

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



KỸ NĂNG CHUYÊN NGHIỆP CHO KỸ SƯ (CO2001)

Bài tập lớn 2 - L01 - Học kì 241

"Thiết Kế Trò Chơi Platformer Cổ Điển"

Instructor(s): GV. Nguyễn Hoàng Thiên Khôi

Student(s):	Trần Phước Nhật	2212412 (Lớp L01 - Nhóm 14)
	Trần Thị Ngọc Huyền	2211311 (Lớp L01 - Nhóm 14)
	Trương Nguyễn Minh Nhiên	2212452 (Lớp L01 - Nhóm 14)

Mục lục

1	Đặt vấn đề	4
2	Mục tiêu của đề tài	5
2.1	Mục tiêu dự án	5
2.2	Mục tiêu học tập của nhóm	5
3	Công nghệ sử dụng	6
3.1	Quản lý toàn bộ dự án	6
3.1.1	Giới thiệu về Trello	6
3.1.2	Quản lý công việc với Trello	7
3.1.2.a	Giao diện chung	7
3.1.2.b	Giám sát chi tiết tiến độ công việc	8
3.1.2.c	Timeline làm việc nhóm	9
3.2	Quản lý mã nguồn	10
3.2.1	Giới thiệu về Github	10
3.2.2	Quản lý mã nguồn bằng Github	10
3.2.3	Quản lý phiên bản bằng Github	12
3.3	Mô phỏng giao diện trò chơi	13
3.3.1	Giới thiệu về Figma	13
3.3.2	Một số hình ảnh trong figma	13
3.4	Thiết kế nhân vật	13
3.4.1	Giới thiệu về Aseprite	13
3.4.2	Thiết kế animation với Aseprite	14
3.5	Phát triển trò chơi	15
3.5.1	Giới thiệu về Unity	15
3.5.2	Phát triển trò chơi bằng Unity	16
4	Mô phỏng trò chơi	19
4.1	Cấu trúc để mô phỏng game bằng Figma.	19
4.2	Giao diện Menu	21
4.3	Giao diện Resume (Tạm dừng) và GameOver (Thua)	22
4.4	Giao diện các thử thách trong trò chơi	22
4.5	Giao diện Kết truyện của Game	25
5	Hiện thực trò chơi	26
5.1	Sơ lược về cốt truyện trò chơi	26
5.2	Chi tiết cốt truyện các màn chơi	26
5.3	Demo sản phẩm	27
5.3.1	Thao tác di chuyển, tấn công	27
5.3.2	Chương ngại vật	28



5.3.3	Kết thúc trò chơi	32
6	Tài nguyên dự án	34
7	Tài liệu tham khảo	34



1 Đặt vấn đề

Hiện nay, các vấn đề về rối loạn tâm thần như lo âu, trầm cảm, stress ở sinh viên được ghi nhận là khá nghiêm trọng và có xu hướng tăng cao do áp lực học tập, tìm kiếm việc làm, gia đình,... Một trong những nguyên nhân phải kể đến là mất cân bằng giữa học tập, làm việc và thư giãn nghỉ ngơi, khiến trạng thái não bộ căng thẳng trong thời gian dài dẫn đến rối loạn. Các hình thức thư giãn phù hợp không chỉ giúp giảm căng thẳng mà nhờ đó còn phát triển khả năng tư duy sáng tạo, khả năng giải quyết vấn đề.

Thư giãn bằng cách chơi các trò chơi điện tử là một trong những biện pháp hữu hiệu nếu biết cân bằng thời gian, không lạm dụng. Nhiều nghiên cứu chỉ ra rằng, việc chơi các game hành động rèn luyện cho người chơi khả năng phản ứng linh hoạt, phản xạ nhanh và giải quyết vấn đề tốt hơn. Chơi game cũng giúp rèn luyện thị lực, tăng khả năng quan sát nhất là trong hoàn cảnh các đồ vật được sắp xếp lộn xộn mà người chơi game biết tiết chế không nhìn quá lâu vào màn hình điện tử. Tăng khả năng tập trung và kiên nhẫn. Bên cạnh đó, những trò chơi thú vị còn tạo chủ đề nói chuyện, trao đổi giao lưu giữa cộng đồng sinh viên, giúp mở rộng mạng lưới giao tiếp.

Hiểu được điều đó, nhóm chúng em thực hiện chủ đề **Thiết kế trò chơi Platformer cổ điển - Loopy's Adventure** với mong muốn xây dựng trò chơi thú vị, dễ thao tác đáp ứng nhu cầu giải trí của người chơi. Loopy's Adventure được xây dựng với đồ họa thân thiện, nhân vật trò chơi gần gũi người và có thao tác đơn giản sẽ là công cụ tuyệt vời đáp ứng nhu cầu thư giãn đối với những người cần thả lỏng. Đồng thời, với xuất phát điểm là sinh viên ngành Khoa học Máy tính, nhóm cũng mong muốn được học hỏi thêm về các công nghệ mới, quy trình phát triển trò chơi điện tử.



