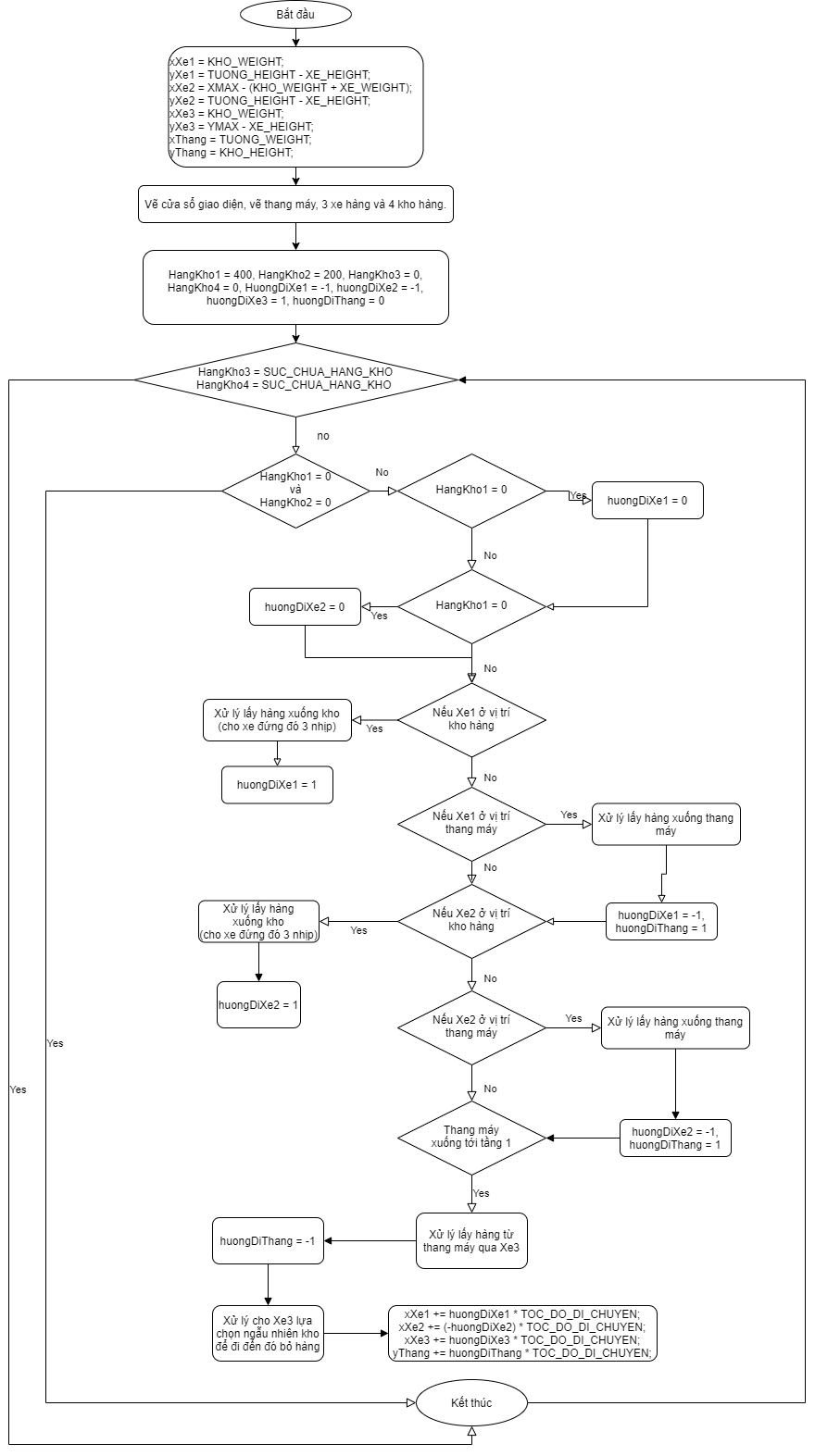
**Báo cáo thuật toán xe hàng**

1. **Lưu đồ thuật toán:**

****

1. **Mô tả hằng số và biến số**
   1. **Hằng số:**
      * KHO\_WEIGHT, KHO\_HEIGHT : Độ rộng và cao của kho hàng khi vẽ trên màn hình
      * TUONG\_HEIGH, TUONG\_WEIGHT: Độ cao và rộng của khung tường ngăn cách tầng 1 và tầng 2.
      * YMAX, XMAX: Độ cao và rộng của toàn bộ khung hình để vẽ các vật thể
      * XE\_HEIGHT, XE\_WEIGHT: Độ cao và rộng của chiếc xe
      * SUC\_CHUA\_HANG\_KHO: Số lượng hàng tối đa mà kho có thể chứa được.
      * TOC\_DO\_THANG\_DI\_QUA: Tốc độ di chuyển của các vật thể trên màn hình
   2. **Biến số:**
      * HangKho1: Số lượng hàng khi bắt đầu game của kho 1 (Tầng trên bên trái)
      * HangKho2: Số lượng hàng khi bắt đầu game của kho 2 (Tầng trên bên phải)
      * HangKho3: Số lượng hàng khi bắt đầu game của kho 3 (Tầng dưới bên trái)
      * HangKho4: Số lượng hàng khi bắt đầu game của kho 4 (Tầng dưới bên phải)
      * huongDiXe1: Hướng đi của xe 1 (Tầng trên bên trái)
      * huongDiXe2: Hướng đi của xe 2 (Tầng trên bên phải)
      * huongDiXe3: Hướng đi của xe 3 (Tầng dưới)
      * huongDiThang: Hướng đi của thang máy
      * xXe1, yXe1: lần lược là hoành độ và tung độ của xe 1
      * xXe2, yXe2: lần lược là hoành độ và tung độ của xe 2
      * xXe3, yXe3: lần lược là hoành độ và tung độ của xe 3
      * xThang, yThang: lần lược là hoành độ và tung độ của xe thang
2. **Mô tả các hàm và thủ tục xử dụng:**

* **void VeKhungTuong():** Dùng để vẽ ra khung của bức tường phân chia tầng 1 và 2
