

기초 컴퓨터 그래픽스

HW2 README

20191559 강상원

1. [환경 명세]

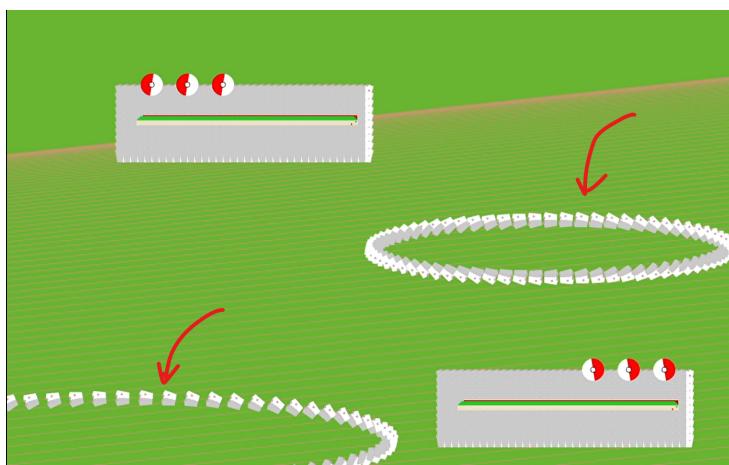
- Windows 10 64bit, Intel® Core™ i5-8257U CPU, Intel® Iris® Plus Graphics 645, Visual Studio Community 2022 Release x64

2. [요구사항]

(a) 다음 왼쪽 그림의 9개의 물체 중 5개를 선택한 후, 각 물체에 대하여 서로 다른 창의적인 동적 효과를 생성할 것. (각 물체당 최대 15점 (최대 총 75점))

1) 사용한 물체: Shirt

- 부여한 창의적인 동적 효과-1: 포켓몬 대전 시 발판 모양 회전하는 타원 구현
- 확인 방법: 프로그램 실행 시 가까이와 멀리, 총 2개의 타원이 보인다. 이로서 색깔을 바꾼 여러 개의 Shirt 모양을 회전시켜 회전하는 타원 모양을 만듦을 알 수 있다.



- 구현 방법: `ellipse_x, ellipse_y` 값을 설정하고 이 값을 각각

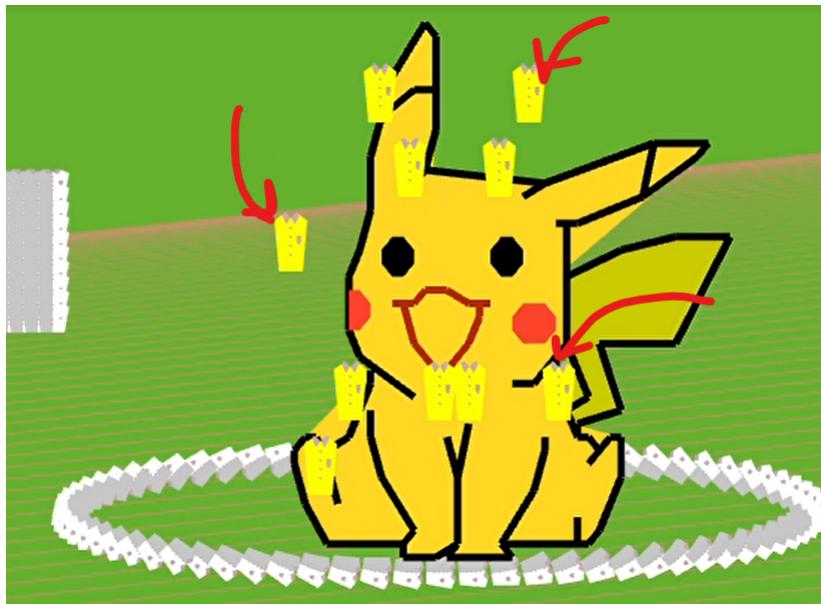
```
float ellipse_x = center_x + a * cos(angle);  
float ellipse_y = center_y + b * sin(angle);
```

으로 설정한다. 각 값을 원을 그리는 x, y 좌표에 translation 변환을 해 주면 타원으로 변

환된다. 또한 72개의 Shirt를 5도씩 회전한 모양을 계속해서 회전시켜 보다 동적인 효과를 생성한다.