2 Mục tiêu của đề tài

Xuyên suốt quá trình thực hiện dự án, nhóm cũng tự đặt ra một số mục tiêu về sản phẩm đầu ra và về kết quả tìm hiểu, nghiên cứu mà các thành viên cần đạt được. Những mục tiêu này vừa là kim chỉ nan hoạt động vừa là tiêu chí đánh giá mức độ hoàn thành dự án, cụ thể:

2.1 Mục tiêu dự án

Với mong muốn ban đầu là tạo ra sản phẩm game có khả năng đáp ứng nhu cầu giải trí cho người chơi, giảm căng thẳng giúp thư giãn sau những giờ học tập mệt mỏi, trò chơi điện tử Loopy's Adventure cần đáp ứng một số yêu cầu sau:

1. Đồ họa: Thân thiện, phù hợp nhiều lứa tuổi. Phù hợp với cốt truyện, thể loại trò chơi. Hiệu ứng hình ảnh và âm thanh đáp ứng nhu cầu cơ bản của trò chơi.
2. Hiệu suất: Trò chơi được xử lý mượt mà, không có lỗi và ít giật lag trong quá trình chơi.
3. Cốt truyện - cơ chế chơi: Cốt truyện thú vị, lôi cuốn có các thử thách cũng như thưởng phạt hợp lý. Thao tác - cơ chế trò chơi dễ hiểu, dễ thực hiện.
4. Giá trị chơi lại: Trò chơi có những thử thách khó, hay tăng khả năng chơi lại, thời lượng game không được quá dài gây mất thời gian, tác động xấu đến sức khỏe người chơi.

2.2 Mục tiêu học tập của nhóm

1. Về kỹ năng:

- Kỹ năng quản lý dự án: Có khả năng lên kế hoạch thực hiện - chia nhỏ - phân công dự án, giám sát và thực hiện theo tiến độ. Làm quen với công cụ quản lý dự án Trello.
- Kỹ năng làm việc nhóm - trình bày ý tưởng: Làm việc nhóm chuyên nghiệp, hiệu quả, bám sát tiến độ, kế hoạch đề ra. Quá trình làm việc nhóm cần trao đổi thông tin liên tục giữa các thành viên, cần thực hiện nhiều biện pháp trao đổi như qua văn bản, qua các cuộc họp, thuyết trình trước lớp. Nhờ đó rèn luyện khả năng trình bày ý tưởng.
- Kỹ năng lập trình, quản lý dự án code: Rèn luyện khả năng lập trình, tổ chức code với các công cụ quản lý như Github, Plastic SCM.

2. Về kiến thức:

- Quy trình xây dựng - phát triển một trò chơi điện tử: Nhận biết, hiểu các bước - bộ phận cần có trong quá trình xây dựng trò chơi đồng thời có khả năng tự triển khai các dự án game nhỏ.
- Học thêm ngôn ngữ mới và cách làm việc với engine Unity: Dự án được thực hiện bằng ngôn ngữ C là cơ hội học hỏi một trong những ngôn ngữ thông dụng nhất hiện nay, đồng thời biết cách thao tác với Unity - engine làm game phổ biến nhất thị trường làm game Việt Nam tính đến nay.



3 Công nghệ sử dụng

3.1 Quản lý toàn bộ dự án

3.1.1 Giới thiệu về Trello

Trello là ứng dụng quản lý công tác làm việc nhóm theo phong cách Kanban (phương pháp Thẻ thị giác) được xây dựng trên môi trường Web Application bởi Atlassian. Với Trello, ta có thể chia nhỏ dự án thành các mục nhỏ và theo dõi tiến độ các mục này qua các trạng thái: To-do, Doing, Review, Done. Nhờ vậy, đối với các dự án nhỏ, chỉ cần nhìn vào cách triển khai trên Trello ta có thể dễ dàng xác định phần việc và tiến độ từng phần.



Hình 1: Logo phần mềm quản lý dự án Trello

Thành phần cơ bản của Trello kể đến như:

1. *Organization*: Tổng hợp các board và nhân sự trong team.
2. *Board*: Tương ứng 1 dự án hoặc 1 mảng công việc, trong board có nhiều list.
3. *List*: List là một tập hợp các card có cùng tính chất như Doing, Done, Review,...
4. *Card*: Một thẻ công việc có tiêu đề, có mô tả, có nhiệm vụ cụ thể, người phụ trách, các ghi chú cũng như tài nguyên liên quan.

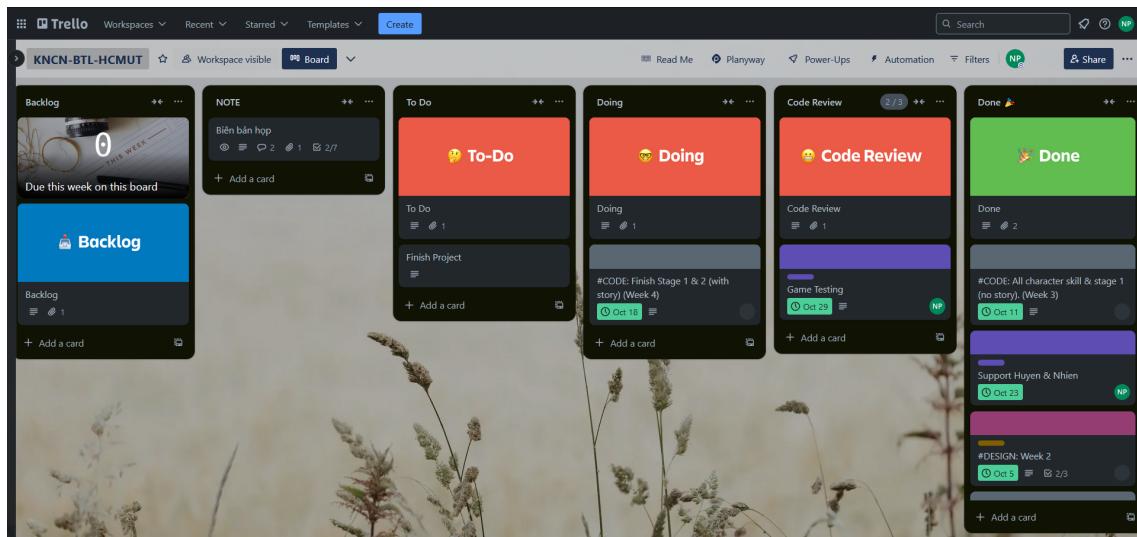
Lợi ích khi làm việc với Trello:

1. Linh hoạt khi lập kế hoạch công việc, ít tốn thời gian và công sức, thao tác trên Trello đơn giản, dễ thực hiện cũng là một điểm cộng lớn.
2. Thời gian làm việc được rút ngắn, sắp xếp có trật tự, rõ ràng và dễ tự động, dễ đánh giá trực quan.



3.1.2 Quản lý công việc với Trello

3.1.2.a Giao diện chung



Hình 2: Giao diện quản lý công việc của Trello

Giao diện quản lý công việc của **Trello** được thiết kế theo phong cách **Kanban**, giúp người dùng dễ dàng tổ chức và theo dõi tiến độ công việc. Các công việc được chia thành các bảng riêng biệt, mỗi bảng có thể chứa nhiều danh sách để phân loại công việc theo các giai đoạn khác nhau, như:

- **Backlog:** Chứa các công việc chưa được phân công hoặc chưa được ưu tiên. Đây là nơi lưu trữ các ý tưởng hoặc nhiệm vụ cần thực hiện trong tương lai.
- **Note:** Chứa các ghi chú, tài liệu tham khảo hoặc thông tin bổ sung cho công việc, giúp người dùng theo dõi các yêu cầu hoặc thông tin chi tiết.
- **To Do:** Các công việc đã được xác định và lên kế hoạch để thực hiện. Đây là danh sách chứa các nhiệm vụ cần được hoàn thành.
- **Doing:** Công việc hiện đang được thực hiện. Khi người dùng bắt đầu làm việc trên một nhiệm vụ, thẻ công việc sẽ được di chuyển vào danh sách này.
- **Code Review:** Dành cho các công việc liên quan đến việc xem xét mã nguồn. Các thẻ công việc trong mục này sẽ được kiểm tra, đánh giá và chỉnh sửa trước khi hoàn thành.
- **Done:** Các công việc đã hoàn thành. Sau khi công việc được hoàn thành và kiểm tra, nó sẽ được chuyển vào danh sách này để đánh dấu hoàn thành.

Mỗi thẻ công việc có thể chứa các thông tin chi tiết như mô tả, hạn chót, tệp đính kèm, nhãn màu, và người chịu trách nhiệm. Ngoài ra, giao diện của **Trello** rất dễ sử dụng, cho phép kéo-thả



các thẻ giữa các danh sách, tạo nên một quy trình làm việc linh hoạt và trực quan, giúp người dùng quản lý và phối hợp công việc hiệu quả.

3.1.2.b Giám sát chi tiết tiến độ công việc

The screenshot shows a Trello card for a project. The card title is "Design main character, background, environment assets". It has a status of "DONE" with a checkmark icon. Below the title are sections for "Members", "Labels", "Notifications", "Dates" (set to Sep 25 - Sep 25, 11:59 PM, marked as "Complete"), and "Description" (instructions for attaching designs using Power-ups like Figma or Invision). A "Planyway" integration is visible with a progress bar at 25% and time tracking options. A note about Google Drive access is present. The "Checklist" section contains three items: "Design main character", "Design background", and "Environment asset (tree, npc, ground,...)". The "Activity" section shows a comment from "Nhật Trần Phước" dated Sep 26, 2024, 8:00 PM, with a reward of 10 points. On the right side, there are various Power-Up options like Dependency, Estimation, Google Drive, Reports, and View on timeline, along with Automation and Share buttons.

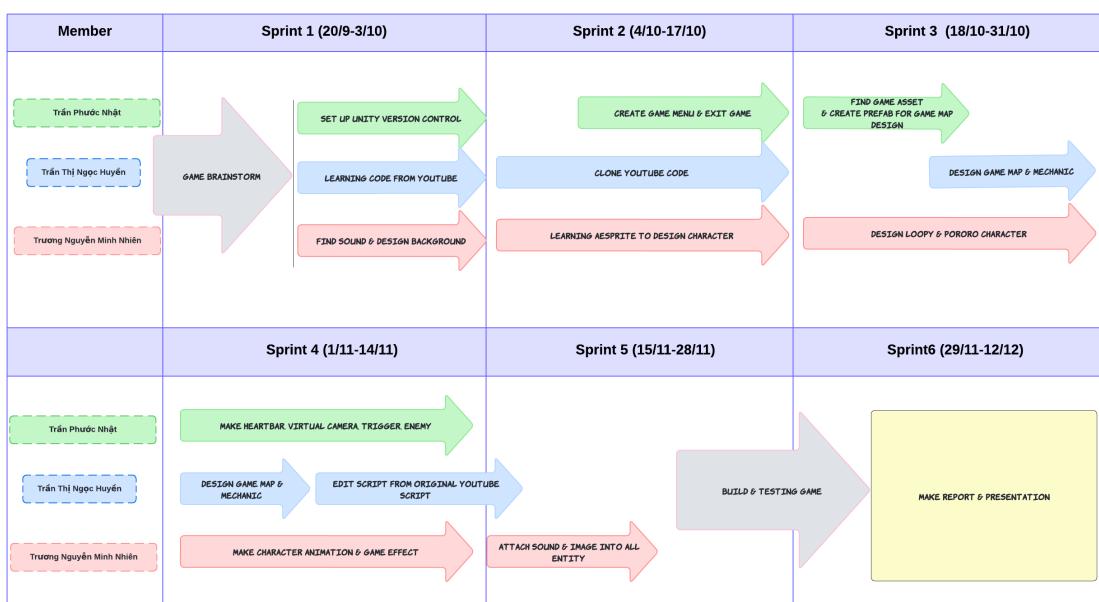
Hình 3: Giao diện chia nhỏ công việc và giám sát tiến độ của Trello

