



NGUYÊN LÝ THIẾT KẾ TRÒ CHƠI MÁY TÍNH SỬ DỤNG NGÔN NGỮ LẬP TRÌNH SCRATCH 3.0

Phạm Xuân nguyên¹

Ngày nhận bài: 24/4/2023

Ngày chấp nhận đăng: 19/5/2023

Tóm tắt: Ngày nay, rất nhiều các em học sinh đang muốn học thêm kiến thức về tin học, đặc biệt cách thức tạo các game nhỏ sử dụng các ngôn ngữ lập trình trực quan và dễ học. Tự xây dựng các Game không chỉ giúp các em rèn luyện tư duy, sự sáng tạo mà còn là hoạt động bổ ích giúp các em tránh được các hoạt động không lành mạnh khi sử dụng máy tính nhiều như nghiện Game, nghiện xem Video trên YouTube... Thời gian gần đây các video, bài viết hướng dẫn xây dựng Game sử dụng ngôn ngữ lập trình Scratch [1] xuất hiện nhiều. Tuy nhiên, rất ít bài viết hướng dẫn tổng quát và có tính hệ thống các bước làm. Ngoài ra, cũng chưa có nhiều bài viết chia sẻ kinh nghiệm cũng như các lỗi thường gặp trong quá trình hoàn thiện Game. Từ thực tế đó, tôi viết một bài hướng dẫn xây dựng trò chơi máy tính với tiêu đề “Nguyên lý Thiết kế trò chơi máy tính sử dụng ngôn ngữ lập trình scratch 3.0”.

Từ khóa: Scratch, thiết kế Game.

PRINCIPLES OF DESIGNING COMPUTER GAMES USING SCRATCH 3.0 PROGRAMMING LANGUAGE

Abstract: Nowadays, a lot of students want to learn more about computer science, especially how to create mini games using intuitive and easy-to-learn programming languages. Building games by themselves not only helps children practice thinking skills and creativity skills, but also a useful activity to help them avoid unhealthy activities when using computers such as game addiction, addiction to watching videos on the Internet (for example YouTube). Recently, there are a lot of videos, article showing how to build a Game using Scratch [1] programming language. However, very few articles provide comprehensive and systematic instructions on how to do it. In addition, there are not many articles to share experiences as well as common errors in the process of completing the Game. From that fact, I wrote a tutorial on building computer games titled “Principles of Designing Computer Games Using Scratch 3.0 Programming Language”.

Keywords: Scratch, Design Computer Games

1. Giới thiệu

Scratch là ngôn ngữ lập trình được nghiên cứu và phát triển bởi nhóm nghiên cứu Lifelong Kindergarten, thuộc trung tâm Media Lab của Viện công nghệ Massachusetts (Massachusetts Institute of Technology – MIT, thành lập năm 1981 tại Thành phố Cambridge, Bang Massachusetts, Hoa Kỳ). Dẫn đầu nhóm nghiên cứu dự án sáng tạo ra phần mềm lập trình Scratch là giáo sư Mitchel Resnick, Giám đốc điều hành Lifelong Kindergarten. Scratch có thể được bản cài đặt trên máy tính cá nhân hoặc lập trình online[1].

Phiên bản ngôn ngữ lập trình Scratch 3.0 là phiên bản lập trình Scratch nâng cao mới được nhóm nghiên cứu phát hành vào ngày 2 tháng 1 năm 2019 với nhiều nâng cấp mới. Đáng chú ý, phần mềm này được công khai và hoàn toàn miễn phí trên máy tính, người dùng có thể tải về laptop, sử dụng mà không mất chi phí hay bị bản quyền.

¹ Trung tâm Ngoại ngữ - Tin học, Trường Đại học Hoa Lu; Email: pxnguyen@hluv.edu.vn



Scratch 3.0 là phần mềm được thiết kế với giao diện đẹp mắt, gần gũi với trẻ em nên rất dễ dàng sử dụng, bởi vừa trực quan, vừa đơn giản phù hợp cho mọi lứa tuổi. Dưới đây là 10 ưu điểm vượt trội có trên ngôn ngữ lập trình scratch 3.0.