- 부여한 창의적인 동적 효과-2: 찌리리공의 전기 공격 시 전기 충격(번개) 모양 효과 구현

- 확인 방법: 5번째 공격 (찌리리공의 3번째 공격) 시 노란색 Shirt 모양들이 빠르게 떨어지며 끊기는 효과가 나타나 전기 충격된 것 같은 효과를 준다.



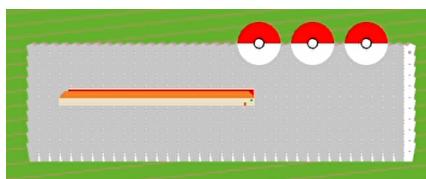
- 구현 방법:

`sin()` 함수의 값에 `ceil()`을 하여 비연속적인 끊기는 듯한 좌표 효과를 구현한다.

```
float x_pos = 150.0f + i * 20.0f; // Adjust the 20.0f value to control the distance between the shirts
float y_offset = ceil(sin(timestamp * lightning_speed + i * 1.5f) * lightning_amplitude / 20.0f) * 50.0f;

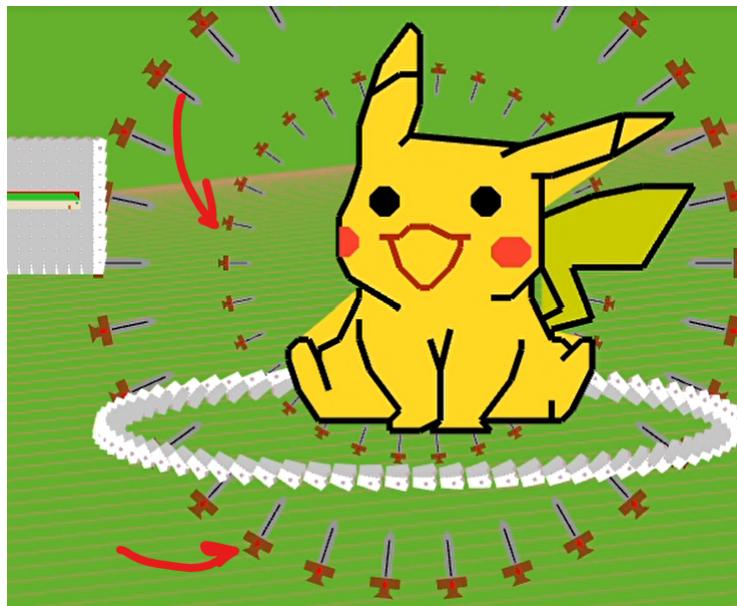
ModelMatrix = glm::translate(glm::mat4(1.0f), glm::vec3(x_pos, 100.0f + y_offset, 0.0f));
ModelMatrix = glm::scale(ModelMatrix, glm::vec3(1, 2, 1.0f));
```

번외: 각 포켓몬의 스탯창 배경도 Shirt 모양 여러 개를 이어 붙여 점선지 효과를 냈다.



2) 사용한 물체: Sword

- 부여한 창의적인 동적 효과: 여러 칼 모양이 피카츄를 향해 찌르는 듯 원형을 유지하며 떨어진다.
- 확인 방법:



첫 번째, 4번째 공격 시 피카츄를 향해 동심원을 그리며 칼들이 꽂힌다. 원은 2개에 걸쳐 있으며, 지정 시간 동안 계속 찌르기 효과가 나타난다.

- 구현 방법:

```
angle = (360.0f / 30.0f) * i;
distance = 300.0f - (8 * timestamp % 300);
distance2 = 500.0f - (8 * timestamp % 500);

float x = cos(angle * TO_RADIANT) * distance;
float x2 = cos(angle * TO_RADIANT) * distance2;
float y = sin(angle * TO_RADIANT) * distance;
float y2 = sin(angle * TO_RADIANT) * distance2;

draw_sword_with_transforms(x + 250.0f, y + 100.0f, 1.0f, 250, 100);
draw_sword_with_transforms(x2 + 250.0f, y2 + 100.0f, 2.0f, 250, 100);
```

목표 지점까지의 \sin , \cos 값을 이용하여 각 sword가 회전할 각도를 정하고, 지정 함수를 호출하여 목표 지점까지의 거리가 점점 가까워지게 함으로써 효과를 구현하였다.