Giao diện của **Trello** cho phép chia nhỏ các công việc lớn thành nhiều **task nhỏ**, giúp giám sát tiến độ hoàn thành một cách chi tiết. Mỗi task nhỏ có thể được gắn thẻ, bổ sung mô tả, hạn

chót và trạng thái riêng biệt, giúp người quản lý dễ dàng theo dõi tiến độ của task lớn dựa trên các task nhỏ này.

Trello hỗ trợ tích hợp các **Power-ups** như Google Drive để đính kèm tài liệu liên quan trực tiếp vào thẻ công việc, giúp người dùng truy cập tài liệu dễ dàng. Ngoài ra, người dùng có thể **bình luận** và **góp ý** ngay trong các thẻ công việc, tạo không gian trao đổi thông tin và cập nhật tiến độ. Tính năng **mô tả task** giúp làm rõ yêu cầu và mục tiêu của từng task, đảm bảo mọi thành viên đều nắm được thông tin cần thiết.

3.1.2.c Timeline làm việc nhóm



Hình 4: Bảng Timeline được thiết kế dựa trên phân chia công việc trên Trello

Dự án được chia thành 6 giai đoạn (sprint) nhằm đảm bảo tiến độ và chất lượng công việc. Mỗi giai đoạn được phân chia cụ thể để tập trung vào các nhiệm vụ khác nhau, từ việc xây dựng ý tưởng ban đầu, phát triển chức năng trò chơi, đến kiểm tra và hoàn thiện sản phẩm. Dưới đây là các công việc chính trong từng giai đoạn:

- **Sprint 1 (20/9 - 3/10):** Brainstorm ý tưởng trò chơi, thiết lập môi trường làm việc, học lập trình từ tài nguyên trực tuyến, tìm kiếm âm thanh và thiết kế nền đồ họa cơ bản.
- **Sprint 2 (4/10 - 17/10):** Tạo menu chính và chức năng thoát trò chơi, sao chép và chỉnh sửa mã nguồn mẫu, đồng thời bắt đầu học thiết kế nhân vật.
- **Sprint 3 (18/10 - 31/10):** Tìm kiếm tài nguyên trò chơi, thiết kế bản đồ và cơ chế chơi, đồng thời xây dựng các nhân vật chính của trò chơi.



- **Sprint 4 (1/11 - 14/11):** Hoàn thiện bản đồ trò chơi, thiết kế hiệu ứng, thêm chức năng camera ảo và cơ chế kích hoạt kẻ thù.
- **Sprint 5 (15/11 - 28/11):** Gắn kết âm thanh và hình ảnh vào các thực thể, tiến hành xây dựng và kiểm tra trò chơi toàn diện.
- **Sprint 6 (29/11 - 12/12):** Chuẩn bị báo cáo và thuyết trình để tổng kết dự án.

3.2 Quản lý mã nguồn

3.2.1 Giới thiệu về Github

Github được ví như "mạng xã hội" của riêng lập trình viên, đây là công cụ quản lý dự án, lưu trữ code cũng như theo dõi tiến độ hiệu suất các dự án phần mềm phổ biến nhất hiện nay.

Hầu hết các dự án phần mềm hiện nay được phát triển nhờ một nhóm lập trình viên, những lập trình viên này có thể không làm việc trong cùng một khu vực địa lý, thời gian,... Việc tập trung code về một nguồn, đồng bộ code, giải quyết xung đột và theo dõi thay đổi mã nguồn, tiến độ dự án là không thể thiếu và Github chính là giải pháp được phát triển nhằm đáp ứng nhu cầu trên.



