Project Dewpsi Design Master Document

All Sprites

Project Dewpsi Design Document

Sprite #000 : Shadow Fly Trap

Synopsis

The sprite described herein uses a derivation of the C++ class `Sprite`, and it omits any kind of movement because of there being no speed calculations. Described below is basic information about the sprite.

Sprite Information

Graphics:	shadowflytrap.png
Hitbox:	Default State: (0, 0) 26x50
	Attack State: (3, 8) 25x27
States:	Withdraw, Default, Attack, Charge, Die
HP:	3
X Speed:	0
Y Speed:	0

Notes: The horizontal and vertical speeds are zero because the sprite is not meant to move.

Base Stats

Attack:	1
Defense:	0
Evasion:	10%
Accuracy	100%

Sprite States

Withdraw State

The sprite is in this state when not fighting the player. It is shown hidden in its bud until the player gets too close to it. The player doesn't collide with the sprite, per se, but the sprite has a rectangular detection field that stretches out to the left and right, 64 pixels away from its center point; and the top of that box stretches 16 pixels upward. The sprite goes to the Default state if the player is in this box.

Default State

In this state the sprite has a timer and a counter. If the timer (*t*) is less than zero it is set to a value according to this formula: $t = 90 + 15c$, where *t* is the new value for timer_wait, and where *c* is the value of the counter. Said counter (*c*) increments every time the sprite's active hitbox hits the player in his Defend State while it is in its Attack state. If *c* is more than a certain value (constant *m* [for max]), then the sprite enters its Charge state; otherwise, when *t* decrements to zero, it enters the Attack State. Before the sprite leaves this state, *t* should be equal to -1 and *c* should be equal to 0.

Attack State

In this state the sprite activates its hitbox for attacking (see page 1). Prior to this state, *t* is set to a certain value dictated by c_1. Also, the hitbox should be set to what's shown on page 1. (Click here). While *t* is greater than zero, if it's greater than c_2, then set the sprite's clipping rectangle to the first attack frame as described here. In the case that *t* is less than or equal to c_2, set the sprite's clipping rectangle to show the second attack frame, as described in the section linked above.

Note: for the values labeled above, see the table below.

Charge State

In this state the sprite charges an attack that is stronger than normal. A short animation plays along with an external effect that conveys energy being concentrated. The effect of this state is that a temporary Charge status effect is placed on the sprite.

Die State

Preface: Prior to this state, the sprite's death animation timer should be set.
In this state, a death animation timer is presumably greater than zero. A special kind of sprite, which I will call a particle sprite (working title).

Notes

This sprite uses a state machine in order to segment off five different parts of the code. This is done using a jump table. In C++ this entails an array of statically defined function pointers to the aforementioned parts. However, I couldn't make this work, so for right now I'm using a `switch` statement. Switch statements could be optimized by the compiler to become jump tables – that depends on how the statement in question is written. In my case it should work out that way.

Values For *t* (Attack State)

Description	**Value**	**Difference**
Initial value (c_1)	55	0
Frame 2 to 0 (c_2)	Frame 30	25
Frame 0 to 1 (C_3)	Frame 20	10

Graphics

Withdraw, Default, Charge, and Die states. The frame size is 38x80. Frames 0 – 2 are the idle animation that’s shown in the Default state. Frames 3 – 5 are a variation of that animation, where the sprite’s mouth is open instead of shut. Frame 6 shows only the bud – this is what the sprite looks like in its Withdraw state.

Attack state. The frame size is 76x80. Each frame’s Y offset for the clipping rectangle is 160. The X offsets, are for each frame: 0 for frame 0, 76 for frame 1, and 152 for frame 2.

The following table showcases the location of each frame and what animation it belongs to.

Default Frames	
These frames are for the animation of the sprite leaning forward by bending its stalk. Its mouth is closed.	Frame 000
	Frame 001
	Frame 002
These frames depict the animation of the sprite leaning forward with its mouth open, same as the former.	Frame 003
	Frame 004
	Frame 005
This frame just shows the bud.	Frame 006
Attack Frames	
Lunging forward, its mouth still wide open.	Frame 000
Lunging forward, its now closed.	Frame 001
Leaning back to pounce.	Frame 002

…

Collision With the Player

There are basically two hitboxes the sprite can choose from. In the Default Charge states, the hitbox is this: (0, 0) 26x50. In the Attack state it is this: (3, 8) 25x27. How those boxes interact with the player depends on the state the sprite is in as well as the player.

Project Dewpsi Design Document

Base Stats

Every sprite that has the capacity to hurt the player have what are known as base stats. Base stats are the following:

- Attack
- Defense
- Evasion
- Accuracy

The base stats are called such because they are the default values a sprite would have when they are not affected by status effects.