3) 사용한 물체: Cocktail

- 부여한 창의적인 동적 효과: 날파리 무리가 꼬이듯이 찌리리공 주변을 정신없이 헤집는다.
- 확인 방법:



두 번째 공격 시 찌리리공 주변으로 작은 칵테일 모양들이 정신없이 날아들어 헤집는다.

- 구현 방법:

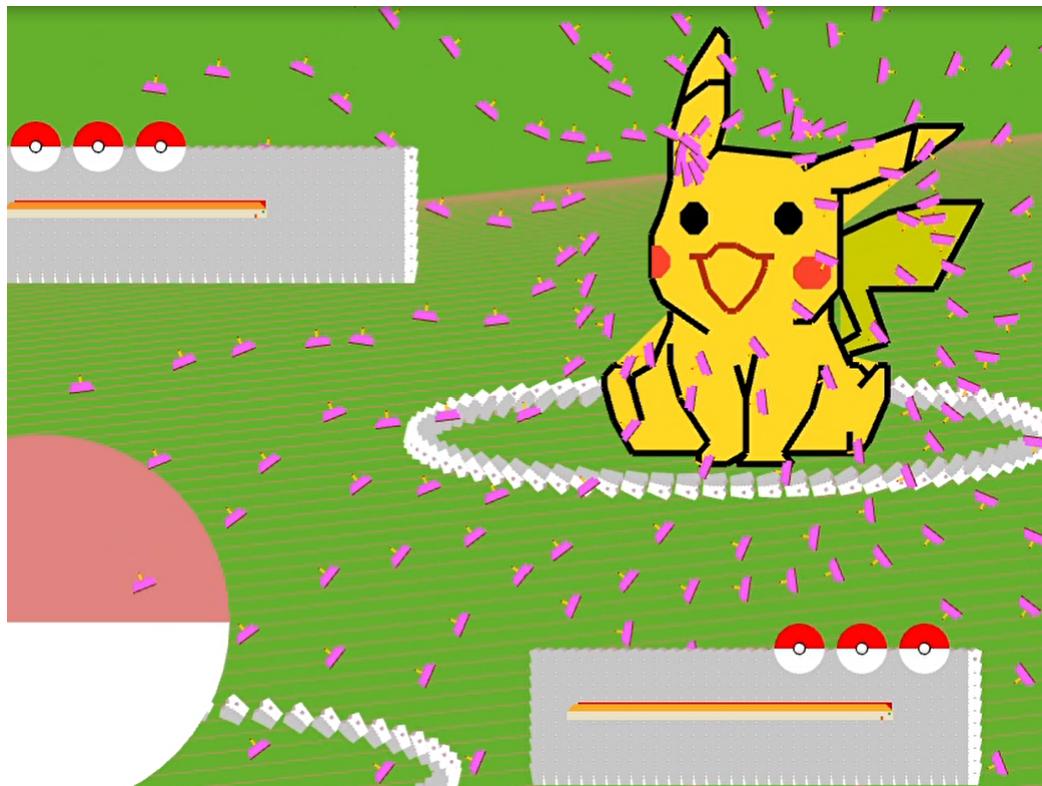
```
// Update the cocktail position
cocktail_positions[i].y -= COCKTAIL_FALL_SPEED * (1.0 + i);
if (cocktail_positions[i].y < -50.0f) {
    cocktail_positions[i].y = static_cast<float>(rand()) /
(static_cast<float>(RAND_MAX / 500.0f)) - 450;
    cocktail_positions[i].x = static_cast<float>(rand()) /
(static_cast<float>(RAND_MAX / 500.0f)) - 550;
}

// Draw the cocktail with the updated position
ModelMatrix = glm::translate(glm::mat4(1.0f),
cocktail_positions[i]);
ModelMatrix = glm::rotate(ModelMatrix, shirt_clock *
TO_RADIAN + i * TO_RADIAN, glm::vec3(0.0f, 0.0f, 1.0f));
ModelMatrix = glm::scale(ModelMatrix, glm::vec3(0.7f, 0.7f,
1.0f));
ModelViewProjectionMatrix = ViewProjectionMatrix *
ModelMatrix;
glUniformMatrix4fv(loc_ModelViewProjectionMatrix, 1,
GL_FALSE, &ModelViewProjectionMatrix[0][0]);
draw_cocktail();
```

Rand 값을 이용하여 cocktail 위치를 translation 변환하여 준다. 반복문을 실행하고, 범위를 지정해 주어 정해진 위치까지 떨어질 수 있도록 한다.