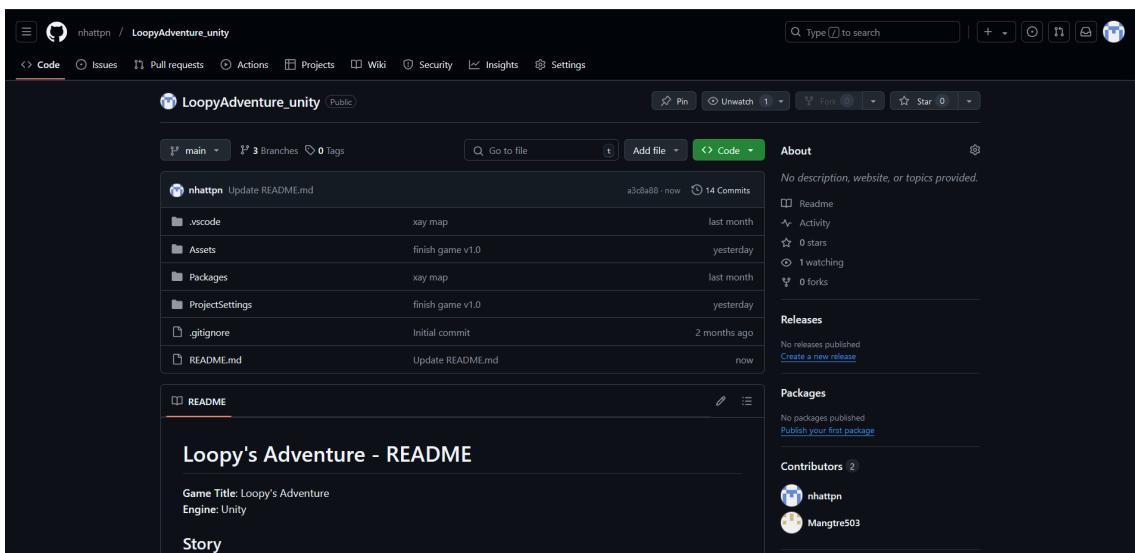
Hình 5: Logo phần mềm quản lý mã nguồn mở Github

Thao tác chính trên Github:

1. *Clone dự án về máy:* Cho phép clone mã nguồn mở public về máy.
2. *Đẩy code lên Github:* Cho phép tạo nhánh (branch) và đẩy mã nguồn lên nhánh, có chức năng commit ghi lại mô tả thay đổi trên mã nguồn.
3. *Cập nhật mã nguồn và đồng bộ dự án:* Có các câu lệnh (pull) cập nhật mã nguồn từ một nhánh xác định, đồng thời cho phép đồng bộ (merge) nhiều nhánh về một nhánh duy nhất.

3.2.2 Quản lý mã nguồn bằng Github

Giao diện repository trên GitHub là nơi người dùng có thể thấy tất cả các tệp mã nguồn của dự án, quản lý các branch, và tương tác với các thành viên trong nhóm. Repository là trung tâm của mọi hoạt động trong một dự án phần mềm, nơi chứa mọi thông tin về mã nguồn, issues, pull requests, và các vấn đề khác liên quan đến dự án.



Hình 6: Giao diện repository dự án nhóm trên GitHub

Hình trên đây minh họa giao diện của repository của nhóm trên **GitHub**. Trong đó, người dùng có thể xem được các tệp mã nguồn, các **issues** của dự án, các **pull requests** đang được xử lý và lịch sử commit của dự án. Mỗi **repository** không chỉ là nơi lưu trữ mã nguồn mà còn là công cụ giúp quản lý và theo dõi tiến độ của dự án. Ngoài việc xem các thay đổi trong mã nguồn, người dùng còn có thể tạo và quản lý các **branch**, từ đó dễ dàng làm việc trên các tính năng mới mà không ảnh hưởng đến mã nguồn chính.

Repository cũng cho phép người dùng quản lý quyền truy cập, đảm bảo chỉ những người có quyền hợp lệ mới có thể chỉnh sửa mã nguồn hoặc thực hiện các thay đổi quan trọng. Quá trình kiểm soát các thay đổi này bao gồm việc thực hiện **commit**, **merge**, và quản lý **pull requests**, giúp các thành viên trong nhóm phối hợp hiệu quả và đồng bộ hóa công việc. Bên cạnh đó, **GitHub** cung cấp các công cụ để theo dõi và xử lý các vấn đề (**issues**) trong dự án, giúp đảm bảo tiến độ và chất lượng của sản phẩm.



3.2.3 Quản lý phiên bản bằng Github

The screenshot shows the GitHub commit history for the 'main' branch. The commits are listed in chronological order, starting from the most recent at the top. Each commit includes the author, date, message, and a copy icon. Verified commits are indicated by a green 'Verified' badge.

- Commits on Dec 10, 2024:
 - Update README.md (nhattpn authored in 7 hours)
 - Update README.md (nhattpn authored in 7 hours)
- Commits on Dec 8, 2024:
 - finish game v1.0 (nhattpn committed yesterday)
- Commits on Dec 7, 2024:
 - stash (nhattpn committed 2 days ago)
- Commits on Nov 25, 2024:
 - merge code (nhattpn committed 2 weeks ago)
- Commits on Nov 23, 2024:
 - stash (nhattpn committed 2 weeks ago)
- Commits on Nov 7, 2024:
 - xay map (Mangtral03 committed on Nov 7)
- Commits on Oct 31, 2024:
 - finish copy dragon warrior to loopy (nhattpn committed on Oct 31)
- Commits on Oct 29, 2024:
 - fix folder structure (nhattpn committed on Oct 29)
 - merge code yt (nhattpn committed on Oct 29)
 - ver 0.1 (nhattpn committed on Oct 29)
- Commits on Oct 17, 2024:
 - Merge pull request #1 from nhattpn/nhatdeeptry (nhattpn authored on Oct 17)
 - initial commit (nhattpn committed on Oct 17)
 - Initial commit (nhattpn authored on Oct 17)

Hình 7: Giao diện lịch sử commit trong Github

Nhờ có tính năng lịch sử commit, **GitHub** không chỉ giúp quản lý mã nguồn mà còn hỗ trợ kiểm tra các thay đổi qua từng giai đoạn phát triển của dự án. Các thành viên trong nhóm có thể thực hiện commit, merge code, và hợp nhất công việc của mình với các thành viên khác. Hệ thống quản lý phiên bản giúp theo dõi tất cả các thay đổi trong mã nguồn, đồng thời cho phép quay lại các phiên bản trước nếu cần thiết. Điều này không chỉ giúp phát hiện và khắc phục lỗi nhanh chóng mà còn tạo ra một môi trường làm việc đồng bộ và hiệu quả giữa các thành viên.



