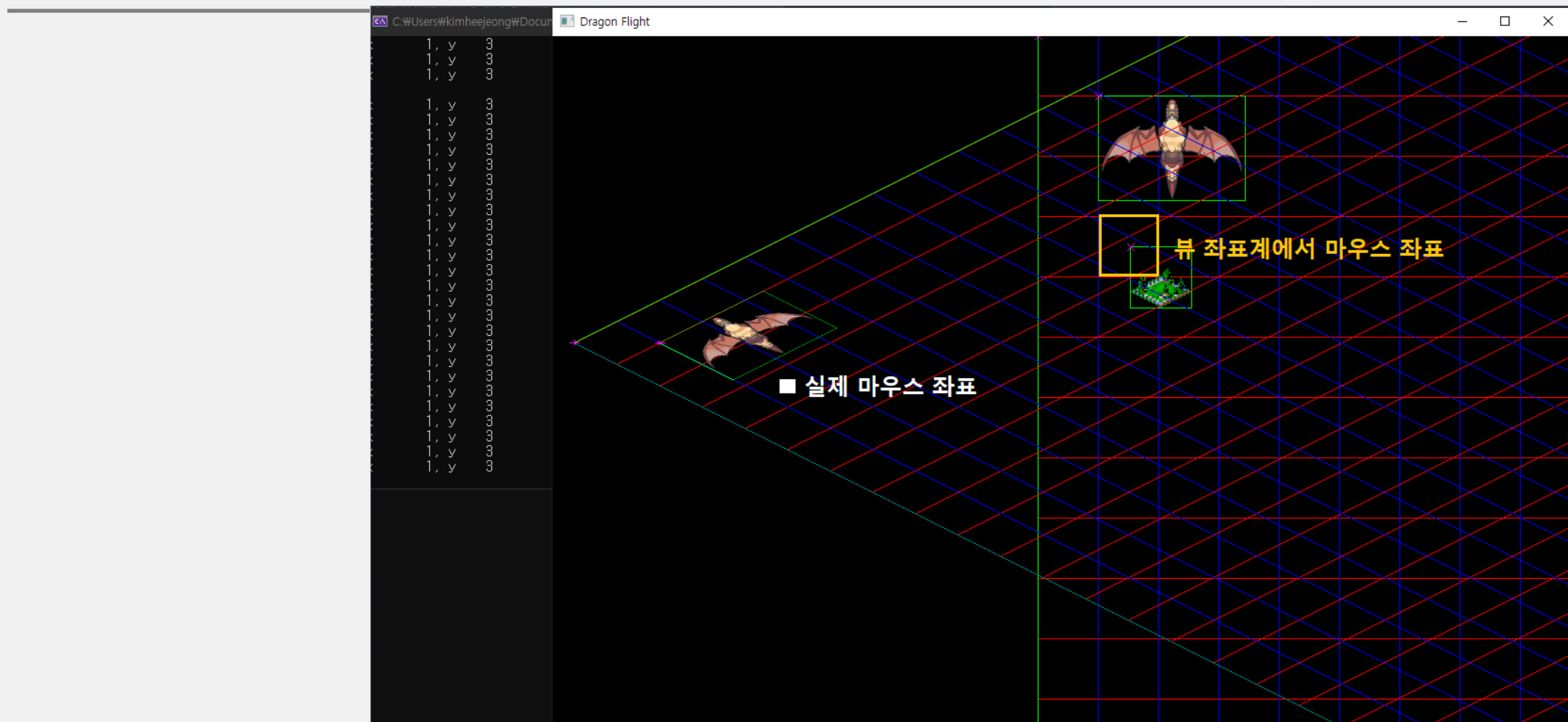
# 구현범위

- 구역 건설

전기선이 연결되어 있고,  
도로타일과 3칸 이내면 구역에  
맞는 건물이 생성됨



# 마우스 타일 충돌 감지



---

```
▼ sf::Vector2f TileView::GetTileCoordinatedPos(const sf::Vector2f& pos) const
{
    return m_TileTransform.getInverse().transformPoint(pos);
}
```

```
▼ CellIndex TileView::GetTileCoordinatedIndex(const sf::Vector2f& pos, bool isTilepos) const
{
    sf::Vector2f tilepos = isTilepos ? pos : GetTileCoordinatedPos(pos);
    return sf::Vector2i(tilepos.x / mcv_Model->m_CellSize.x, tilepos.y / mcv_Model->m_CellSize.y);
}
```

```
▼ sf::Vector2f TileView::GetTileCoordinatedCenterPosByTileIndex(const CellIndex& tileIndex)
{
    return 1.5f * sf::Vector2f(mcv_Model->m_CellSize.x * tileIndex.x, mcv_Model->m_CellSize.y * tileIndex.y);
}
```