* **void VeKhoHang(int left, int top, int status):** Dùng để vẽ một kho hàng ở vị trí tọa độ (left, top) trên màn hình và có trạng thái là status với status bằng 1 thì kho hàng vẽ màu trắng, bằng 0 thì kho hàng vẽ bằng màu đỏ.
* **void VeXe(int left, int top):** Dùng để vẽ xe ở vị trí có tọa độ là (left, top) trên màn hình
* **void VeThang(int left, int top, int status):** Dùng để vẽ thang máy ở vị trí có tọa độ là (left, top) với trạng thái **Status** bằng 1 thì vẽ thang đang chở hàng, bằng 0 thì vẽ thang trống
* **int kiemTraViTriXe1(int x1):** hàm trả về 1 nếu xe đã tới chỗ thang, 0 nếu đang đi giữa đường và -1 nếu xe tới kho.
* **int kiemTraViTriXe2(int x2):** hàm trả về 1 nếu xe đã tới chỗ thang, 0 nếu đang đi giữa đường và -1 nếu xe tới kho.
* **int kiemTraViTriThang(int y):** hàm trả về 1 nếu xe đã tới mặt đất tầng 1, 0 nếu đang đi giữa hai tầng và -1 nếu xe tới tầng 2.
* **int kiemTraViTriXe3(int x3, int xthang, int yThang):** trả về 2 nếu xe 3 chạm tới chỗ thang ở bên phải, 1 là vừa tới kho bên phải, -1 là tới kho bên trái, -2 là chạm tới thang ở bên trái.
* **void Xe1VaThang(int& huongDiXe1, int& hangXe1, int& xThang, int yThang, int& huongDiThang, int& sttThang, int& hangThang):** Xử lý chuyển động cho xe 1 bỏ hàng lên thang.
* **void Xe2VaThang(int& huongDiXe2, int& hangXe2, int& xThang, int yThang, int& huongDiThang, int& sttThang, int& hangThang):** Xử lý chuyển động cho xe 2 bỏ hàng lên thang.
* **void Xe3VaThang(int& xXe3, int& huongDiXe3, int& hangXe3, int& xThang, int yThang, int& huongDiThang, int& sttThang, int& hangThang):** Xử lý chuyển động cho xe 3 lấy hàng từ thang.
* **void VeGiaoDien():** Thực hiện quản lý và điều phối mọi hoạt động trên màn hình.
* **int main():** Hàm thực thi chương trình.

