The Revenge of Rowlet

Group members: 111550061 邱冠崴、111550066 王裕昕、111550148 陳冠達

- Introduction

You are now a Pokémon Trainer, embarking on an adventure through a scorching magma-filled region. Suddenly, the embodiment of the land itself—Groudon—emerges, blocking your path. It's clear that a fierce battle is inevitable. Determined, you call forth your strongest Pokémon—Rowlet… wait, what?
Rowlet turns its head to glance at you, utterly bewildered. However, there's no turning back now. With no other choice, Rowlet makes up its mind to fight with Groudon. An epic battle is about to start!!!

- Implementation detail
  - ➢ Timing Control

  We use the function glfwGetTime() to get the current time. The initial time is stored and used to estimate how much time has passed by calculating (currentTime - initTime).

  - ➢ Camera Rotation Opening

  For the opening of the epic battle, we rotate the camera view to create an atmosphere of an arena.

```
else if ((currentTime - initTime < 4.0f)) {
    // rotate camera
    camera.rotationY += max(0.5f, (6.0f - (currentTime - initTime)) * (6.0f - (currentTime - initTime)) / 18.0f);
}
else if (currentTime - initTime < 9.0f) {
    // camera view set to 30 degree
    if (abs(camera.rotationY - 30.0f) > 0.5f) {
        camera.rotationY += 0.5f;
    }
    else {
        // rowlet y-axis oscillation
        Rowlet.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
    }
}
```

  We rotate the camera along the Y-axis and let the rotating speed be proportional to the square of (6 – (currentTime – initTime)), this creates an effect of decreasingly slower rotation. After the rotation, we want to ensure that the camera is at the right position, so we continue to rotate the camera until it is sufficiently near 30 degree. Also, we apply oscillation along Y-axis to Rowlet as if it is flying in the air.

  - ➢ Rowlet Turn and Glance

```
else if ((currentTime - initTime < 10.0f) && Rowlet.rotation.y > -15.0f) {
    // rowlet turn around to stare
    Rowlet.rotation.y -= 1.0f;
}
else if (currentTime - initTime < 10.0f) {
    // rowlet stop and stare
}
else if (currentTime - initTime < 11.0f) {
    // rowlet turn back
    if (abs(Rowlet.rotation.y - 90.0f) > 0.5f) {
        Rowlet.rotation.y += 1.5f;
    }
    else {
        // rowlet y-axis oscillation
        Rowlet.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
    }
}
```

  To turn Rowlet around, we rotate Rowlet along Y-axis until the rotation degree reaches -15° (345°). And then we turn the Rowlet back to its initial degree (90°). After that, continue applying the Y-axis oscillation for flying effect.

- Rowlet Skill “Tackle”

```
else if (currentTime - initTime < 12.0f) {
    // rowlet attack
    Rowlet.position.z -= 0.15f;
    Rowlet.rotation.x = 30.0f;
}
else if (currentTime - initTime < 12.5f) {
    // rowlet stop and groudon mock
    // Groudon fast oscillation along x-axis
    Groudon.position.x += 0.2f * sin(10.0f * (currentTime - initTime));
}
else if (currentTime - initTime < 13.5f) {
    // rowlet return
    Rowlet.position.z += 0.15f;
}
else if (currentTime - initTime < 14.0f) {
    // waiting for groudon's turn
}
```

In the first round, Rowlet uses the skill “Tackle”. We decrease the Z-axis position so that it runs toward Groudon and set the rotation along X-axis to 30 degree to create a dashing effect. After Rowlet hits Groudon, the Groudon shakes horizontally, which is implemented with oscillation along X-axis. And then Rowlet move backwards to its original position.

