Một thứ khác ta có thể làm với SDL timer là giới hạn tốc độ khung hình theo cách thủ công. Ở đây, chúng ta sẽ vô hiệu hóa vsync và duy trì tốc độ khung hình tối đa.

|  |
| --- |
| //Screen dimension constants  const int SCREEN\_WIDTH = 640;  const int SCREEN\_HEIGHT = 480;  **const int SCREEN\_FPS = 60;**  **const int SCREEN\_TICK\_PER\_FRAME = 1000 / SCREEN\_FPS;** |

* Trong bản demo này, chúng ta sẽ render khung hình bình thường, nhưng ở cuối mỗi khung hình chúng ta sẽ chờ thời gian của khung hình kết thúc. Ví dụ, khi chúng ta muốn render ở 60 fps, chúng ta cần 16 + 2/3 ms mỗi khung hình. Đó là lý do tại sao chúng ta tính số ticks mỗi khung hình theo ms.

|  |
| --- |
| //Create renderer for window              gRenderer = SDL\_CreateRenderer( gWindow, -1, SDL\_RENDERER\_ACCELERATED ); |

* Như bạn có thể thấy, chúng ta sẽ vô hiệu hóa VSync cho bản demo này vì chúng ta sẽ giới hạn tốc độ khung hình.

|  |
| --- |
| //Main loop flag              bool quit = false;              //Event handler              SDL\_Event e;              //Set text color as black              SDL\_Color textColor = { 0, 0, 0, 255 };              //The frames per second timer              LTimer fpsTimer;              //The frames per second cap timer              LTimer capTimer;              //In memory text stream              std::stringstream timeText;              //Start counting frames per second              int countedFrames = 0;              fpsTimer.start(); |

* Trong chương trình này, chúng ta sẽ không chỉ cần 1 timer để tính fps, mà còn cần 1 timer để giới hạn fps. Ở đây, trước khi ta vào main loop, chúng ta sẽ khai báo vài biến và bắt đầu timer tính fps.

|  |
| --- |
| //While application is running              while( !quit )              {                  //Start cap timer                  capTimer.start(); |

* Để giới hạn FPS, ta cần biết render một khung hình mất bao lâu. Ta chỉ cần start một timer ở đầu mỗi khung hình.

|  |
| --- |
| //Handle events on queue                  while( SDL\_PollEvent( &e ) != 0 )                  {                      //User requests quit                      if( e.type == SDL\_QUIT )                      {                          quit = true;                      }                  }                  //Calculate and correct fps                  float avgFPS = countedFrames / ( fpsTimer.getTicks() / 1000.f );                  if( avgFPS > 2000000 )                  {                      avgFPS = 0;                  }                  //Set text to be rendered                  timeText.str( "" );                  timeText << "Average Frames Per Second (With Cap) " << avgFPS;                  //Render text                  if( !gFPSTextTexture.loadFromRenderedText( timeText.str().c\_str(), textColor ) )                  {                      printf( "Unable to render FPS texture!\n" );                  }                  //Clear screen                  SDL\_SetRenderDrawColor( gRenderer, 0xFF, 0xFF, 0xFF, 0xFF );                  SDL\_RenderClear( gRenderer );                  //Render textures                  gFPSTextTexture.render( ( SCREEN\_WIDTH - gFPSTextTexture.getWidth() ) / 2, ( SCREEN\_HEIGHT - gFPSTextTexture.getHeight() ) / 2 );                  //Update screen                  SDL\_RenderPresent( gRenderer );                  ++countedFrames; |

* Ở đây chúng ta có đoạn code render khung hình và tính FPS như trước.

|  |
| --- |
| //If frame finished early                  int frameTicks = capTimer.getTicks();                  if( frameTicks < SCREEN\_TICK\_PER\_FRAME )                  {                      //Wait remaining time                      SDL\_Delay( SCREEN\_TICK\_PER\_FRAME - frameTicks );                  } |

* Cuối cùng, chúng ta có đoạn code để giới hạn tốc độ khung hình. Đầu tiên, chúng ta cần lấy số ticks mà khung hình cần để hoàn thành. Nếu số ticks khung hình cần quá ít, chúng ta sẽ delay thêm một chút thời gian để ngăn chương trình chạy quá nhanh.
* Đây là lý do tại sao chúng ta dùng VSync cho những bài trước thay vì giới hạn tốc độ khung hình thủ công. Khi chạy chương tình này, bạn sẽ nhận ra rằng nó chạy khá nhanh. Vì chúng ta sử dụng số nguyên (vì số dấu phẩy động khồn chính xác), số ticks mỗi khung hình sẽ là 16ms thay vì 16+2/3 ms. Cách này sẽ giải quyết trong trường hợp phần cứng không hỗ trợ VSync.