Render texture được tăng tốc phần cứng của SDL2 cũng giúp chúng ta lật và xoay hình ảnh nhanh. Trong bài viết này, chúng ta sẽ sử dụng điều này để làm một texture mũi tên quay và lật.

|  |
| --- |
| //Texture wrapper class  class LTexture  {      public:          //Initializes variables          LTexture();          //Deallocates memory          ~LTexture();          //Loads image at specified path          bool loadFromFile( std::string path );          //Deallocates texture          void free();          //Set color modulation          void setColor( Uint8 red, Uint8 green, Uint8 blue );          //Set blending          void setBlendMode( SDL\_BlendMode blending );          //Set alpha modulation          void setAlpha( Uint8 alpha );            //Renders texture at given point          void render( int x, int y, SDL\_Rect\* clip = NULL, double angle = 0.0, SDL\_Point\* center = NULL, SDL\_RendererFlip flip = SDL\_FLIP\_NONE );          //Gets image dimensions          int getWidth();          int getHeight();      private:          //The actual hardware texture          SDL\_Texture\* mTexture;          //Image dimensions          int mWidth;          int mHeight;  }; |

* Ở đây, chúng ta thêm một chức năng cho texture class. Hàm render bây giờ có góc xoay, điểm để xoay texture xung quanh và [hằng số lật](http://wiki.libsdl.org/SDL_RendererFlip).
* Giống như lật HCN, chúng tôi để tham số có giá trị mặc định trong trường hợp bạn muốn render texture mà không xoay hay lật.

|  |
| --- |
| void LTexture::render( int x, int y, SDL\_Rect\* clip, double angle, SDL\_Point\* center, SDL\_RendererFlip flip )  {      //Set rendering space and render to screen      SDL\_Rect renderQuad = { x, y, mWidth, mHeight };      //Set clip rendering dimensions      if( clip != NULL )      {          renderQuad.w = clip->w;          renderQuad.h = clip->h;      }      //Render to screen      SDL\_RenderCopyEx( gRenderer, mTexture, clip, &renderQuad, angle, center, flip );  } |

* Như bạn có thể thấy, tất cả những gì chúng ta đang làm là truyền tham số từ hàm của chúng ta vào SDL\_RenderCopyEx(). Hàm này hoạt động tương tự với hàm gốc SDL\_RenderCopy(), nhưng thêm tham số cho xoay và lật.

|  |
| --- |
| //Main loop flag  bool quit = false;  //Event handler  SDL\_Event e;  //Angle of rotation  double degrees = 0;  //Flip type  SDL\_RendererFlip flipType = SDL\_FLIP\_NONE;  //While application is running  while( !quit )  { |

* Trước khi chúng ta vào main loop, chúng ta khai báo biến để theo dõi góc xoay và cách lật hình.

|  |
| --- |
| //Handle events on queue  while( SDL\_PollEvent( &e ) != 0 )  {      //User requests quit      if( e.type == SDL\_QUIT )      {          quit = true;      }      else if( e.type == SDL\_KEYDOWN )      {          switch( e.key.keysym.sym )          {              case SDLK\_a:              degrees -= 60;              break;                case SDLK\_d:              degrees += 60;              break;              case SDLK\_q:              flipType = SDL\_FLIP\_HORIZONTAL;              break;              case SDLK\_w:              flipType = SDL\_FLIP\_NONE;              break;              case SDLK\_e:              flipType = SDL\_FLIP\_VERTICAL;              break;          }      }  } |

* Trong event loop, chúng ta có thể tăng/giảm góc xoay với a/d hay đổi kiểu flip với q, w, e.

|  |
| --- |
| //Clear screen  SDL\_SetRenderDrawColor( gRenderer, 0xFF, 0xFF, 0xFF, 0xFF );  SDL\_RenderClear( gRenderer );  //Render arrow  gArrowTexture.render( ( SCREEN\_WIDTH - gArrowTexture.getWidth() ) / 2, ( SCREEN\_HEIGHT - gArrowTexture.getHeight() ) / 2, NULL, degrees, NULL, flipType );  //Update screen  SDL\_RenderPresent( gRenderer ); |

* Đến đây chúng ta mới thật sự render. Đầu tiên, chúng ta truyền tọa độ x, y. Nhìn code có vẻ giống một phương trình phức tạp nhưng chỉ để căn giữa cái ảnh mà thôi.
* Tham số tiếp theo là HCN cắt và vì chúng ta render toàn bộ texture nên chúng là NULL. Tham số tiếp theo là góc xoay (theo độ). Tham số tiếp theo là điểm chúng ta xoay quanh. Khi điểm là NULL thì nó sẽ xoay quanh tâm hình ảnh. Tham số cuối cùng là cách xoay ảnh.
* Cách tốt nhất để nhớ cách quay ảnh là chơi với nó. Trải nghiệm để xem các hiệu ứng khi bạn kết hợp các kiểu quay và lật.