1. **Giới thiệu sơ lược về game.**
   1. **Giới thiệu về Nguyễn Hà Đông**

* Có hình ảnh về tác giả Nguyễn Hà Đông
* Sinh năm 1985, Hà Nội
* Kỹ sư Công nghệ thông tin chuyên ngành Hệ thống thông tin tại Đại học Bách Khoa Hà Nội
  1. **Giới thiệu về game Flappy Bird**
* Phần này sẽ có hình ảnh game trên di động, nếu có thêm biểu đồ so sánh với game khác thì càng tốt- mục đích nhằm nhấn mạnh
* Xuất bản vào tháng 5 – 2013
* Đạt 50 triệu lượt tải trên toàn thế giới
* Gỡ khỏi Appstore vào ngày 10 – 2 – 2014.
* Mang thiết kế tối giản, đồ họa pixel
* Theo tạp chí CNET, Flappy Bird đã lọt top 25 ứng dụng có tầm ảnh hưởng lớn nhất trong suốt 1 thập kỷ qua (từ 2010 đến 2019).
  1. **Ý tưởng sơ lược về game**
* Phần này sẽ làm 1 ảnh để mô tả chuyển động của các đối tượng để người xem dễ hình dung hơn
* 3 đối tượng chính là bird, pipes và ground ( con chim và cặp ống cống và mặt đất)
* Đối tượng Bird có chuyển động mặc định là rơi tự do từ trên xuống, nếu click sẽ bay lên 1 khoảng (Bird di chuyển theo trục Y, rơi là +y, bay lên là –y)
* Đối tượng Pipes là 1 cặp ống top và bottom ở giữa có khoảng trống space, có chuyển động từ phải sang trái theo trục X ( -x )
* Đối tượng Ground cũng sẽ di chuyển tương tự Pipes và cùng vận tốc
* Ngoài ra còn có các đối tượng phụ:
* Background – Bối cảnh
* Score – Điểm số
* Medal – Huy Chương

1. **Trình bày phần Code**
   1. **Trước khi nói phần code hãy nói phần drawImage và giới thiệu về spriters trước.**

* Giới thiệu Sprites: *Sprites là một hình ảnh lớn được tạo ra bằng cách gộp nhiều ảnh nhỏ lại với nhau theo một cách được định trước sao cho có thể tái sử dụng lại từng ảnh nhỏ mà không bị ảnh hưởng bởi các ảnh khác.*
* *Vì 1 game sẽ có rất nhiều đối tượng cần được giả dụ như đối tượng bird lát nữa chúng ta sẽ code có 3 animation, nếu mỗi animation là 1 ảnh khác nhau, kèm theo đó là các đối tượng khác sẽ có nhiều animation khác nhau nữa sẽ ảnh hưởng đến perfomance. Vậy nên việc tích hợp những ảnh nhỏ thành 1 Sprites để tối ưu hóa hiệu năng cho trang web.*
* Ok giờ đến phần cú pháp method drawImage();

**ctx.drawImage( image, sX, sY, sW, sH, cX, cY, cW, cH)**

trong đó:

* Image: là element từ bất cứ nguồn canvas image nào. Ở bài này tôi sẽ tạo biến image có source là cái ảnh Sprites
* sX và sY là tọa độ điểm bắt đầu của phần ảnh muốn lấy trên Sprites
* sW và sH là chiều rộng và chiều dài của phần ảnh muốn lấy trên Sprites
* cX và cY là tọa độ điểm bắt đầu của ảnh muốn vẽ trên Canvas
* cW và cH là chiều ngang và chiều dọc của phần ảnh muốn vẽ trên Canvas