- Hỗ trợ đắc lực cho trẻ rèn luyện tính kiên trì, tỉ mỉ trong học tập và trong cuộc sống.
- Hình thành thói quen tự giác học tập, giải quyết công việc từ những nội dung thực hành phần mềm.
- Giúp trẻ phát huy trí tưởng tượng mới mẻ, sáng tạo trong quá trình vừa học vừa chơi phần mềm.
- Bồi dưỡng đam mê học tập trong trẻ từ sớm, nhờ đồ họa, nội dung hấp dẫn với độ tuổi.
- Nội dung lập trình Scratch 3.0 giúp trẻ học cách diễn đạt ý tưởng, suy nghĩ logic, chặt chẽ. Đặc biệt là sự phát triển của tư duy lập trình ở trẻ.
- Scratch cũng yêu cầu khả năng kết nối với những người tham gia khác. Từ đó, trẻ biết cách làm việc nhóm hiệu quả.
- Trẻ học được tư duy chọn lọc, đưa ra quyết định, thử trải nghiệm điều mới mẻ trong phần mềm.
- Những hoạt động trong phần mềm sẽ dần hoàn thiện và nâng cao kỹ năng giao tiếp của trẻ.
- Chia sẻ và học hỏi kiến thức về Scratch bằng hình thức trực tuyến thông qua các nhóm cộng đồng, bởi hiện nay trên thế giới, lượng người sử dụng ngôn ngữ lập trình Scratch là rất đông.
- Scratch là ngôn ngữ nền tảng giúp bạn trang bị những kiến thức hàn lâm về lập trình.

Một trong những ứng dụng rất nổi bật của phần mềm Scratch là thiết kế ra các trò Game trí tuệ, hoạt ảnh, mô phỏng... Với sự hỗ trợ đa phương tiện cùng các khối lệnh cơ bản, trẻ em sẽ dễ dàng sáng tạo ra các Game theo sở thích và ý tưởng của bản thân. Việc này sẽ giúp các trẻ có cơ hội khám phá và nâng cao trí tưởng tượng của bản thân, đồng thời hình thành và phát triển khả năng tư duy Logic, trí tưởng tượng một cách hiệu quả. Bởi trẻ được học các kỹ năng lập trình từ dễ đến khó, tạo ra các trò Game từ cơ bản đến phức tạp.

Scratch chỉ thực sự bùng nổ từ năm 2014 khi một số quốc gia như Anh, Mỹ đã đổi mới đột phá chương trình giảng dạy môn Tin học trong nhà trường, đưa nội dung kiến thức Khoa học máy tính vào nhà trường ngay từ cấp Tiểu học.

Ở Việt nam, sau năm 2018, Bộ Giáo dục & Đào tạo cũng đã quyết định đưa nội dung kiến thức Khoa học máy tính trong môn Tin học vào ngay từ cấp Tiểu học. Ngôn ngữ lập trình Scratch đã được sử dụng để tiếp cận mô hình giáo dục STEM [2], [3].

Nguồn tài liệu hướng dẫn sử dụng về ngôn ngữ lập trình Scratch rất phong phú [4], [5]. Trong các tài liệu này đều có nhiều phần hướng dẫn xây dựng các Game. Ngoài ra, có rất nhiều bài viết, video trên mạng hướng dẫn xây dựng Game bằng cả tiếng Anh và tiếng Việt.

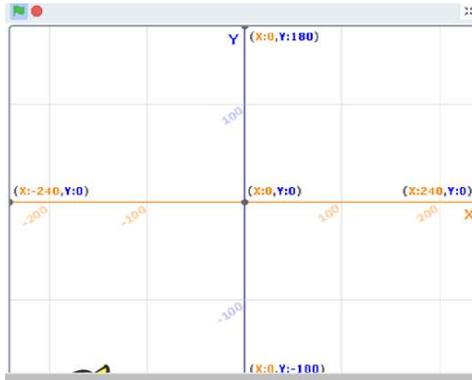
Các bài hướng dẫn xây dựng Game đều có hình ảnh hoặc video minh họa chi tiết các bước để người dùng có thể làm theo. Tuy nhiên, nếu không nắm vững nguyên lý cơ bản để xây dựng các Game, người dùng sẽ khó làm được các Game có chất lượng cao, tạo sự hấp dẫn cho người chơi. Ngoài ra, trong quá trình xây dựng Game, người dùng có thể gặp nhiều lỗi phát sinh mà phần kiến thức cơ bản không nói đến. Trong bài viết này, tôi sẽ tổng hợp lại các nguyên lý cơ bản để xây dựng Game cũng như chỉ ra những kinh nghiệm trong quá trình hoàn thiện Game. Bài viết này phù hợp cho cả giáo viên và học sinh, những người đã biết kiến thức cơ bản về Scratch.