Lịch sử commit cũng là cơ sở để người quản lý dự án theo dõi tiến độ và tình trạng của dự án, từ đó đưa ra các quyết định kịp thời.

3.3 Mô phỏng giao diện trò chơi

3.3.1 Giới thiệu về Figma

Figma là một ứng dụng hoạt động trên nền tảng web, được ứng dụng chủ yếu trong việc thiết kế giao diện web/app và trải nghiệm người dùng (thiết kế UI/UX). Với Figma, người dùng có thể thiết kế áp phích, quảng cáo, logo,... và đặc biệt là thiết kế website, app do khả năng tạo tương tác với người dùng như một bản thử nghiệm tương đối hoàn thiện mà không cần code. Nhờ vậy, đây là công cụ product design tuyệt vời đối với các dự án phần mềm.



Hình 8: Logo phần mềm biên tập đồ họa Figma

Ưu điểm của Figma trong ứng dụng vào dự án phần mềm kể đến như:

1. *Dơn giản, dễ sử dụng:* Khác với những công cụ design chuyên nghiệp PS, AI, AE các thao tác trên Figma rất dễ sử dụng, trực quan và có khả năng thực hiện các dự án cơ bản mà không tốn nhiều thời gian học tập, tìm hiểu.
2. *Khả năng tương thích:* Sử dụng dễ dàng trên laptop, điện thoại, máy tính bảng, dễ dàng truy cập và ổn định.
3. *Cho phép chia sẻ dự án:* Figma hỗ trợ làm việc theo nhóm, cho phép nhiều người cùng làm việc và chỉnh sửa trên cùng 1 đối tượng, cửa sổ, các thay đổi được đồng bộ ngay lập tức.
4. *Cho phép tạo Prototype:* Cho phép thực hiện các tác vụ chuyển trang, khởi thả xuống,... về cơ bản cho phép thao tác trên sản phẩm Figma như một sản phẩm web/app thực tế.

3.3.2 Một số hình ảnh trong figma

3.4 Thiết kế nhân vật

3.4.1 Giới thiệu về Aseprite

Aseprite là ứng dụng chỉnh sửa ảnh tạo hiệu ứng bắt mắt - như hiệu ứng Pixel, cho phép tạo hoạt ảnh 2D pixel cho các trò chơi điện tử. Aseprite có nhiều công cụ hỗ trợ hoạt ảnh (khung thê,

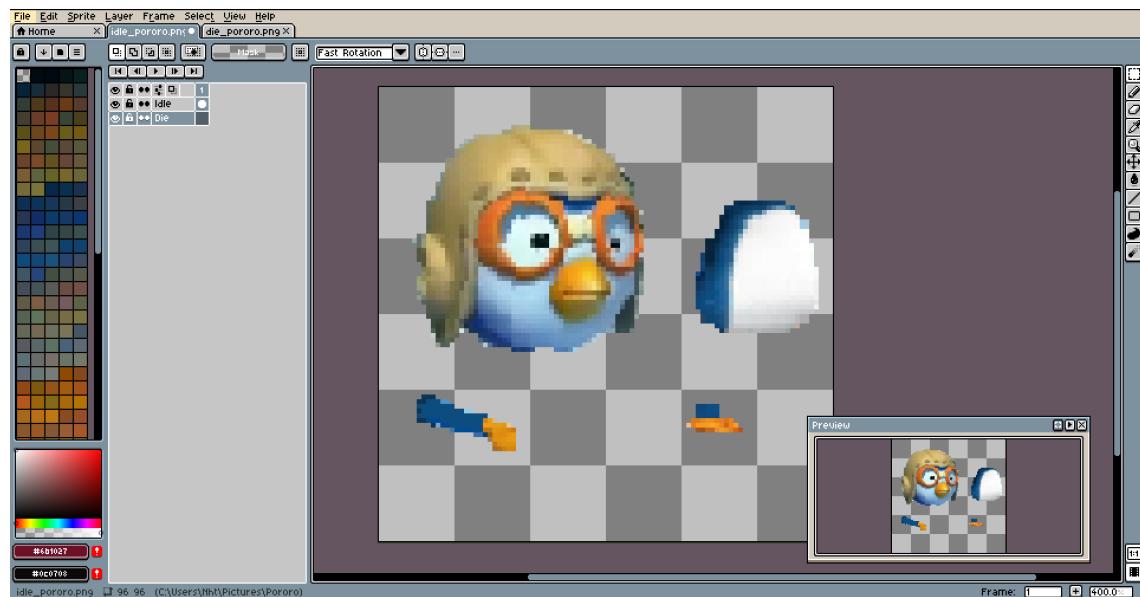


lột da củ hành, ảnh thời gian thực), công cụ nghệ thuật pixel (tô bóng, lát gạch, đường viền), cho phép chỉnh sửa màu sắc và xuất nhập nhiều dạng tệp tin khác nhau (PNG, GIF, JPG..).