Phần này nên bổ xung thêm ảnh để giải thích dễ hiểu hơn

* 1. **Tạo canvas và setup**
* Do height của background là 625 và height của ground là 143 ta sẽ tạo height của canvas nhỏ hơn tổng height của bg và ground ở bài này tôi đặt là 710
* Phần Width của canvas tôi đặt cho tỉ lệ vừa phải với Height canvas, ở đây tôi đặt là 530.
* Đến phần CSS canvas chúng ta dùng position để căn canvas ra giữa ( nên tạo border cho canvas để có thể thấy canvas đã nằm ra giữa hay chưa)
* Phần tiếp theo chúng ta sẽ tạo 3 function đại diện cho 3 hành động chính
* Function update – chứa tất cả các update của đối tượng như là chuyển động lên xuống của bird, chuyển động sang ngang của pipes và ground…
* Function draw – chứa tất cả các function draw của các đối tượng.
* Function animate – có method *requestAnimationFrame()*quá quen thuộc rồi, gọi lại các function. Function này gọi function update và draw ở trên.
* Phần tiếp theo là tạo 1 biến sprites là 1 element Image mới có src là đường dẫn đến file sprites

1. **Background và ground**

* Background sẽ không chuyển động được vẽ từ vị trí ( 0, 0) và có width và height trên canvas bằng trên sprites, như các bạn thấy thì bg đang không phủ hết phần trên của canvas, chúng ta cần vẽ thêm 1 bg nữa nối tiếp với bg vừa nãy vị trí (0, 229) ( *Ở phần code sX là 1 mảng chứa 2 giá trị, ở đây chúng ta có thể lấy được 2 background khác nhau – mục đích của việc này là tạo ra 1 button để thay đổi background của game*)
* Ở đây tôi tạo 1 object bg chứa các tất cả các giá trị để draw trên canvas, đồng thời obj này cũng chứa 1 hàm draw( ) ( bg.draw( ) ) chúng ta sẽ gọi trong Func draw() tổng vừa tạo ở bên dưới đoạn code
* Đến phần ground, ground là đối tượng chuyển động (ở screen start chưa chuyển động, đến screen play là chuyển động theo chiều -x)
* Ở phần này mục đích là sẽ tạo 1 mảng **arrGround[]** chứa nhiều các phần tử ground, khi **arrGround[0]** di chuyển vượt ngoài phạm vi hiển thị của canvas chúng ta có hàm **updateArrGround()** xóa bỏ **arrGround[0]-** đã vượt ra ngoài phạm vi hiển thị và thêm 1 phần tử vào cuối của mảng

1. **Các trạng thái game, đối tượng hiển thị trong các trạng thái đó và lắng nghe sự kiện click chuột**

* Phần này ta sẽ tạo 1 obj GAME chứa trạng thái của game, giả dụ ở trạng thái start sẽ hiển thị những gì, event click sẽ làm hành động gì đó tùy thuộc vào obj GAME đang ở trạng thái gì, ở đây game sẽ có 3 trạng thái
  + Start: trạng thái ban đầu người chơi chưa điều khiển bird,trạng thái này sẽ có thông tin về game và hướng dẫn người chơi click chuột để bắt đầu chơi game, event click ở phần này sẽ chuyển trạng thái game sang ***play***
  + Play: bird, ground và pipes sẽ thêm các update ( bird sẽ rơi và lắng nghe click để bay lên, ground và pipes sẽ chuyển động) event click ở phần này sẽ truyền 1 giá trị -v ngược với v là vận tốc khi bird rơi, để bird có hành động bay lên trên. Ở phần này chúng ta cũng sẽ có các function để kiểm tra các hành động va chạm giữa bird và pipes, bird và ground. Khi sảy ra va chạm trạng thái game sẽ chuyển qua ***end***
  + End: sẽ draw khung điểm, số điểm, điểm cao nhất và huân chương, sẽ có func tính huân chương theo số điểm cao nhất. event click ở phần này sẽ chuyển trạng thái game trở về ***start***

\*phần này nên có thêm ảnh để mô tả cho người xem

* Game sẽ được chia thành 3 phần:
  + start – lúc bắt đầu sẽ hiển thị image tên Flappy Bird, Get Ready và Tap tap những image này ta sẽ dùng hàm drawImage vẽ ra tương tự như phần bg