2. Nội dung (sử dụng scratch để thiết kế trò chơi)

2.1. Thiết kế sân khấu (Stage)

Người dùng chọn sân khấu phù hợp với nội dung Game. Ví dụ, Game dưới nước thì chọn sân khấu nhóm UnderWater. Để nhanh chóng chọn được sân khấu phù hợp thì dùng từ khóa tiếng Anh trong ô tìm kiếm. Nếu không có sân khấu phù hợp thì có thể tải ảnh từ mạng internet làm thành sân khấu hoặc tự vẽ sân khấu. Lưu ý, để có thể xem được kích thước sân khấu và các điểm mốc của sân khấu, có thể chọn loại sân khấu có tên Xy-grid. Việc nắm được kích thước sân khấu, và tọa độ các điểm mốc là cực kỳ cần thiết để tạo được chuyển động cho các nhân trong Game.





Hình 2.1.1. Tọa độ sân khấu

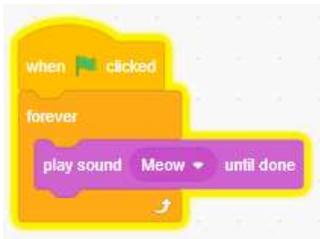
Sau khi đã có sân khấu, cần làm thêm các loại phong nền (Backdrops) khác nhau tùy theo nội dung Game. Có ba loại phong nền cơ bản là: Phong nền lúc bắt đầu trò chơi, phong nền lúc người chơi chiến thắng và phong nền lúc người chơi thua cuộc. Phong nền lúc bắt đầu trò chơi thường xuất hiện sau khi người chơi ấn nút “Flag” hoặc nút “Start”. Phong nền lúc người chơi chiến thắng thường xuất hiện khi người chơi tích lũy được số điểm nào đó. Phong nền này thường có chứa chữ “WIN” hoặc ảnh mang ý nghĩa chiến thắng như cúp vàng, pháo hoa, pháo giấy... Phong nền lúc người chơi thất bại thường chứa chữ (hoặc ảnh) “GAME OVER”, “YOU LOSE”... Phong nền này thường xuất hiện khi người chơi hết giờ, bị trừ số điểm vượt quá ngưỡng cho phép... Ngoài ra, cũng cần tạo thêm các phong nền khác tùy vào các tình huống trong Game.

Với Game yêu cầu nhân vật di chuyển nhiều, có lộ trình thì có thể tự xây dựng sân khấu dạng đường vòng, có các mốc để quan sát nhân vật di chuyển được nhiều hơn.

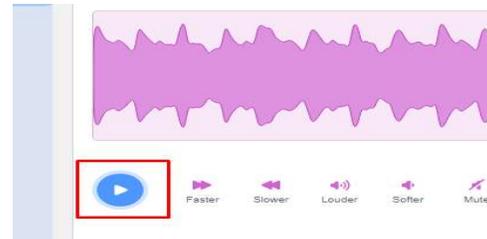


Hình 2.1.2. Sân khấu dạng đường vòng có các chặng

Chọn âm thanh nền cho Game. Âm thanh cho Game nên tạo sự hồi hộp, gay cấn, kích thích người chơi. Ví dụ, nhập từ khóa “Video Game” sẽ chọn âm thanh nền cho Game có sẵn trong hệ thống. Nếu không chọn âm thanh này, người sử dụng có thể tải âm thanh từ trên mạng. Âm thanh tải về ngoài yêu cầu bắt buộc là phù hợp với nội dung Game thì cũng cần loại bỏ tạp âm, không nên để quá to hoặc thời gian quá dài, quá ngắn so với thời gian trong Game. Sau khi tải xong cần dùng các công cụ có sẵn để cắt bỏ, chỉnh sửa cho phù hợp. Âm thanh chỉ có thời gian giới hạn, để âm thanh này có thể lặp lại liên tục trong quá trình tạo Game, cần có câu lệnh “Play sound..until” done và nằm trong vòng lặp forever.