- Groudon Skill “Dragon Tail”

```
else if (currentTime - initTime < 15.0f) {
    // groudon attack
    Groudon.position.z += 5.0f;
    Groudon.rotation.y -= 1.0f;
    Rowlet.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
}
else if (currentTime - initTime < 15.5f) {
    // groudon stop and rowlet fall
    RowletDeltax -= 0.6f;
    RowletDeltay -= 0.6f;
}
else if (currentTime - initTime < 16.5f) {
    // groudon returns and rowlet arises
    Groudon.position.z -= 5.0f;
    Groudon.rotation.y += 1.0f;
    RowletDeltax += 0.3f;
    RowletDeltay += 0.3f;
}
```

After Rowlet’s attack, it’s now time for Groudon to fight back! Groudon uses the skill “Dragon Tail”, which is to slam the opponent with its strong tail. While increasing the Z-axis position so Groudon moves towards Rowlet, we rotate Groudon along Y-axis so that Groudon hits Rowlet with its tail. After Groudon hits Rowlet, we rotate Rowlet along Y-axis and then Z-axis to make the Rowlet fall under Groudon’s attack. Finally, Groudon and Rowlet both return to their original position.

- Rowlet Evolution

```
else if (currentTime - initTime < 18.5f) {
    // shimmering rowlet
    Rowlet.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
    Rowlet.rotation.y += 1.0f * (currentTime - initTime);
}
else if (currentTime - initTime < 19.5f) {
    // show the evolved rowlet
    Rowlet_evolve.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
}
```

```
shaderPrograms[6]->use();
shaderPrograms[6]->set_uniform_value("model", RowletModel);
shaderPrograms[6]->set_uniform_value("view", view);
shaderPrograms[6]->set_uniform_value("projection", projection);
if (currentTime - initTime - 16.5f > 0) {
    shaderPrograms[6]->set_uniform_value("shimmer", 0.5f + 0.5f * sin((currentTime - initTime - 16.5f) * 5.0f));
}
else {
    shaderPrograms[6]->set_uniform_value("shimmer", 0.0f);
}
```

After being hit, Rowlet is now pissed off. Rowlet turns the anger into its power and starts to evolve. While evolving, we make Rowlet rotate along Y-axis faster and faster. Also, we make Rowlet shimmer with golden light by setting the uniform value to a sine value related to currentTime – initTime.

- Pokeball Parabola

```
else if (currentTime - initTime < 20.5f) {
    // rotate to throw the bomb pokeball
    Rowlet_evolve.rotation.x -= 0.3f;
    Pokeball.position.z -= 0.01f;
    Pokeball.position.y += 0.01f;
    ballInitialPosition = Pokeball.position;
    throwStartTime = currentTime;
}
else if ((currentTime - initTime < 35.5f) && Pokeball.position.y > 0.0f) {
    // rotate the rowlet back
    if (currentTime - initTime < 21.5f) {
        Rowlet_evolve.rotation.x += 0.3f;
    }
    else {
        Rowlet_evolve.position.y += 0.01f * sin(2.0f * (currentTime - initTime));
    }
    // throw the bomb pokeball
    float t = currentTime - throwStartTime;
    Pokeball.position.x = ballInitialPosition.x;
    Pokeball.position.y = ballInitialPosition.y + ballVelocity.y * t - 0.5f * gravity * t * t;
    Pokeball.position.z = ballInitialPosition.z + ballVelocity.z * t;
}
```

After evolution, Rowlet is now in anger mode and holding a pokeball-shaped bomb. The evolved Rowlet is a new Rowlet model that is rendered when the currentTime – initTime > 19.5, and the original Rowlet model is not rendered afterwards. The pokeball-shaped bomb is only rendered when currentTime – initTime > 19.5.

The evolved Rowlet throws the bomb toward Groudon to show its unstoppable fury. To throw the bomb, we rotate Rowlet along X-axis and also move the pokeball-shaped bomb a little to match the movement of Rowlet. Then we rotate the Rowlet back to original position and start to calculate the parabola of the bomb. The initial velocity of the bomb is (0, 15, -12) and the gravitational acceleration is 9.8. The variable t is to subtract currentTime with throwStartTime(the time when the ball is thrown out). The new Y_position is calculated by initial Y_position + Y_velocity * t – 0.5 * g * t^2. The new Z_position is to add the initial Z_position with Z_velocity * t. For the pokeball-shaped bomb model, we apply additional translation to match the new Y_position and Z_position. After the pokeball-shaped bomb hits Groudon, the bomb is no longer rendered and Groudon starts to explode upon hit.

