

2006년 2학기 윈도우 게임 프로그래밍

## 제13강 스크롤링과 타일링 (III)

이대현

한국산업기술대학교



한국산업기술대학교

### 오늘의 학습 내용

- 동축 타일링(Isometric Tiling) 맵의 상하좌우 스크롤링.
- 마우스 좌표를 타일 상의 좌표로 변환.

실습



Lecture13-01:

동축타일링 맵의 상하좌우스크롤링

## Lecture13-01 프로젝트의 구성

### ■ C++ 소스 파일들

- background.cpp – 실습 시간에 작성.
- gameengine.cpp
- main.cpp
- introstate.cpp
- playstate.cpp
- sprite.cpp

### ■ C++ 헤더 파일들

- background.h
- gameengine.h
- gamestate.h
- introstate.h
- playstate.h
- sprite.h

# background.cpp (1)

```
void Background::scrollRight(void)
{
    framex = (framex + MAP_TILE_WIDTH - 1) % MAP_TILE_WIDTH;
    offsetx = 0;
}

void Background::scrollLeft(void)
{
    framex = (framex + 1) % MAP_TILE_WIDTH;
    offsetx = 0;
}

void Background::scrollDown(void)
{
    framey = (framey + MAP_TILE_HEIGHT - 1) % MAP_TILE_HEIGHT;
    offsety = 0;
}

void Background::scrollUp(void)
{
    framey = (framey + 1) % MAP_TILE_HEIGHT;
    offsety = 0;
}
```

# background.cpp (2)

```
#define MIN(x, y) (((x) < (y)) ? (x) : (y))

void Background::draw(SDL_Surface *screen)
{
    SDL_Rect frame, targetFrame;

    // left top
    frame.x = framex;
    frame.y = framey;
    frame.w = MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
    frame.h = MIN(MAP_TILE_HEIGHT - framey, SCREEN_TILE_HEIGHT);

    blitFrame(&frame, screen, NULL);

    // right top
    frame.x = 0;
    frame.y = framey;
    frame.w = SCREEN_TILE_WIDTH - MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
    frame.h = MIN(MAP_TILE_HEIGHT - framey, SCREEN_TILE_HEIGHT);

    targetFrame.x = MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
    targetFrame.y = 0;

    blitFrame(&frame, screen, &targetFrame);
}
```

# background.cpp (3)

```
// left bottom
frame.x = framex;
frame.y = 0;
frame.w = MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
frame.h = SCREEN_TILE_HEIGHT - MIN(MAP_TILE_HEIGHT - framey,
SCREEN_TILE_HEIGHT);

targetFrame.x = 0;
targetFrame.y = MIN(MAP_TILE_HEIGHT - framey, SCREEN_TILE_HEIGHT);

blitFrame(&frame, screen, &targetFrame);

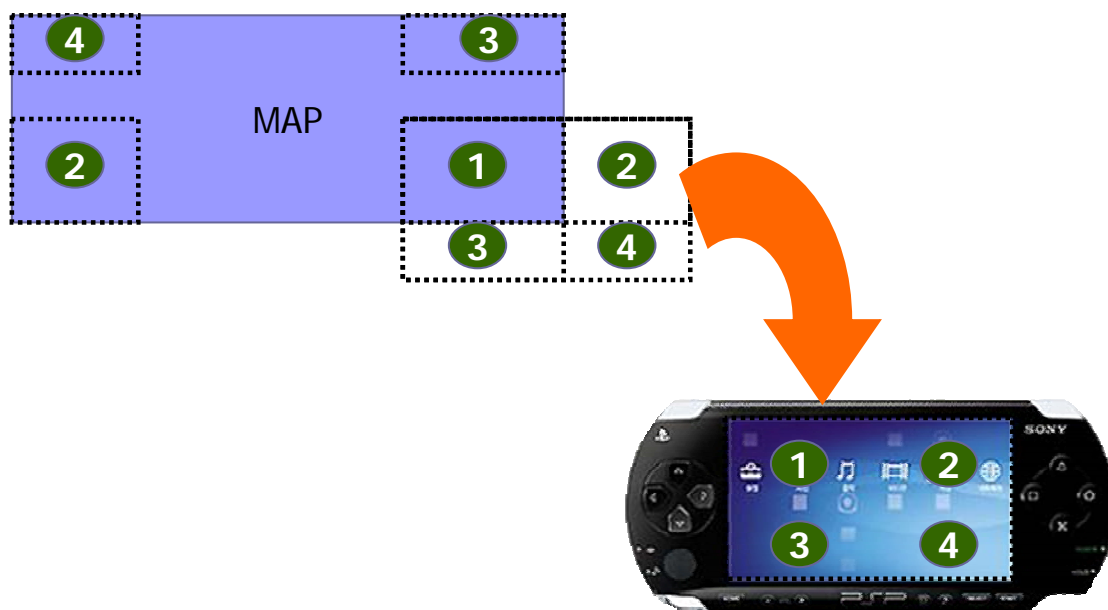
// right bottom
frame.x = 0;
frame.y = 0;
frame.w = SCREEN_TILE_WIDTH - MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
frame.h = SCREEN_TILE_HEIGHT - MIN(MAP_TILE_HEIGHT - framey,
SCREEN_TILE_HEIGHT);

targetFrame.x = MIN(MAP_TILE_WIDTH - framex, SCREEN_TILE_WIDTH);
targetFrame.y = MIN(MAP_TILE_HEIGHT - framey, SCREEN_TILE_HEIGHT);

blitFrame(&frame, screen, &targetFrame);

putTile(screen, 3, SCREEN_TILE_WIDTH / 2, SCREEN_TILE_HEIGHT / 2);
}
```