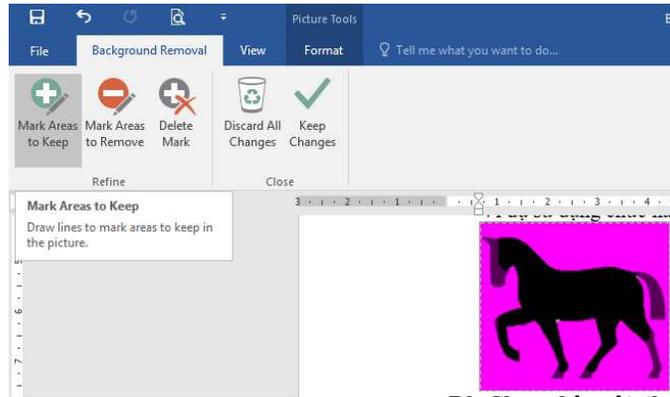
Hình 2.1.3. Cách tạo âm thanh nền chạy liên tục



Hình 2.1.4. Cách chỉnh sửa âm thanh

2.2. Thiết kế nhân vật

Để chọn được nhân vật theo đúng yêu cầu Game, ngoài các nhân vật có sẵn thì người dùng cũng có thể download các ảnh từ trên mạng Internet. Để tìm được đúng ảnh thì nên thêm chữ “icon” khi tìm kiếm trên google. Các ảnh sử dụng cũng nên có dạng hoạt hình, phù hợp với Game thiết kế trong Scratch. Khi ảnh được lấy từ trên mạng, nếu vẫn còn nền ảnh xung quanh nhân vật (*Brackground*) thì cần loại bỏ nền ảnh (ví dụ sử dụng chức năng “*Remove Background*” trong phần mềm MS Word). Nếu không loại bỏ các nền này thì khi đưa nhân vật vào Game, nền của ảnh sẽ che mất một phần hình ảnh của sân khấu, gây mất thẩm mỹ.



Hình 2.2.1. Loại bỏ Back Ground xung quanh ảnh nhân vật

Với các ảnh có sẵn trong hệ thống nên ưu tiên chọn nhân vật có nhiều trang phục (*Costumes*) khác nhau bởi vì khi lập trình chúng ta có thể thông qua việc thay đổi trang phục của nhân vật để tạo cảm giác là nhân vật đang hoạt động.

Sau khi đã chọn được nhân vật cần chỉnh sửa các thông số liên quan tới nhân vật tùy theo sân khấu, nhân vật khác và nội dung trò chơi. Cụ thể, với kích thước nhân vật nên để 50% hoặc nhỏ hơn tùy bối cảnh. Hướng (*direction*) ban đầu là 90. Nếu Game có nút Start thì ban đầu cho nhân vật ẩn và chỉ hiện sau khi nút *Start* được ấn và một tin nhắn “Start” được phát đi. Lưu ý các thông số này nên được khai báo đầy đủ và chi tiết khi bắt đầu trò chơi vì nếu không Game có thể sử dụng các thông số sau cùng của nhân vật. Ví dụ sau câu lệnh di chuyển, nhân vật sẽ ở mép màn hình của sân khấu, khi người chơi ấn vào nút *Flag* để chơi lại Game thì nhân vật vẫn ở mép sân khấu trong khi vị trí của nhân vật nên là giữa sân khấu.

Mỗi nhân vật có nhiều trang phục (*Costumes*) khác nhau vì thế nên đặt tên các trang phục tương ứng với hành động, hoặc trạng thái của nhân vật.

Chọn lựa âm thanh cho nhân vật: Mỗi cử động của nhân vật, mỗi tình huống (chết, trúng đạn, chạm đồ vật, vượt qua chướng ngại vật, có điểm...) đều nên có âm thanh tương ứng. Có thể download thêm các âm thanh này từ trên mạng[6], nhưng cần chỉnh sửa âm lượng, độ dài cho phù hợp. Cần lưu ý là để Game có thể cùng lúc phát ra nhiều âm thanh khác nhau thì câu lệnh phát âm thanh nên là lệnh “*Start sound*”.

