Bên cạnh việc đưa ảnh lên màn hình, chúng ta còn phải xử lý nhưng dữ liệu vào của người dùng. Bạn có thể làm điều này với SDL: Hệ thống xử lý sự kiện.

|  |
| --- |
| // Main loop flag  bool quit = false;  // Event handler  SDL\_Event e; |

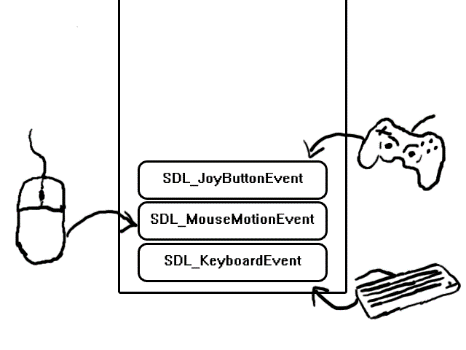
* quit: flag để xem người dùng đã tắt chưa. Vì vừa chạy chương trình nên dĩ nhiên khởi tạo nó là false
* Khối SDL\_Event là 1 thứ như kiểu nhấn phím, di chuột hay bấm nút trên bộ điều khiển game… Ở đây, ta đang quan tâm quit events để đóng ứng dụng.

|  |
| --- |
| // While application is running  while (!quit) |

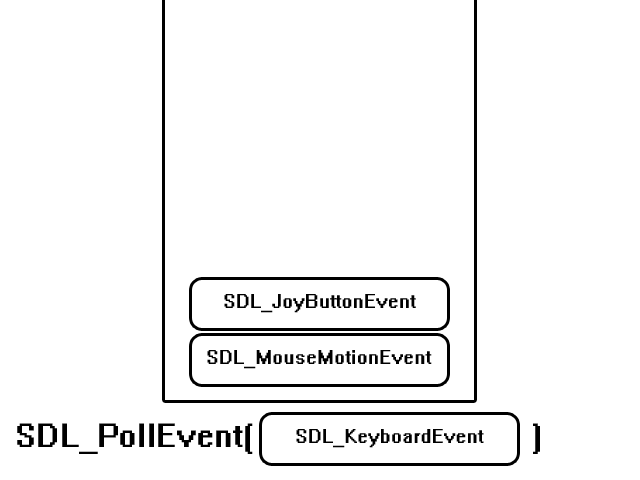
* Trong bài trước, ta có 1 chương trình tự tắt sau vài giây. Bài này, chúng ta sẽ tạo 1 chương trình mà người dùng sẽ phải tự kết thúc nó. Ta chỉ cần lặp cho tới khi người dùng tắt nó.
* Vòng lặp này là vòng lặp chính, hay gọi là main loop, game loop. Đây là phần chính của bất kỳ game nào.

|  |
| --- |
| // Handle events on queue  while (SDL\_PollEvent(&e) != 0)  {      // User requests quit      if (e.type == SDL\_QUIT)      {          quit = true;      }  } |

* Đầu tiên, chúng ta sẽ có 1 event loop. Nó sẽ liên tục xử lý hàng đợi event cho đến khi hết event.
* Chỉ cần nhấn chuột, di chuột hay chạm vào màn hình cảm ứng, ta đã thêm 1 event vào hàng đợi.



* Hàng đợi event sẽ store chúng chờ đến khi bạn xử lý chúng. Khi bạn muốn tìm xem event nào đang diễn ra thì bạn lấy event gần nhất từ event queue bằng SDL\_PollEvent().
* SDL\_PollEvent sẽ lấy event gần nhất trong queue rồi chuyển thông tin sang SDL\_Event ta truyền cho nó.



* SDL\_PollEvent sẽ lấy cho đến khi hết event. Khi hết, nó trả về 0.
* Nếu 1 sự kiện bằng SDL\_QUIT, ta set quit là true rồi kết thúc chương trình.

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

* Xong thì ta sẽ vẽ lên màn hình