1. **Mô tả môi trường cài đặt chương trình:**

* Chương trình được cài đặt bằng ngôn ngữ C với sự hỗ trợ của IDE Visual studio 2019
* Đồ họa trong chương trình được dựng lên nhờ vào sự hỗ trợ của thư viện đồ họa ***Borland Graphics Interface (BGI)***

1. **Tích hợp chương trình và quá trình hoàn thành chương trình:**

* Ban đầu là quá trình tìm kiếm thư viện hỗ trợ đồ họa. Với tiêu chí nhẹ, dễ sử dụng và tương thích với Visual studio 2019 em đã tìm ra được thư viện **BGI** và tích hợp nó vào chương trình của mình.
* Để tích hợp được thư viện **BGI** vào visual studio 2019, ta tải project **BGI** dành cho visual studiomới nhất về và mở project lên, tạo 1 file Main.cpp trong project này và tiến hành cài đặt trong đó.
* Thư viện BGI chỉ hỗ trợ vẽ các loại hình học cơ bản như hình chữ nhật, hình tròn và đường thẳng. Nên em đã tạo ra những hàm mới vẽ xe, kho, thang máy và khung tường cho game
* Sau khi vẽ được giao diện tĩnh em tiến hành xử dụng kỹ thuật xóa vật thể cũ cần di chuyển và vẽ lại cái mới dịch chuyển đi 5 bước so với vị trí ban đầu, giúp tạo cảm giác chuyển động cho vật thể.
* Sau cùng là xử lý sự kiện đụng độ giữa các vật thể với nhau để tạo ra các hiệu ứng tương tác giữa các vật thể trên màn hình,

1. **Mô tả về thử nghiệm và đánh giá nó**

* Sau loạt thư nghiệm em thấy mọi thứ đã đi vào đúng quỹ đạo của nó, tuy nhiên 1 khuyết điểm của thư viện này là khi vật thể chuyển động sẽ để lại bóng mờ trên màn hình, cảm giác chuyển động của vật thể không mượt.
* Ngoài ra các xử lý khác đều hoàn hảo. Ví dụ như khi kho 1 hoặc kho 2 hết hàng nó sẽ chuyển sang màu đỏ và lấy hàng của nó sẽ không di chuyển nữa, còn thang máy cũng không đến lấy hàng ở bên hết hàng nữa. Nếu cả hai kho cùng hết hàng sẽ có thông báo kết thúc trò chơi hiện lên và trò chơi dừng lại.
* Xử lý ở kho 3 và kho 4 cũng được xử lý hoàn hảo khi kho nào đầy nó sẽ đổi sang màu đỏ và xe 3 sẽ không đem hàng về hướng đó nữa. Khi cả hai kho đầy thì chương trình cũng kết thúc và hiện chữ trò chơi kết thúc lên màn hình.

1. **Đóng gói chương trình và hướng dẫn sử dụng**

* Chương trình thực thi là file .exe nằm trong thư mục Debug.
* Để sử dụng chương trình bạn chỉ cần khởi chạy chương trình và quan sát kết quả trả ra của chương trình.