Chọn đúng cách thức di chuyển cho nhân vật

Để tạo sự di chuyển cho nhân vật cần áp dụng một trong các cách sau:

Lệnh tạo di chuyển	Tình huống và nhân vật hay sử dụng	Ví dụ
Go to	Di chuyển nhân vật lúc bắt đầu Game, kết thúc Game Di chuyển nhân vật ngẫu nhiên Nhân vật hay sử dụng là nút bấm, quả bóng...	Go to random position Go to x: 0 y:0
Glide	Di chuyển nhân vật nhanh (lướt) đến vị trí có tọa độ sẵn. Lệnh này áp dụng cho các nhân vật có thời gian di chuyển, không cần quan tâm tới hướng (<i>direction</i>) của nhân vật	Glide 1 sec to mouse point
Move	Di chuyển nhân vật theo hướng hiện thời. Lệnh này thường áp dụng cho nhân vật cần di chuyển chậm, liên tục và có hướng di chuyển	Move 10 Waite 0.01

Tạo hiệu ứng nhân vật di chuyển liên tục bằng lệnh Move. Ví dụ, con cá đang bơi, quả rơi,... Nguyên lý cơ bản để tạo hiệu ứng nhân vật di chuyển là:

- Thiết lập tọa độ ban đầu.
- Tạo vòng lặp liên tục.
- Thay đổi tọa độ nhân vật sau mỗi khoảng thời gian.
- Khi tọa độ giảm đến ngưỡng nào đó thì quay lại tọa độ ban đầu.

Để di chuyển nhân vật theo chiều ngang thì thay đổi tọa độ X, để di chuyển nhân vật theo chiều dọc thì thay đổi tọa độ Y. Có thể xem lại phần tọa độ sân khấu để biết được các vị trí mép sân khấu. Để làm cho nhân vật di chuyển nhanh hơn thì giá trị *change* cần tăng lên. Cuối mỗi phần di chuyển luôn cần xử lý trường hợp nhân vật chạm mép sân khấu.

Tạo hiệu ứng nhân vật cử động. Ví dụ như chim vỗ cánh, cá quẫy đuôi. Nguyên lý là sử dụng một vòng lặp liên tục để thay đổi trang phục nhân vật và câu lệnh wait. Để tạo cảm giác nhân vật di chuyển nhanh thì điều chỉnh thời gian wait giảm xuống. Lưu ý: Có thể thêm hiệu ứng thay đổi *direction* để nhân vật cử động linh hoạt, hoặc thêm nhiều trang phục hơn để đối tượng cử động mượt hơn.



Hình 2.2.2. Sử dụng vòng lặp liên tục để tạo cảm giác nhân vật cử động.

2.3. Thiết kế các hoạt động chính trong Game

Tạo nút Start

Thay vì dùng nút *Flag* để bắt đầu trò chơi, Game nên sử dụng nút *Start* để bắt đầu trò chơi. Cách làm này vừa giúp người chơi dễ thao tác bắt đầu Game, vừa tạo ấn tượng chuyên nghiệp cho Game. Nút *Start* này ban đầu hiện ở chính giữa màn hình còn các nhân vật khác trong Game ẩn đi. Sau khi ấn nút thì một thông điệp, ví dụ “start” được gửi tới các nhân vật khác trong Game đồng thời nút *Start* này cũng ẩn đi.

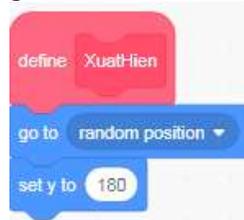
Để giúp người chơi nhanh chóng quay lại Game thì nút *Start* này có thể xuất hiện lần nữa ở phòng nền khi người chơi chiến thắng hoặc thua cuộc. Cũng có thể tạo thêm một nhân vật khác có chức năng quay lại Game và sử dụng chữ “PLAY AGAIN”.

