Một trong những chức năng mới của SDL 2 là khả năng xử lý nhiều windows cùng lúc. Trong bài viết này chúng ta sẽ sử dụng cùng lúc 2 window thay đổi được kích thước.

|  |
| --- |
| class Lwindow {      public:          //Intializes internals          LWindow();          //Creates window          bool init();          //Handles window events          void handleEvent( SDL\_Event& e );          //Focuses on window          void focus();          //Shows windows contents          void render();          //Deallocates internals          void free();          //Window dimensions          int getWidth();          int getHeight();          //Window focii          bool hasMouseFocus();          bool hasKeyboardFocus();          bool isMinimized();          bool isShown();      private:          //Window data          SDL\_Window\* mWindow;          SDL\_Renderer\* mRenderer;          int mWindowID;          //Window dimensions          int mWidth;          int mHeight;          //Window focus          bool mMouseFocus;          bool mKeyboardFocus;          bool mFullScreen;          bool mMinimized;          bool mShown;  }; |

Đây là window wrapper từ trước, có một vài điều chỉnh. Chúng ta cần lấy cái focus và báo là window có hiện hay không, nên chúng ta thêm hàm vào đó.

Mỗi window sẽ có render riêng để thêm biến thành viên lên đó. Chúng ta cũng theo dõi window ID để xem sự kiện gắn với window nào và chúng ta cũng có thể thêm một flag để theo dõi xem window có hiện không.

|  |
| --- |
| //Our custom windows  LWindow gWindows[ TOTAL\_WINDOWS ]; |

Ở bài này, ta có 3 window.

|  |
| --- |
| bool LWindow::init()  {      //Create window      mWindow = SDL\_CreateWindow( "SDL Tutorial", SDL\_WINDOWPOS\_UNDEFINED, SDL\_WINDOWPOS\_UNDEFINED, SCREEN\_WIDTH, SCREEN\_HEIGHT, SDL\_WINDOW\_SHOWN | SDL\_WINDOW\_RESIZABLE );      if( mWindow != NULL )      {          mMouseFocus = true;          mKeyboardFocus = true;          mWidth = SCREEN\_WIDTH;          mHeight = SCREEN\_HEIGHT;          //Create renderer for window          mRenderer = SDL\_CreateRenderer( mWindow, -1, SDL\_RENDERER\_ACCELERATED | SDL\_RENDERER\_PRESENTVSYNC );          if( mRenderer == NULL )          {              printf( "Renderer could not be created! SDL Error: %s\n", SDL\_GetError() );              SDL\_DestroyWindow( mWindow );              mWindow = NULL;          }          else          {              //Initialize renderer color              SDL\_SetRenderDrawColor( mRenderer, 0xFF, 0xFF, 0xFF, 0xFF );              //Grab window identifier              mWindowID = SDL\_GetWindowID( mWindow );              //Flag as opened              mShown = true;          }      }      else      {          printf( "Window could not be created! SDL Error: %s\n", SDL\_GetError() );      }      return mWindow != NULL && mRenderer != NULL;  } |

Đây là code tạo window và renderer. NÓ khá giống trước, chỉ là giờ nó ở trong wrapper class của chúng ta. Ta phải đảm bảo lấy window ID sau khi tạo cửa sổ vì ta cần ID để xử lý sự kiện.

|  |
| --- |
| void LWindow::handleEvent( SDL\_Event& e )  {      //If an event was detected for this window      if( e.type == SDL\_WINDOWEVENT && e.window.windowID == mWindowID )      {          //Caption update flag          bool updateCaption = false; |

Tất cả sự kiện từ tất cả window đều đi vào cùng event queue, vì thế để biết sự kiện của window nào thì chỉ cần check window ID của sự kiện đó.

|  |
| --- |
| switch( e.window.event )          {              //Window appeared              case SDL\_WINDOWEVENT\_SHOWN:              mShown = true;              break;              //Window disappeared              case SDL\_WINDOWEVENT\_HIDDEN:              mShown = false;              break;              //Get new dimensions and repaint              case SDL\_WINDOWEVENT\_SIZE\_CHANGED:              mWidth = e.window.data1;              mHeight = e.window.data2;              SDL\_RenderPresent( mRenderer );              break;              //Repaint on expose              case SDL\_WINDOWEVENT\_EXPOSED:              SDL\_RenderPresent( mRenderer );              break;              //Mouse enter              case SDL\_WINDOWEVENT\_ENTER:              mMouseFocus = true;              updateCaption = true;              break;                //Mouse exit              case SDL\_WINDOWEVENT\_LEAVE:              mMouseFocus = false;              updateCaption = true;              break;              //Keyboard focus gained              case SDL\_WINDOWEVENT\_FOCUS\_GAINED:              mKeyboardFocus = true;              updateCaption = true;              break;                //Keyboard focus lost              case SDL\_WINDOWEVENT\_FOCUS\_LOST:              mKeyboardFocus = false;              updateCaption = true;              break;              //Window minimized              case SDL\_WINDOWEVENT\_MINIMIZED:              mMinimized = true;              break;              //Window maximized              case SDL\_WINDOWEVENT\_MAXIMIZED:              mMinimized = false;              break;                //Window restored              case SDL\_WINDOWEVENT\_RESTORED:              mMinimized = false;              break;              //Hide on close              case SDL\_WINDOWEVENT\_CLOSE:              SDL\_HideWindow( mWindow );              break;          } |

Khi bạn có nhiều window, X ra khỏi window không cần thiết nghĩa là bạn đang kết thúc chương trình. Thay vào đó là ẩn từng window khi X ra. Vì vậy chúng ta sẽ theo dõi khi window bị ẩn/hiện bằng cách kiểm tra SDL\_WINDOWEVENT\_SHOWN / SDL\_WINDOWEVENT\_HIDDEN.