4) 사용한 물체: Cake

- 부여한 창의적인 동적 효과: 케이크 모양들이 흔들리며 피카츄 얼굴 주변을 돌아가 최면 (상태이상) 공격을 하는 듯한 모양을 취한다.
- 확인 방법:



세 번째 공격 시 케이크 모양들이 흔들리며 피카츄 얼굴 주변을 돌아가 최면 (상태이상) 공격을 하는 듯한 모양을 취한다. (상태 이상에 걸린 피카츄는 왜곡되어 흔들리게 되며, 이후 공격을 받을 때 데미지를 크게 받는다.)

구현 방법:

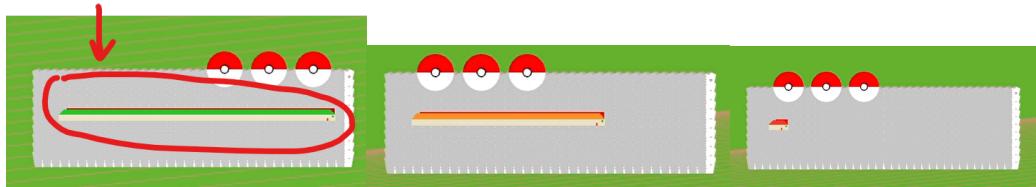
동심원을 그리는 cake 모양들에 timestamp 값에 따라 \sin 함수에 의한 oscillation 값을 변하게 하여 이 값으로 위치 translation을 한다.

```
float oscillation = sin((timestamp + i * 3) * 0.1f) * 100.0f;
```

5) 사용한 물체: House

- 부여한 창의적인 동적 효과: 포켓몬의 공격에 따라 데미지를 입은 포켓몬의 HP가 줄어드는 효과가 나타난다. 남은 HP가 적을수록 색깔이 초록색에서 빨간색으로 바뀐다.
- 확인 방법: 매번 포켓몬의 공격 이후 포켓몬이 빨간색으로 깜박이며 데미지를 입을 때

에 여러 개의 House 모양을 나열하여 만든 HP 막대가 변화한다.



구현 방법:

```
int num_houses_to_draw = (int)(10 * loading_percentage);

// Draw the houses with a constant horizontal spacing
for (int i = 0; i < num_houses_to_draw; ++i) {
    // Calculate the x-coordinate for each house
    float house_x = -325.0f + i * house_spacing;

    // Set the model matrix for the house position and scaling
    ModelMatrix = glm::translate(glm::mat4(1.0f),
        glm::vec3(house_x, 150.0f, 0.0f));
    ...
    draw_house();
}
```

현재 포켓몬의 HP 값에 따라 그릴 House 모양의 개수를 결정하고, 일정한 간격으로 그려 상태 막대 모양을 만든다. HP 값이 줄어들 때, House 모양 색상의 R 값을 증가시키고, G 값을 감소시켜 초록색에서 점점 빨간색이 되게 한다.

(b) 충분히 복잡도가 있는 2차원 기하 물체를 자신이 한 개 모델링 한 후, 위의 물체들처럼 창의적인 동적 효과를 생성하라. (최대 총 25점)

- 부여한 창의적인 동적 효과-1: 피카츄가 처음 대전장에 등장할 때, 크기가 점점 커지면서 나타난 후 세 번 점프하여 생동감을 준다.
- 확인 방법: 큰 모양의 포켓볼이 굴러간 이후 처음 포켓몬 대전이 시작될 시, 포켓몬들이 커지면서 나타나고 점프한다.