Hình 9: Logo phần mềm chỉnh sửa hoạt ảnh Aseprite

3.4.2 Thiết kế animation với Aseprite



Hình 10: Giao diện làm việc của Aseprite

Giao diện làm việc của **Aseprite** được thiết kế đơn giản và trực quan, phù hợp với người dùng muốn tạo và chỉnh sửa đồ họa pixel. Phần chính của giao diện là không gian làm việc, nơi người dùng có thể xem và chỉnh sửa các khung hình (frames) của sprite. Bên trái là các công cụ vẽ, tô màu và chỉnh sửa, trong khi phía trên cung cấp các tùy chọn điều chỉnh cho các lớp (layers), frames, và các công cụ khác. Các công cụ chính như bút vẽ, tô màu, tẩy và chọn vùng đều có sẵn dễ dàng trong thanh công cụ. Bên phải, cửa sổ màu sắc và bảng điều khiển cho phép người dùng nhanh chóng chọn màu và tùy chỉnh các thiết lập. Ngoài ra, Aseprite còn hỗ trợ timeline để người dùng có thể dễ dàng quản lý và xem các khung hình chuyển động trong sprite.



3.5 Phát triển trò chơi

3.5.1 Giới thiệu về Unity

Unity - game engine đa nền tảng được phát triển bởi Unity Technologies, dùng để phát triển video game cho nhiều nền tảng như máy tính, điện thoại,... Hiện nay Unity đang chiếm khoảng 50% thị phần game, các tựa game nổi tiếng kể đến như Call of Duty, Hearthstone,...

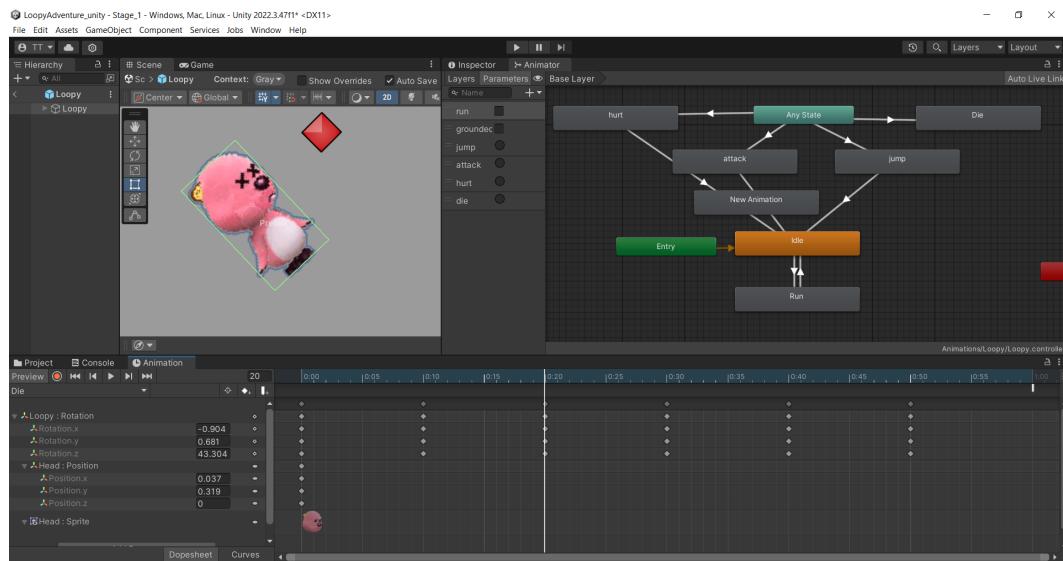


Hình 11: Logo phần mềm phát triển trò chơi Unity

Một vài ưu điểm của engine này:

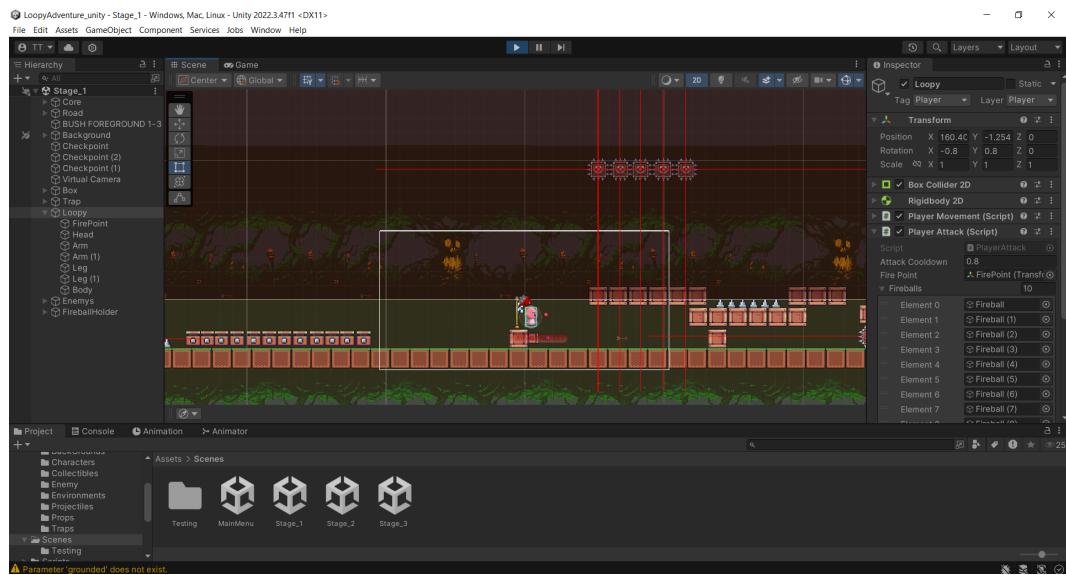
1. Editor: Cung cấp Editor riêng, các đối tượng trong game không nhất thiết phải code mà có thể dùng kéo thả để gán, sắp xếp vị trí, trình tự đối tượng.
2. Da nền tảng: Với Unity, ta có thể xây dựng game tương thích trên nhiều nền tảng như Desktop (Window và Linux), Mobile(iOS, Android) hay Web.
3. Miễn phí: Các dự án nhỏ phát triển trên Unity là miễn phí, Unity cũng không yêu cầu người dùng trả phí theo định kì như một số ứng dụng chuyên nghiệp khác.