Sử dụng bàn phím để điều khiển nhân vật: Ví dụ như khi ấn mũi tên bên phải (*right arrow*) và mũi tên trái (*left arrow*), mũi tên lên (*up arrow*), mũi tên xuống (*down arrow*) để di chuyển tương ứng nhân vật sang phải, sang trái, lên trên và xuống dưới. Với hoạt động thường xuyên của nhân vật (ví dụ đạn được bắn đi) thì nên chọn sử dụng phím cách (*space*). Ưu điểm của cách sử dụng bàn phím để điều khiển nhân vật là dễ lập trình, có thể tạo các Game có nhiều người chơi bằng cách sử dụng các phím khác nhau.

Sử dụng chuột để điều khiển nhân vật. Ví dụ như trong game khi ăn chuối[10], khi chuột nhấn vào vị trí nào thì vị trí khi di chuyển tới vị trí đó. Chúng ta sẽ so sánh vị trí hiện thời của nhân vật và vị trí chuột để có hướng di chuyển tương ứng. Cách sử dụng chuột này sẽ hữu ích khi người dùng muốn phát triển ứng dụng lên phiên bản điện thoại di động. Lúc này mỗi thao tác di chuyển chuột sẽ biến thành thao tác chạm vào màn hình.

Tạo sự xuất hiện với vị trí ngẫu nhiên cho nhân vật. Ví dụ với Game khi ăn quả chuối, nên tạo sự xuất hiện của chuối là ngẫu nhiên trước khi rơi xuống. Nguyên lý thiết kế như sau:

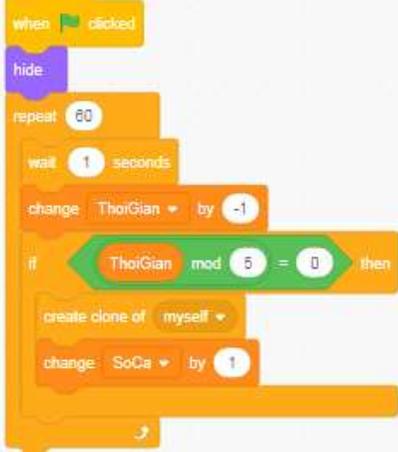
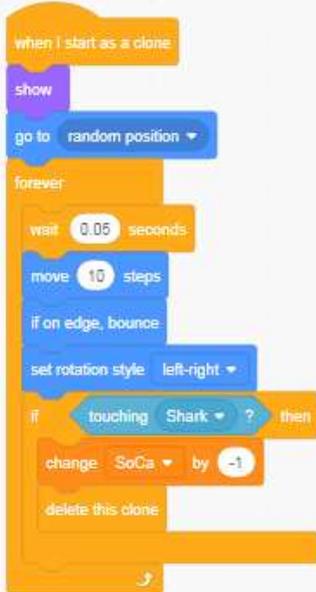
- Di chuyển nhân vật đến vị trí ngẫu nhiên.
- Chính lại y (hoặc x) để đặt đối tượng bắt đầu ở trên cao (hoặc thấp) hoặc bên lề trái (phải)



Hình 2.3.1. Tạo vị trí xuất hiện ngẫu nhiên cho nhân vật.

Tạo bản sao cho nhân vật. Ví dụ trò chơi bắn rồng [8] thì các nhân vật con rồng liên tục xuất hiện sau một khoảng thời gian nhất định. Các bản sao sẽ có một nhân vật gốc tạo ra, các bản sao sẽ có code, trang phục giống nhau.

Khi tạo bản sao cần tăng biến đếm số lượng nhân vật. Nhân vật gốc sẽ cần đứng yên một chỗ, chỉ phục vụ việc “đề” ra các bản sao. Sau đây là ví dụ các bản sao của nhân vật cá sẽ xuất hiện liên tục sau mỗi 5 giây:

	<p>Sau khi click vào lá cờ</p> <p>Ẩn nhân vật</p> <p>Biến thời gian giảm dần sau mỗi giây</p> <p>Cứ sau mốc 5 lần giảm thì tạo một bản sao và tăng biến đếm số lượng cá</p>
	<p>Khi bản sao bắt đầu hoạt động</p> <p>Hiện nhân vật</p> <p>Cho nhân xuất hiện ngẫu nhiên</p> <p>Cứ sau mỗi 0.05 giây thì đi chuyển 10 bước</p> <p>Nếu chạm cạnh thì quay đầu và chuyển hướng</p> <p>Nếu chạm cá mập thì giảm biến đếm và xóa bản sao</p>

Code của bản gốc chỉ nên chứa thao tác liên quan tới tạo bản sao, các biến toàn cục (như thời gian, số lượng bản sao) và không nên chứa code liên quan tới hoạt động thực sự của nhân vật. Nhân vật gốc nên cho ẩn.