## 상하좌우 스크롤링 공식



# 실행 화면



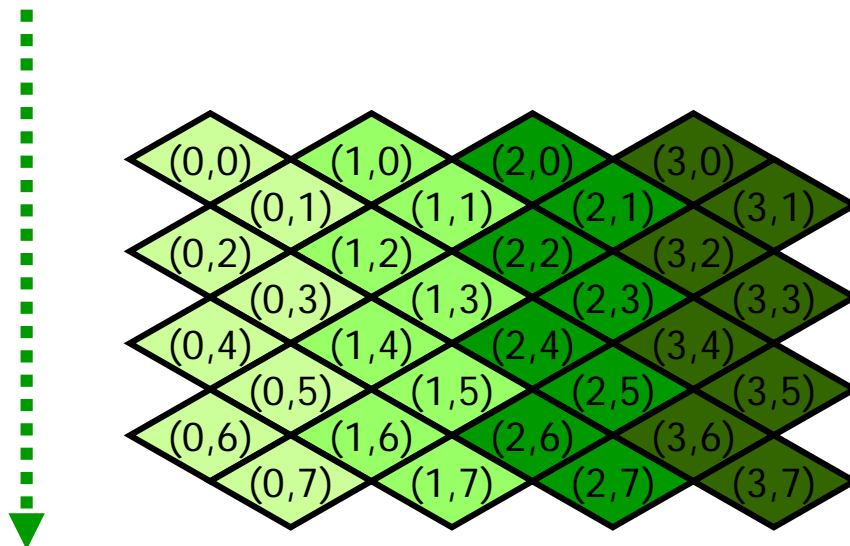
캐릭터를 위로 한칸 이동



캐릭터를 위로 다시 한칸 이동



# 상하 스크롤링



y값을 1씩 증가시키면, 타일의 x 축 위치가 계속해서 변경되므로, 스크롤이 깨진다  
 → 방법은? y값을 2씩 증가시키면 됨.

```

void Background::scrollRight(void)
{
    framex = (framex + MAP_TILE_WIDTH - 1) % MAP_TILE_WIDTH;
    offsetx = 0;
}

void Background::scrollLeft(void)
{
    framex = (framex + 1) % MAP_TILE_WIDTH;
    offsetx = 0;
}

void Background::scrollDown(void)
{
    framey = (framey + MAP_TILE_HEIGHT - 2) % MAP_TILE_HEIGHT;
    offsety = 0;
}

void Background::scrollUp(void)
{
    framey = (framey + 2) % MAP_TILE_HEIGHT;
    offsety = 0;
}