- 구현 방법:

```
// Update animation state
// Increase the scale until the desired size is reached
if (time_table[0].second <= timestamp) {
    if (pikachu_scale < 8.0f) {
        pikachu_scale += 0.3f;
    }
    else {
        // Make Pikachu jump three times
        if (jump_count < 4) {
```

```

        jump_height += value * 3.0f; // Adjust the speed
of the jump by changing the multiplier
        if (jump_height >= 20.0f) { // Adjust the
maximum jump height
            value = -value;
            jump_count++;
        }
        else if (jump_height <= 0.0f) {
            value = -value;
            jump_height = 0.0f;
        }
    }
}

```

timer 함수에서 점프 횟수를 세어 세 번 동안 점프를 하게 하고, scale 변수를 지정한 크기까지 시간에 따라 커지게 한다.

- 부여한 창의적인 동적 효과-2: 피카츄가 공격을 받을 때, 흔들리며 데미지를 받는 듯한 효과를 준다.
- 확인 방법: 피카츄가 공격 효과를 받고 있을 시, 상하좌우로 흔들리는 효과를 볼 수 있다.
- 구현 방법:

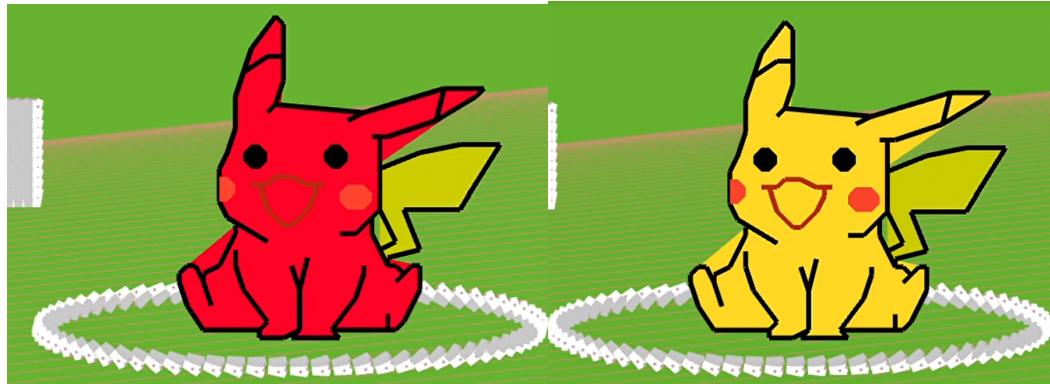
```

// Calculate the trembling effect
float tremble_amplitude = 1.0f; // Adjust this value to control the
trembling intensity
float tremble_frequency = 3.0f; // Adjust this value to control the
trembling speed
float tremble_offset = sin(timestamp * tremble_frequency) *
tremble_amplitude;

```

삼각함수를 이용하여 tremble_offset 값이 timestamp에 따라 변하게 만들고, 상하좌우 transition을 통해 떠는 듯한 효과를 만들었다.

- 부여한 창의적인 동적 효과-3: 피카츄가 공격을 받은 후 react 할 때, 붉은색으로 깜박이며 데미지를 받았음을 시각적으로 표현한다.
- 확인 방법: 피카츄가 공격을 받은 후, 붉은색으로 깜박이는 효과를 관측할 수 있다.



- 구현 방법:

timestamp 값에 따라 삼각함수 offset 크기를 비교하여 색깔 값을 바꿔 깜박이게 만든다.

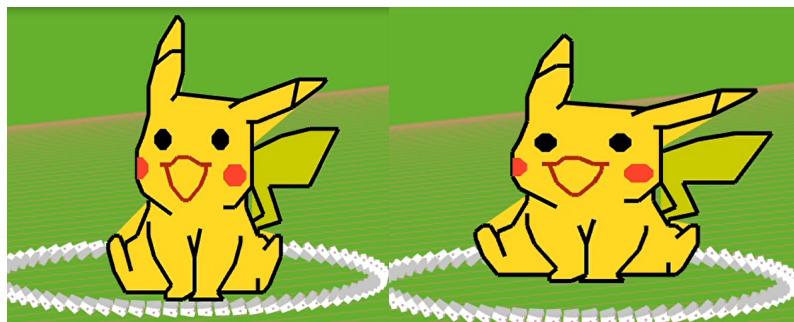
```

if (in_time(FIRST_RESP) || in_time(THIRD_RESP) || in_time(FIFTH_RESP))
{
    float blink_factor = sin(timestamp * 0.2f) * 0.5f + 0.5f;
    if (blink_factor > 0.5f) {
        pikachu_color[0][1] = 0;
    }
    else {
        pikachu_color[0][1] = 217 / 255.0f;
    }
}
else pikachu_color[0][1] = 217 / 255.0f;