Code của bản sao liên quan tới hoạt động của nhân vật và không được chứa điều khiển tạo bản sao mới (“create clone of”). Trong code của bản sao cần có lệnh “delete this clone” để xóa bản sao.

Để có thể kiểm soát được việc tạo bản sao thì ban đầu cần thử nghiệm với bản gốc trước, sau khi hoàn thiện mới tạo bản sao. Kiểm tra từng trường hợp đơn giản, sau đó mới tăng số lượng bản sao, tăng code chứa hoạt động bản sao.

Tạo biến thời gian

Cách 1: Dùng đối tượng Timer (trong nhóm Sensing), và thêm điều khiển ở sân khấu. Cách này thời gian tăng dần. Khi bắt đầu trò chơi cần lệnh “Reset Timer” biến này về giá trị ban đầu.

Cách 2: Tạo biến Time và thay đổi theo thời gian thực. Câu lệnh được gán trong điều khiển sân khấu và hiện biến thời gian. Các trò chơi thường cho thời gian giảm dần về không.



Hình 2.3.2. Tạo biến thời gian giảm dần

Biến thời gian cũng có thể bị giảm nhiều hơn 1 trong trường hợp người chơi vi phạm quy tắc nào đó của trò chơi như trả lời sai câu hỏi, chạm vào nhân vật cấm...

Tạo biến chứa điểm. Biến điểm nên được thiết lập ban đầu là 0. Biến đếm phải có cả sự kiện tăng lên và giảm đi khi người chơi vi phạm quy tắc trò chơi. Mức tăng của biến điểm cũng cần khác nhau tùy độ khó của các tình huống khác nhau.

Truyền và nhận thông điệp. Khi Game có một trạng thái mới (khởi đầu Game, kết thúc Game...) cần có một thông điệp được gửi đi (broadcast message). Các nhân vật trong Game sau khi nhận được thông điệp sẽ có phản ứng tương ứng với thông điệp được nhận được. Để đảm bảo thông điệp phát đi chỉ một lần duy nhất cần sử dụng kết hợp việc phát đi thông điệp và câu lệnh “stop this script”



Hình 2.3.3. Dừng phát thông báo start nhiều lần bằng câu lệnh stop this script

Cách thức tính điểm trò chơi. Để thực hiện việc tính điểm cho Game thường sử dụng các cách sau:

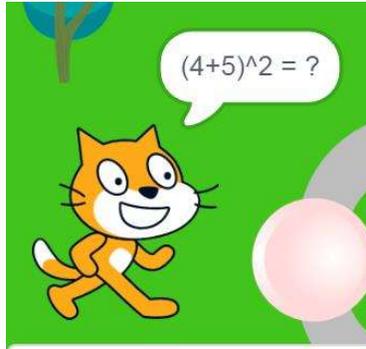
• Nhân vật chạm của nhân vật khác trong Game. Ví dụ trong Game khi ăn chuối thì khi con khi chạm vào quả chuối sẽ được một điểm. Sự kiện này nên được đặt riêng trong một khối lệnh *forever*.

	<p>Khi click vào lá cờ</p> <ul style="list-style-type: none"> • Một quá trình liên tục • Nếu chạm vào con khi • Phát ra âm thanh ăn chuối • Thay đổi điểm • Làm cho quả chuối nhỏ dần • Quả chuối xuất hiện ở vị trí ban đầu tạo cảm giác di chuyển
--	---

• Nhân vật chạm con trỏ chuột (touching mouse-pointer) để tính điểm. Ví dụ trò chơi bắt bóng. Lưu ý: phần code liên quan tới thao tác chạm để tính điểm nên để riêng một khối, không

nên để chung với các câu lệnh khác như di chuyển. Trong nhiều trường hợp, thao tác di chuyển (như move, glide...) diễn ra quá nhanh thì không kiểm tra được sự kiện hai nhân vật chạm nhau.

