```cpp
#include "Bat.h"
#include "SDL_image.h"
#include "Window.h"

Uint8 Bat::bat_count = 0;

Bat::Bat() {
    bat_count = bat_count + 1;
    batTx = NULL;
    appearance.x = Window::getInstance()->getWindowSizeW() + rand() % 100;
    appearance.y = (Window::getInstance()->getWindowSizeH() / 5) + (32 *
      bat_count);
    appearance.w = 32;
    appearance.h = 32;
    z_index = rand() % 100 + 1;
    if (z_index > 80) {
        z_index = 2;
    }
    else {
        z_index = 1;
    }
    ticked = 0;
    curr_tx = 0;
    flipped = false;
    speed = rand() % 3 + 1;
};

Bat::~Bat() {
    SDL_DestroyTexture(batTx);
    bat_count = bat_count - 1;
};

int Bat::getZIndex() {
    return z_index;
};

void Bat::restart() {
    appearance.x = Window::getInstance()->getWindowSizeW() + rand() % 100;
    appearance.y = (Window::getInstance()->getWindowSizeH() / 5) + (32 *
      bat_count);
    flipped = false;
    ticked = 0;
    curr_tx = 0;
};

bool Bat::loadMedia() {
    bool success = true;

    //Load Bat texture
    batTx = loadTexture("assets/sprite_sheets/bat/sheet_bat_fly.png");
    if (batTx == NULL) {
        printf("Failed to create bat texture. SDL Error: %s\n",
          SDL_GetError());
```

```
51          success = false;
52      }
53
54      return success;
55  };
56
57  void Bat::renderFlipped() {
58      SDL_Rect clip_rect;
59
60      //Set up Clip Rectangle
61      clip_rect.x = curr_tx * 32;
62      clip_rect.y = 0;
63      clip_rect.w = 32;
64      clip_rect.h = 32;
65
66      SDL_RenderCopyEx(Window::getInstance()->getRenderer(), batTx,
          &clip_rect, &appearance, 0.0, NULL, SDL_FLIP_HORIZONTAL);
67  };
68
69  void Bat::renderUnflipped() {
70      SDL_Rect clip_rect;
71
72      //Set up Clip Rectangle
73      clip_rect.x = curr_tx * 32;
74      clip_rect.y = 0;
75      clip_rect.w = 32;
76      clip_rect.h = 32;
77
78      SDL_RenderCopy(Window::getInstance()->getRenderer(), batTx,
          &clip_rect, &appearance);
79  };
80
81  void Bat::flipIfNecessary() {
82      //If bat reached the left side of the window
83      if ((appearance.x <= (-1 * appearance.w)) && (!flipped)) {
84          flipped = true;
85          z_index++;
86          if (z_index > 2)
87              z_index = 1;
88      }
89      else if ((appearance.x >= Window::getInstance()->getWindowSizeW() +
          appearance.w) && (flipped)) {
90          flipped = false;
91          z_index++;
92          if (z_index > 2)
93              z_index = 1;
94      }
95  };
96
97  void Bat::render() {
98      //render appropriately
99      if (flipped) {
100         renderFlipped();
```

```
101     }
102     else {
103         renderUnflipped();
104     }
105 };
106 
107 void Bat::tick() {
108     //Tick Bat
109     ticked++;
110 
111     //Move Bat
112     if (!flipped)
113         appearance.x = appearance.x - speed;
114     else
115         appearance.x = appearance.x + speed;
116 
117     //Change textures to create the animation
118     if (ticked == speed + 5) {
119         curr_tx++;
120         if (curr_tx >= 4) {
121             curr_tx = 0;
122         }
123         ticked = 0;
124     }
125 
126     //flip if necessary
127     flipIfNecessary();
128 };
```