```

- 부여한 창의적인 동적 효과-4: 피카츄가 상태 이상 공격을 받은 이후, 왜곡되게 상하좌우가 늘어나며 흔들린다. (어지러움 탐)

- 확인 방법: 피카츄가 어지러움 공격(3번째 공격)을 받으면서 상태 이상에 걸려 아래와 같이 왜곡되게 흔들린다.



- 구현 방법:

```

float spring_bounce = sin(timestamp * 0.05f) * 0.1f + 1.0f;

if (timestamp > time_table[THIRD_ATK].first) ModelMatrix =
glm::scale(ModelMatrix, glm::vec3(pikachu_scale * spring_bounce,
pikachu_scale / spring_bounce, pikachu_scale));

```

피카츄의 가로 scale, 세로 scale 값을 위와 같이 조정하여 효과를 구현하였다.

번외)

- 피카츄가 5번째 공격을 받고 HP가 모두 고갈되면 쓰러지는 효과(똑 떨어짐)가 나타난다.
- 전기 충격 공격을 받을 시, 이미 상태 이상 효과가 걸린 상태에서 떨림 효과가 중첩되어 표현된다. 이후 빨간색으로 깜박일 때도 왜곡 효과와 함께 일어난다.

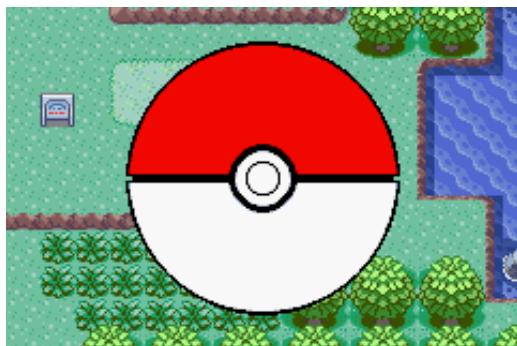
(c) 제출한 숙제 중 가장 재미있고 복잡도가 있는 기하 변환 및 애니메이션 효과를 생성한 학생을 적절히 선정하여 최대 25점까지 추가 점수를 부여할 수 있음 (정확한 숫자는 상황에 따라 유동적임). 공정을 기하기 위하여 컴퓨터 그래픽스 연구실 대학원생들이 공동으로 심사함.

- 추가 점수를 부여 받아야 할 이유

(복잡한 기하 변환과 재미있는 애니메이션 효과 사용과 확인 방법 등을 서술)

추억의 포켓몬 게임 대전 과정을 OpenGL로 재창조해보았다.

처음에 큰 포켓볼이 굴러가며 대전장이 나오는 효과도 구현하였고,



처음에 자세히 보면 상태창 위의 포켓볼들도 미끄러져 굴러서 들어온다.

뒤의 line들의 간격을 조합하여 배경의 깊이감 효과를 주었으며, 추가로 구현한 피카츄 모양은 5가지 이상의 서로 다른 움직임을 보여준다.

각 포켓몬들의 특성에 맞는 공격(ex. 전기 공격) 모양을 만들었으며, 한 번에 하나의 효과만이 나타나는 것이 아니라 실시간으로 공격을 받으면서 떨리고, 깜박이면서 HP 상태가 줄어들며 색깔이 바뀌는 등 역동적으로 표현하고자 노력하였다.

피카츄 대전장의 받침대 모양도 단순한 타원 모양에서 좀 더 동적으로 표현하기 위해 빠

르게 회전하는 여러 개의 Shirt 모양으로 바꾸었으며 한 도형으로 여러 모양과 효과를 나타내었다. (ex. 받침대, 번개 모양, 상태창, ..)



↑본 과제에서는 위와 반대의 상황: 지우의 포켓몬(찌리리공)이 야생의 포켓몬을 만난 상황을 구현하였다.