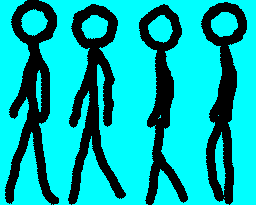
Như đã nói ở bài 11, animated sprites là việc tận dụng sprite sheet và tọa độ để tạo hiệu ứng động.

Còn **Vertical Synchronization** hay **Vsync** là công nghệ đồng bộ hóa khả năng dựng hình của GPU với tần số quét màn hình. Từ đó giới hạn FPS để đảm bảo khung hình mượt mà, ko bể ảnh (như kiểu chưa vẽ xong frame 1 đã phải vẽ frame 2). Tuy nhiên tốc độ phản hồi vs bàn phím và chuột sẽ chậm hơn. Và cần đảm bảo GPU đủ mạnh, tần số >= màn hình.

Nói ngắn gọn, hoạt hình chỉ là show một ảnh ngay sau một ảnh khác để tạo hiệu ứng chuyển động. Ở đây, chúng ta sẽ show các sprite khác để tạo hoạt hình người que.



Vì ảnh trong SDL 2 cơ bản là SDL\_Texture nên hoạt ảnh trong SDL chỉ là show các phần khác nhau của 1 texture liên tiếp.

|  |
| --- |
| // Walking animation  const int WALKING\_ANIMATION\_FRAMES = 4;  SDL\_Rect gSpriteClips[WALKING\_ANIMATION\_FRAMES];  LTexture gSpriteSheetTexture; |

* Vì thế ở đây chúng ta có sprite sheet cho chúng ta dùng làm hoạt ảnh

|  |
| --- |
| // Create vsynced renderer for window  gRenderer = SDL\_CreateRenderer(gWindow, -1, SDL\_RENDERER\_ACCELERATED | SDL\_RENDERER\_PRESENTVSYNC);  if (gRenderer == NULL)  {      printf("Renderer could not be created! SDL Error: %s\n", SDL\_GetError());      success = false;  } |

* Từ nay, chúng ta sẽ dùng Vsync. VSync cho phép render để update cùng lúc như khi monitor của bạn updates trong lúc làm mới theo chiều dọc. Trong bài này, chắc chắn hoạt ảnh sẽ không chạy quá nhanh. Phần lớn monitor chạy khoảng 60 fps và đó là giả định chúng tôi đưa ra ở đây. Nếu bạn có một tỉ lệ làm mới màn hình khác, điều đó giải thích tại sao hoạt ảnh chạy quá nhanh hay quá chậm.

|  |
| --- |
| bool loadMedia()  {      // Loading success flag      bool success = true;      // Load sprite sheet texture      if (!gSpriteSheetTexture.loadFromFile("foo.png"))      {          printf("Failed to load walking animation texture!\n");          success = false;      }      else      {          // Set sprite clips          gSpriteClips[0].x = 0;          gSpriteClips[0].y = 0;          gSpriteClips[0].w = 64;          gSpriteClips[0].h = 205;          gSpriteClips[1]…          gSpriteClips[2]…          gSpriteClips[3]…      }      return success;  } |

* Sau khi chúng ta load sprite sheet xong, chúng ta sẽ xác định sprite cho từng khung hình của hoạt ảnh.

|  |
| --- |
| // Main loop flag  bool quit = false;  // Event handler  SDL\_Event e;  // Current animation frame  **int frame = 0;**  // While application is running  while (!quit)  { |

* Trước main loop, ta khai báo biến để theo dõi khung hình hiện tại của hoạt ảnh.

|  |
| --- |
| // Render current frame  SDL\_Rect \*currentClip = &gSpriteClips[frame / 4];  gSpriteSheetTexture.render((SCREEN\_WIDTH - currentClip->w) / 2, (SCREEN\_HEIGHT - currentClip->h) / 2, currentClip);  // Update screen  SDL\_RenderPresent(gRenderer); |

* Sau khi màn hình được clear trong main loop, chúng ta sẽ render frame hiện tại của hoạt ảnh.
* Hoạt ảnh chạy từ 0 đến 3. Vì chỉ có 4 khung hình nên chúng tôi sẽ giảm tốc độ của hoạt ảnh xuống một chút. Đây là lý do tại sao khi chúng ta lấy sprite được cắt, chúng ta sẽ chia khung hình cho 4. Cách này thì khung hình sẽ lặp lại sau mỗi 4 khung hình.
* Sau khi lấy được sprite hiện tại thì ta sẽ render nó lên màn hình.

|  |
| --- |
| // Go to next frame  ++frame;  SDL\_Delay(100);  // Cycle animation  if (frame / 4 >= WALKING\_ANIMATION\_FRAMES){      frame = 0;  } |

* Bây giờ, để khung hình được cập nhật, ta cần tăng frame sau mỗi mỗi khung hình. Nếu chúng ta không làm thì hoạt ảnh sẽ dừng mãi ở khung hình đầu tiên.
* Chúng tôi muốn hoạt ảnh lặp lại nên khi khung hình đạt đến cuối thì chúng tôi cho nó trở lại 0 để hoạt ảnh bắt đầu lại từ đầu.