```

## 부드러운 상하 스크롤링

```

void Background::scrollRight(void)
{
    leftx = (leftx + (TILE_WIDTH * MAP_TILE_WIDTH) - 8) % (TILE_WIDTH *
        MAP_TILE_WIDTH);
    framex = leftx / TILE_WIDTH; offsetx = leftx % TILE_WIDTH;
}

void Background::scrollLeft(void)
{
    leftx = (leftx + 8) % (TILE_WIDTH * MAP_TILE_WIDTH);
    framex = leftx / TILE_WIDTH; offsetx = leftx % TILE_WIDTH;
}

void Background::scrollDown(void)
{
    lefty = (lefty + (TILE_HEIGHT * MAP_TILE_HEIGHT / 2) - 4) % (TILE_HEIGHT *
        MAP_TILE_HEIGHT / 2);
    framey = (lefty / TILE_HEIGHT) * 2; offsety = lefty % (TILE_HEIGHT);
}

void Background::scrollUp(void)
{
    lefty = (lefty + 4) % (TILE_HEIGHT * MAP_TILE_HEIGHT / 2);
    framey = (lefty / TILE_HEIGHT) * 2; offsety = lefty % (TILE_HEIGHT);
}

```

```
lefty = (lefty + 4) % (TILE_HEIGHT * MAP_TILE_HEIGHT / 2);
```

```
framey = (lefty/TILE_HEIGHT)*2;
```

```
offsety = lefty % (TILE_HEIGHT);
```

실습



Lecture13-02:

좌표 변환: 마우스 → 타일 좌표

# Lecture13-02 프로젝트의 구성

## ■ C++ 소스 파일들

- background.cpp – 실습 시간에 작성.
- gameengine.cpp
- main.cpp
- introstate.cpp
- playstate.cpp – 실습 시간에 작성.
- sprite.cpp

## ■ C++ 헤더 파일들

- background.h
- gameengine.h
- gamestate.h
- introstate.h
- playstate.h
- sprite.h

## background.cpp (1)

```
void Background::putMouse(SDL_Surface *screen, SDL_Rect *mouseRect)
{
    int nx, ny, tilex, tiley;

    mouseRect->x += offsetx;
    mouseRect->y += offsety;

    mouseRect->x += (TILE_WIDTH / 2);
    mouseRect->y += (TILE_HEIGHT / 2);

    nx = mouseRect->x % TILE_WIDTH;
    ny = mouseRect->y % TILE_HEIGHT;
    tilex = mouseRect->x / TILE_WIDTH;
    tiley = (mouseRect->y / TILE_HEIGHT) * 2;
```



## background.cpp (2)

```
if (nx < TILE_WIDTH / 2 - (TILE_WIDTH * ny / TILE_HEIGHT)) {
    tilex--;
    tiley--;
} else if (nx < (TILE_WIDTH * ny / TILE_HEIGHT) - (TILE_WIDTH / 2)) {
    tilex--;
    tiley++;
} else if (nx > (TILE_WIDTH * ny / TILE_HEIGHT) + (TILE_WIDTH / 2)) {
    tiley--;
} else if (nx > (3 * TILE_WIDTH / 2) - (TILE_WIDTH * ny / TILE_HEIGHT)) {
    tiley++;
}

int mapx = (tilex + framex) % (MAP_TILE_WIDTH);
int mapy = (tiley + framey) % (MAP_TILE_HEIGHT);
printf("(nx, ny, tilex, tiley, mapx, mapy) = (%d, %d, %d, %d, %d, %d)\n",
        nx, ny, tilex, tiley, mapx, mapy);

putTile(screen, 3, tilex, tiley);
}
```