- Groudon Explosion (geometry shader)

```
2   layout (triangles) in;
3   layout (triangle_strip, max_vertices = 3) out;
4
5   in VS_OUT{
6       vec2 texCoords;
7   } gs_in[];
8
9   out vec2 TexCoords;
10  uniform float time;
11
12  vec4 explode (vec4 position, vec3 normal){
13      float magnitude = 40.0;
14      vec3 direction = normal * time * magnitude;
15      return position + vec4(direction, 0.0);
16  }
17
18  vec3 GetNormal(){
19      vec3 a = vec3(gl_in[0].gl_Position) - vec3(gl_in[1].gl_Position);
20      vec3 b = vec3(gl_in[2].gl_Position) - vec3(gl_in[1].gl_Position);
21      return normalize(cross(a, b));
22  }
23
24  void main(){
25      vec3 normal = GetNormal();
26      for(int i=0 ; i<3 ; i++){
27          gl_Position = explode(gl_in[i].gl_Position, normal);
28          TexCoords = gs_in[i].texCoords;
29          EmitVertex();
30      }
31
32      EndPrimitive();
```

  - Line 2-3 define the input and output primitive types for the geometry shader. The input is a set of triangles, and the output is a triangle strip with a maximum of 3 vertices.
  - Line 5-7 defines VS_OUT that contains texture coordinates (texCoords). The array gs_in holds this data for each vertex of the input triangle.
  - Line 9 declares the output that will be passed to fragment shader.
  - Line 10 is the uniform variable used to control the explosion effect.
  - Line 12-16 is the explode function that calculates the new position of a vertex by moving it along its normal vector. The amount of movement is controlled by the time variable and a magnitude factor. Finally it returns the new position of the vertex.
  - Line 18-22 is the function GetNormal that calculates the normal vector of the input triangle. The result is the normalized cross product of two edges of the triangle.
  - Line 24-30 calculates the normal vector of the input triangle and then processes each vertex of the triangle: It calls the explode function to calculate the new position of the vertex and sets the texture coordinates.

- Discussion

  The Proposer(邱冠崴) initially suggested recreating a popular meme by having Rowlet battle Groudon, incorporating flashy effects like Groudon using Fire Blast or Solar Beam to enhance the visual richness. The Critic(王裕昕) pointed out that using such moves could make the overall scene overly complex. Additionally, the implementation difficulty might result in it being challenging to realistically replicate the effects seen in animations. Moreover, with only 30 seconds available for the scene, including these moves might cause the runtime to exceed the limit.

The Negotiator(陳冠達) synthesized the opinions of the Proposer and Critic, agreeing that the idea of Rowlet battling Groudon is both interesting and feasible. However, the moves could be simplified by using physical attacks instead. For instance, Groudon could use Dragon Tail while Rowlet uses Tackle. These moves can be achieved more easily by leveraging simple translate and rotate to create the battle effects. Furthermore, the Negotiator suggested adding a twist—at the end of the battle, Rowlet evolves. This can be achieved using the material-changing techniques learned in previous assignments. The final surprise could feature Rowlet using a "Pokéball Grenade" to defeat Groudon, resulting in an unexpected and humorous ending.

- Work assignment
  Find model and texture: 邱冠崴、陳冠達
  Code Implementation: 邱冠崴、王裕昕、陳冠達
  Video Record: 王裕昕、陳冠達
  Report: 邱冠崴、王裕昕

- Results
  Video link: https://www.youtube.com/watch?v=ee2r4-YMxHo

- References
  Rowlet model:
  https://www.models-resource.com/3ds/pokemonsunmoon/model/18589/
  https://sketchfab.com/3d-models/rowlet-4cb5ae6644f4486b9290c838a2d645ad
  Groudon model:
  https://www.models-resource.com/3ds/pokemonxy/model/10579/
  Cube mapping texture:
  https://gamebanana.com/mods/7191