Nhờ có render tăng tốc phần cứng mới, độ trong suốt đã nhanh hơn rất nhiều trong SDL 2.0. Ở đây chúng tôi sử dụng điều chế alpha để kiểm soát độ trong suốt của texture.

|  |
| --- |
| // 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);      // 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 sẽ thêm 2 hàm để hỗ trợ tăng độ trong suốt alpha của texture. Đầu tiên, setAlpha() có chức năng khá giống setColor(). setBlendMode() sẽ kiểm soát cách pha trộn texture. Để quá trình pha trộn hoạt động bình thường, bạn phải thiết lập chế độ pha trộn của texture. Chúng tôi sẽ nói chi tiết sau.

|  |
| --- |
| bool loadMedia()  {      // Loading success flag      bool success = true;      // Load front alpha texture      if (!gModulatedTexture.loadFromFile("fadeout.png"))      {          printf("Failed to load front texture!\n");          success = false;      }      else      {          // Set standard alpha blending          gModulatedTexture.setBlendMode(SDL\_BLENDMODE\_BLEND);      }      // Load background texture      if (!gBackgroundTexture.loadFromFile("fadein.png"))      {          printf("Failed to load background texture!\n");          success = false;      }      return success;  } |

* Đây là hàm load texture, chúng tôi load texture đằng trước và sẽ pha trộn alpha và nền texture. Vì texture phía trước trong suốt hơn, nên có thể thấy rõ hơn texture đằng sau. Như bạn thấy trong code, sau khi chúng tôi load texture trước thành công, chúng tôi thiết lập SDL\_BlendMode sang blend để pha trộn dduocjjw bật. Vì nền sẽ không trong suốt nên ta không phải thiết lập pha trộn cho chúng.
* Còn alpha hoạt động như thế nào? Alpha là độ mờ và độ mờ càng thấp thì ta càng dễ nhìn xuyên qua nó. Như R,G,B, nó cũng chạy từ 0 đến 255 khi pha trộn nó.

|  |
| --- |
| void LTexture::setBlendMode(SDL\_BlendMode blending)  {      // Set blending function      SDL\_SetTextureBlendMode(mTexture, blending);  }  void LTexture::setAlpha(Uint8 alpha)  {      // Modulate texture alpha      SDL\_SetTextureAlphaMod(mTexture, alpha);  } |

* Đây là những hàm SDL thực tế sẽ hoạt động. SDL\_SetTextureBlendMode() cho phép ta bật chế độ pha trộn và SDL\_SetTextureAlphaMod() cho phép chúng ta thiết lập lượng alpha cho cả texture.

|  |
| --- |
| // Main loop flag  bool quit = false;  // Event handler  SDL\_Event e;  // Modulation component  Uint8 a = 255;  // While application is running  while (!quit){ |

* Trước khi vào trong main loop, chúng ta khai báo 1 biến để kiểm soát lượng alpha của texture.Vì được khởi tạo là 255 nên ban đầu texture đằng trước hoàn toàn rõ ràng.

|  |
| --- |
| // Handle events on queue  while (SDL\_PollEvent(&e) != 0)  {      // User requests quit      if (e.type == SDL\_QUIT)      {          quit = true;      }      // Handle key presses      else if (e.type == SDL\_KEYDOWN)      {          // Increase alpha on w          if (e.key.keysym.sym == SDLK\_w)          {              // Cap if over 255              if (a + 5 > 255)              {                  a = 255;              }              // Increment otherwise              else              {                  a += 5;              }          }          // Decrease alpha on s          else if (e.key.keysym.sym == SDLK\_s)          {              // Cap if below 0              if (a - 5 < 0)              {                  a = 0;              }              // Decrement otherwise              else              {                  a -= 5;              }          }      }  } |

* Event loop sẽ làm alpha tăng/giảm dựa vào w/s.

|  |
| --- |
| // Clear screen  SDL\_SetRenderDrawColor(gRenderer, 0xFF, 0xFF, 0xFF, 0xFF);  SDL\_RenderClear(gRenderer);  // Render background  gBackgroundTexture.render(0, 0);  // Render front blended  gModulatedTexture.setAlpha(a);  gModulatedTexture.render(0, 0);  // Update screen  SDL\_RenderPresent(gRenderer); |

* ở cuối main loop, chúng ta sẽ render. Sau khi clear màn hình, ta render nền trước và sau đó chúng ta render phần texture trước được pha trộn lên trên nó. Trước khi render texture trước, chúng ta thiết lập giá trị alpha của nó.