## playstate.cpp (1)

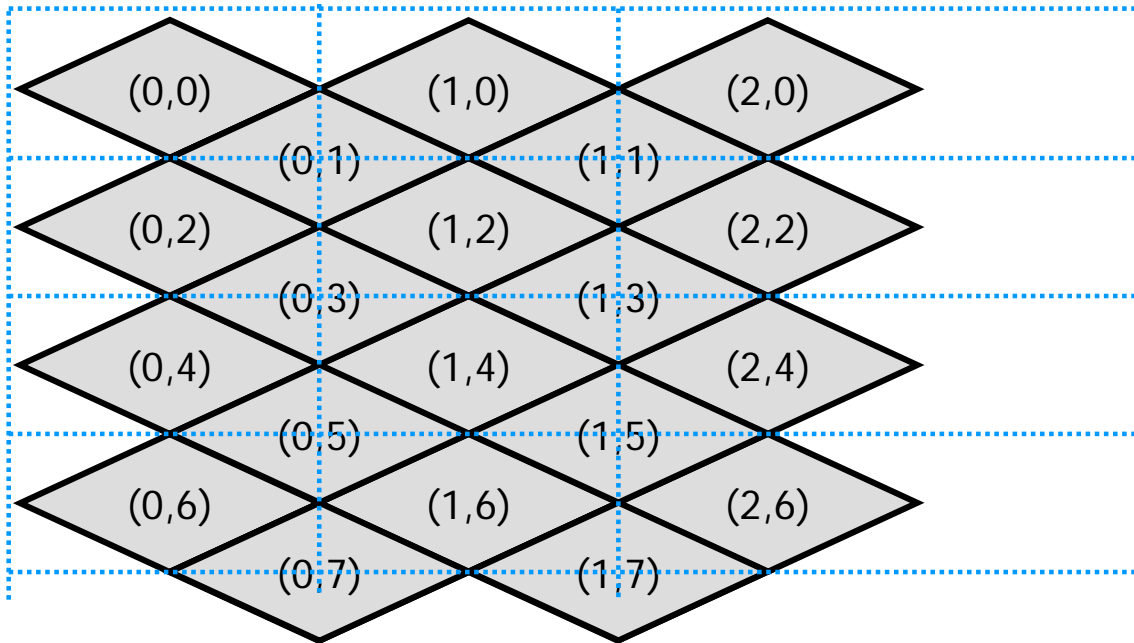
```
void CPlayState::HandleEvents(CGameEngine* game)
{
    // 전략

    case SDL_MOUSEBUTTONDOWN:
        switch (event.button.button) {
            case SDL_BUTTON_LEFT:
                mouseRect.x = event.button.x;
                mouseRect.y = event.button.y;
                mouseClicked = true;
                break;
        }
        break;
}

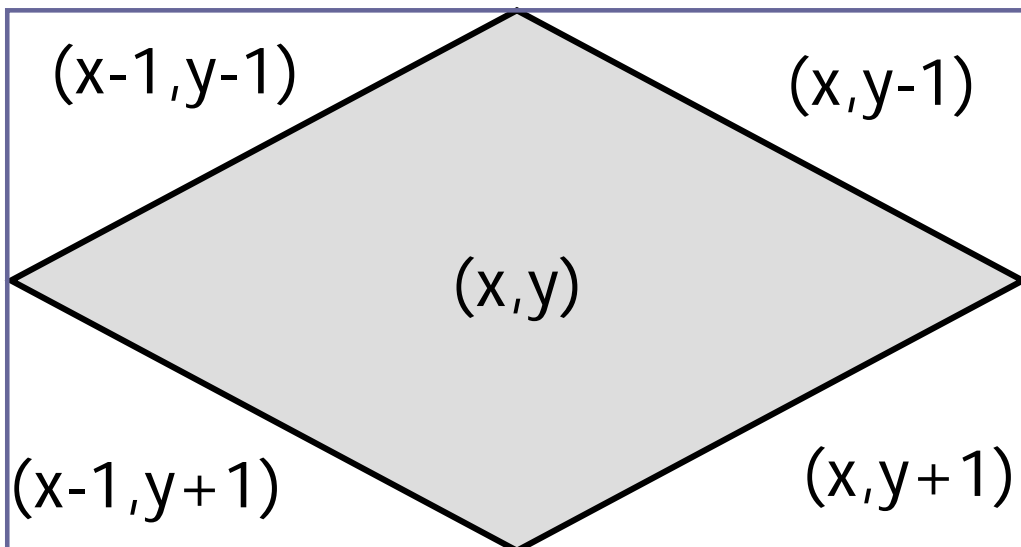
void CPlayState::Draw(CGameEngine* game)
{
    bg->draw(game->screen);
    if (mouseClicked)
        bg->putMouse(game->screen, &mouseRect);
    player->draw();
}
```