- Thông qua việc nhập nội trả lời câu hỏi.



Hình 2.3.4. Game trả lời câu hỏi

- Thông qua việc người chơi vượt qua chướng ngại vật. Ví dụ trong trò chơi Flappy bird [7], mỗi lần chướng ngại vật biến mất là một lần được cộng điểm.
- Ngoài ra, có thể dùng thư viện mở rộng Video Sensing của Scratch để sử dụng webcam như một thiết bị cảm biến phát hiện vị trí có sự chuyển động và màu sắc, từ đó tương tác với nhân vật trong Game.

Cách thức hiện điểm của trò chơi. Để có thể hiện điểm của trò chơi ấn tượng hơn, ta có thể làm các bước như sau:

- Thêm một nhân vật là con số 0
- Thêm 9 trang phục cho nhân vật này tương ứng là các số từ 1 tới 9
- Tùy theo điểm số của Game mà hiện con số tương ứng và tìm vị trí thích hợp
- Nếu điểm lớn hơn 10 thì có thể tạo thêm bản sao để hiển thị



Hình 2.3.5. Cách sử dụng các số để hiện điểm

Code	Giải thích
<pre> when I receive game over show forever if point < 10 then switch costume to point + 1 if point > 9 and point < 100 then go to x: 40 y: -45 switch costume to letter 1 of point create clone of myself go to x: 60 y: -45 switch costume to letter 2 of point </pre>	<p>Khi kết thúc Game thì hiện con số</p> <p>Nếu điểm số < 10 thì hiện trang phục tương ứng. Ví dụ nếu điểm số là 5 thì hiện trang phục là số 6.</p> <p>Nếu điểm lớn hơn 10 thì tạo hai bản sao. Bản thứ nhất thể hiện điểm tương ứng với kí tự đầu tiên của biến điểm.</p> <p>Bản thứ hai thể hiện điểm tương ứng với kí tự thứ hai của biến điểm.</p> <p>Hai bản sao được di chuyển tới hai vị trí cạnh nhau</p>

TÀI LIỆU THAM KHẢO

- [1] Giới thiệu về Scratch, <https://scratch.mit.edu/about>, ngày truy cập 21/04/2023.
- [2] ThS Nguyễn Bùi Hậu, ThS Phạm Thị Thu Hiền, 2019. “*Tiếp cận mô hình giáo dục STEM thông qua phần mềm SCRATCH*”. Tạp chí Thiết bị Giáo dục. Quyển: 202, Trang: 12-15.
- [3] Bùi Anh Tuấn, Bùi Lê Diễm, Trương Quốc Tuấn và Lâm Minh Huy, 2019. “*Tiếp cận mô hình giáo dục STEM thông qua phần mềm Scratch cho sinh viên sư phạm toán tại Trường Đại học Cần Thơ*”. Tạp chí Khoa học Đại học Cần Thơ. Tập. 55, Số. 3 (2019), trang 56-64.
- [4] Nguyễn Hữu Hưng, Tạ Thị Bích, Dương Lục, Nguyễn Thị Minh, 2021. “*Sách Lập trình với Scratch 3 (Dành cho học sinh 8-14 tuổi)*”.
- [5] Bùi Việt Hà (2019), *Tự học lập trình Scratch*, ebook.
- [6] Website chứa âm thanh, <https://elements.envato.com/sound-effects/>, ngày truy cập 21/04/2023.
- [7] Trò chơi Flappy Bird, <https://scratch.mit.edu/projects/768347448>, ngày truy cập 21/4/2023.
- [8] Trò chơi bắn rồng, <https://scratch.mit.edu/projects/838835778>, ngày truy cập 21/04/2023.
- [9] Trò chơi dưới nước, <https://scratch.mit.edu/projects/768814124>, ngày truy cập 21/04/2023.
- [10] Trò chơi khi ăn chuối, <https://scratch.mit.edu/projects/783175783/>, ngày truy cập 21/04/2023.
- [11] Công nghệ Video Sensing của scratch, <https://scratch.mit.edu/search/studios?q=video%20sensing>, ngày truy cập 21/4/2023.

