X ra khỏi window chỉ là một trong những events SDL có thể xử lý. Một kiểu khác là input bàn phím. Trong bài này, ta sẽ làm những hình ảnh khác nhau xuất hiện phụ thuộc vào phím mà ta nhấn.

|  |
| --- |
| // Key press surfaces constants  enum KeyPressSurfaces  {      KEY\_PRESS\_SURFACE\_DEFAULT, // = 0      KEY\_PRESS\_SURFACE\_UP, // = 1      KEY\_PRESS\_SURFACE\_DOWN, // = 2 …      KEY\_PRESS\_SURFACE\_LEFT,      KEY\_PRESS\_SURFACE\_RIGHT,      KEY\_PRESS\_SURFACE\_TOTAL  }; |

* Mở đầu, ta liệt kê tất cả surface ta có. Ta sẽ để mặc định từ 0,1,2… Ta có thể gán chúng giá trị cụ thể, nhưng chúng ta không đề cập ở bài này.
* **Một thói quen xấu của beginner là đặt 1 số tùy ý thay vì 1 hằng số tượng trưng. Chương trình nhỏ thì điều này ổn, nhưng với những code ngàn dòng thì rất đau đầu. VD: ta nên dùng if (opt == MAIN\_MENU) thay vì if (opt == 1).**

|  |
| --- |
| // Starts up SDL and creates window  bool init();  // Loads media  bool loadMedia();  // Frees media and shuts down SDL  void close();  // Loads individual image  **SDL\_Surface \*loadSurface(std::string path);**  // The window we'll be rendering to  SDL\_Window \*gWindow = NULL;  // The surface contained by the window  SDL\_Surface \*gScreenSurface = NULL;  // The images that correspond to a keypress  **SDL\_Surface \*gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_TOTAL];**  // Current displayed image  SDL\_Surface \*gCurrentSurface = NULL; |

* Bên cạnh nhưng hàm đã nói, ta viết 1 hàm mới: loadSurface . Thay vì phải copy/paste code cho nhiều ảnh, ta sẽ viết 1 hàm xử lý hết việc đó.
* Chúng ta cũng có thêm các pointers gKeyPressSurfaces chứa tất cả các ảnh.

|  |
| --- |
| **SDL\_Surface \*loadSurface(std::string path)**  **{**  **// Load image at specified path**  **SDL\_Surface \*loadedSurface = SDL\_LoadBMP(path.c\_str());**  **if (loadedSurface == NULL)**  **{**  **printf("Unable to load image %s! SDL Error: %s\n", path.c\_str(), SDL\_GetError());**  **}**  **return loadedSurface;**  **}** |

* Hàm loadSurface sẽ load ảnh và báo lỗi. Để vào cùng 1 hàm sau này sẽ dễ debug load ảnh.

|  |
| --- |
| bool loadMedia()  {      // Loading success flag      bool success = true;      // Load default surface      gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DEFAULT] = loadSurface("04\_key\_presses/press.bmp");      if (gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DEFAULT] == NULL)      {          printf("Failed to load default image!\n");          success = false;      }      // Load up surface      gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_UP] = loadSurface("04\_key\_presses/up.bmp");      if (gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_UP] == NULL)      {          printf("Failed to load up image!\n");          success = false;      }      // Load down surface      gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DOWN] = loadSurface("04\_key\_presses/down.bmp");      if (gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DOWN] == NULL)      {          printf("Failed to load down image!\n");          success = false;      }      // Load left surface      gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_LEFT] = loadSurface("04\_key\_presses/left.bmp");      if (gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_LEFT] == NULL)      {          printf("Failed to load left image!\n");          success = false;      }      // Load right surface      gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_RIGHT] = loadSurface("04\_key\_presses/right.bmp");      if (gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_RIGHT] == NULL)      {          printf("Failed to load right image!\n");          success = false;      }      return success;  } |

* Hàm loadMedia sẽ load tất cả các ảnh định render

|  |
| --- |
| // Main loop flag  bool quit = false;  // Event handler  SDL\_Event e;  **// Set default current surface**  **gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DEFAULT];**  // While application is running  while (!quit)  { |

* Trong hàm main, trước khi vào main loop, ta set surface mặc định để display.

|  |
| --- |
| // Handle events on queue  while (SDL\_PollEvent(&e) != 0)  {      // User requests quit      if (e.type == SDL\_QUIT)      {          quit = true;      }      // User presses a key      else if (e.type == SDL\_KEYDOWN)      {          // Select surfaces based on key press          switch (e.key.keysym.sym)          {          case SDLK\_UP:              gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_UP];              break;          case SDLK\_DOWN:              gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DOWN];              break;          case SDLK\_LEFT:              gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_LEFT];              break;          case SDLK\_RIGHT:              gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_RIGHT];              break;          default:              gCurrentSurface = gKeyPressSurfaces[KEY\_PRESS\_SURFACE\_DEFAULT];              break;          }      }  } |

* Đây là event loop. Ta đã xử lý đóng window như trong bài trước, sau đó xử lý SDL\_KEYDOWN.
* Trong SDL\_Event là SDL\_Keyboard\_event chứa thông tin cho key event. Trong đó là SDL\_Keysym chưa thông tin về phím được nhấn. Keysym đó chứ SDL\_Keycode, cái mà sẽ xác định phím nào đã được nhấn.
* Như đã thấy thì code trên set cái surface dựa trên phím được nhấn. Tìm trong tài liệu SDL để biết hêm keycoddes cho các phím khác. [SDL Event](http://wiki.libsdl.org/SDL_Event?highlight=%28\bCategoryStruct\b%29|%28CategoryEvents%29)  [SDL Keyboard event](http://wiki.libsdl.org/SDL_KeyboardEvent) [SDL Keysym](http://wiki.libsdl.org/SDL_Keysym) [SDL Keycode](http://wiki.libsdl.org/SDL_Keycode)

|  |
| --- |
| // Apply the current image  SDL\_BlitSurface(gCurrentSurface, NULL, gScreenSurface, NULL);  // Update the surface  SDL\_UpdateWindowSurface(gWindow); |

* Cuối cùng là ta sẽ blit surface lên màn hình.