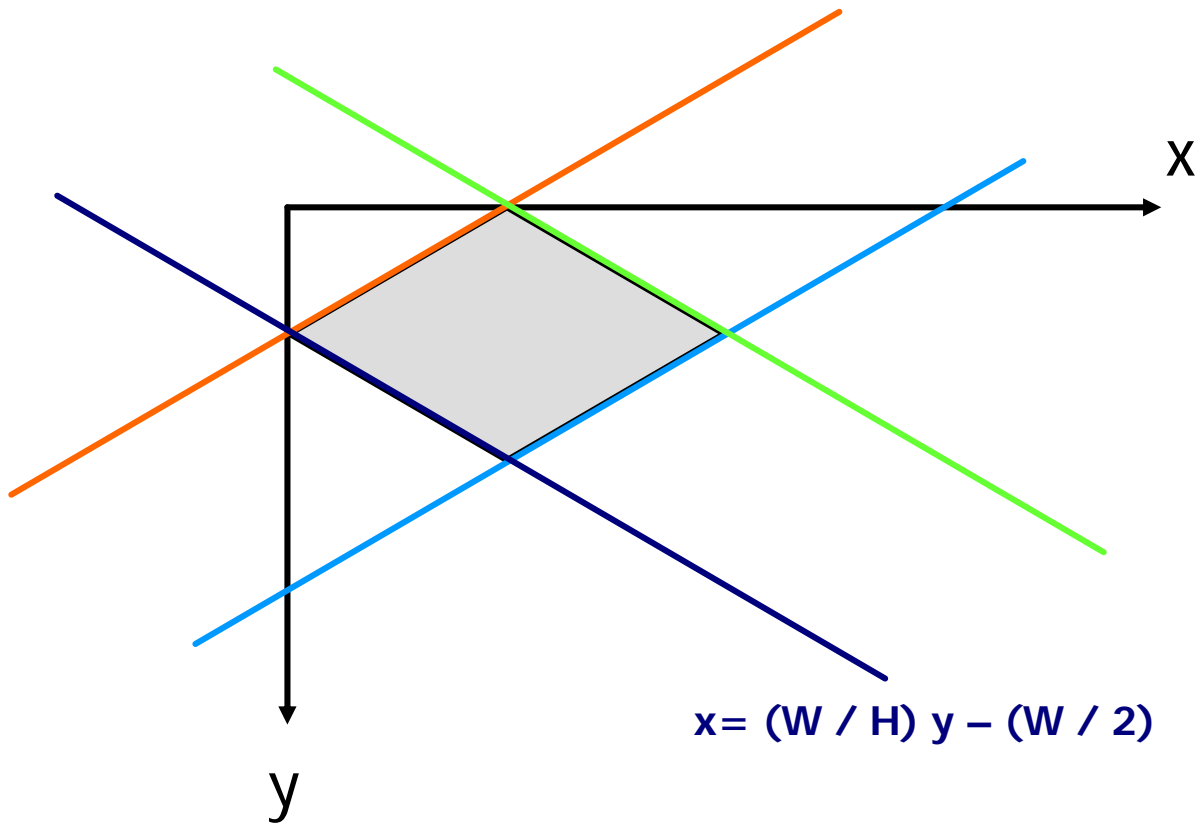
# 마우스 좌표를 타일 좌표로 변환

```
tilex = mousex / TILE_WIDTH;  
tiley = mousey / TILE_HEIGHT * 2;
```



# 타일 좌표의 보정





## 실습 과제 #11 및 과제 #4

- 동축 타일맵을 배경 위에 축구공을 배치
  - lecture13-02-dist.zip 이용. ball.bmp 사용.
  - 마우스를 클릭하여, 타일 위에 축구공을 생성.
  - 캐릭터가 조정하여, 축구공을 먹음.
  - 화면의 오른쪽 상단에 현재의 축구공의 개수를 출력.
  - 캐릭터와 공의 그림자등 간단히 표시함.