3.5.2 Phát triển trò chơi bằng Unity



Hình 12: Thiết kế chuyển động nhân vật với Unity

Trong Unity, thiết kế animation cho nhân vật được thực hiện chủ yếu thông qua **Animator** và **Animation** window. **Animator** cho phép người dùng điều khiển và tổ chức các trạng thái animation của nhân vật, đồng thời thiết lập các chuyển tiếp giữa các trạng thái này. Các trạng thái này được thể hiện trong **Animator Controller**, nơi người dùng có thể tạo và kết nối các animation clips. Để kích hoạt các animation, Unity sử dụng **Parameters**, là các giá trị như **Trigger**, **Bool**, **Float**, hoặc **Int** để kiểm soát trạng thái và chuyển tiếp giữa các animation. **Triggers** được sử dụng để kích hoạt một animation khi điều kiện nhất định được thỏa mãn, trong khi các **Parameters** khác như **Bool** hoặc **Float** có thể được sử dụng để điều khiển hành vi phức tạp hơn, chẳng hạn như thay đổi tốc độ di chuyển hoặc kích hoạt các động tác đặc biệt khi nhân vật đạt đến một trạng thái nhất định.



Hình 13: Thiết kế màn chơi với Unity

Trong Unity, thiết kế màn chơi có thể được thực hiện dễ dàng bằng cách sử dụng các **Prefab** có sẵn. **Prefab** là các đối tượng chứa nhiều thành phần như **Sprite**, **Script**, **Sound**, **Animation** và các thành phần khác, giúp tái sử dụng và tổ chức các đối tượng trong trò chơi một cách hiệu quả. Các **Prefab** này có thể được kéo thả vào cảnh (scene) để tạo thành các đối tượng trong màn chơi, đồng thời các thuộc tính của chúng có thể được điều chỉnh tùy theo yêu cầu. **Script** được gắn vào **Prefab** để xử lý logic và hành vi, trong khi **Animation** và **Sound** cung cấp các hiệu ứng trực quan và âm thanh cho các đối tượng trong trò chơi, giúp tạo ra trải nghiệm sống động cho người chơi.



The screenshot shows the Visual Studio Code (VSCode) interface with the Unity project structure on the left sidebar. The main editor area displays two C# scripts: `UIManager.cs` and `PlayerAttack.cs`. `UIManager.cs` contains code for managing UI elements like restart and quit buttons. `PlayerAttack.cs` contains logic for player attacks, including projectile behavior and damage handling. The bottom status bar shows file paths and other development details.

```
using UnityEngine;
public class UIManager : MonoBehaviour
{
    public void Restart()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
    public void Mainmenu()
    {
        Time.timeScale = 1;
        SceneManager.LoadScene(0);
    }
    //Quit game/exit play mode if in editor
}

public class PlayerAttack : MonoBehaviour
{
    [SerializeField] private float speed;
    private float direction;
    private bool hit;
    private float lifetime;
    private Animator anim;
    private BoxCollider2D boxCollider;
    private void Awake()
    {
        anim = GetComponent();
        boxCollider = GetComponent<BoxCollider2D>();
    }
    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && Grounded())
        {
            Jump();
        }
        if (horizontalInput > 0.1f)
        {
            move_left=false;
            transform.localScale= Vector3.one;
        }
        else if (horizontalInput < -0.1f)
        {
            move_left=true;
            transform.localScale= new Vector3(-1,1,1);
        }
        anim.SetBool("IsRunning", horizontalInput!=0);
        anim.SetBool("IsGrounded", Grounded());
    }
    private void Jump()
    {
        if (Grounded())
        {
            transform.Translate(0, jumpForce, 0);
        }
    }
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.tag == "Enemy")
        {
            hit = true;
            boxCollider.enabled = false;
            anim.SetTrigger("Explode");
            collision.GetComponent<Health>()?.TakeDamage(1);
        }
    }
}
```

Hình 14: Lập trình trò chơi với Unity

Lập trình script trò chơi trong Unity thường được thực hiện bằng ngôn ngữ C#, và các công cụ như **VSCode** là lựa chọn phổ biến để viết mã. **VSCode** cung cấp môi trường phát triển mạnh mẽ với tính năng tự động hoàn thành mã (auto-completion), gợi ý mã, và hỗ trợ gỡ lỗi, giúp lập trình viên dễ dàng viết và kiểm tra mã nguồn. Khi sử dụng **VSCode**, người dùng có thể kết nối trực tiếp với Unity, làm việc hiệu quả trên các script để điều khiển hành vi của nhân vật, các đối tượng trong trò chơi, và xử lý các sự kiện trong game. Các tính năng mở rộng và plugin của **VSCode** cũng giúp tối ưu hóa quy trình phát triển, nâng cao năng suất và giảm thiểu lỗi trong mã.



4 Mô phỏng trò chơi

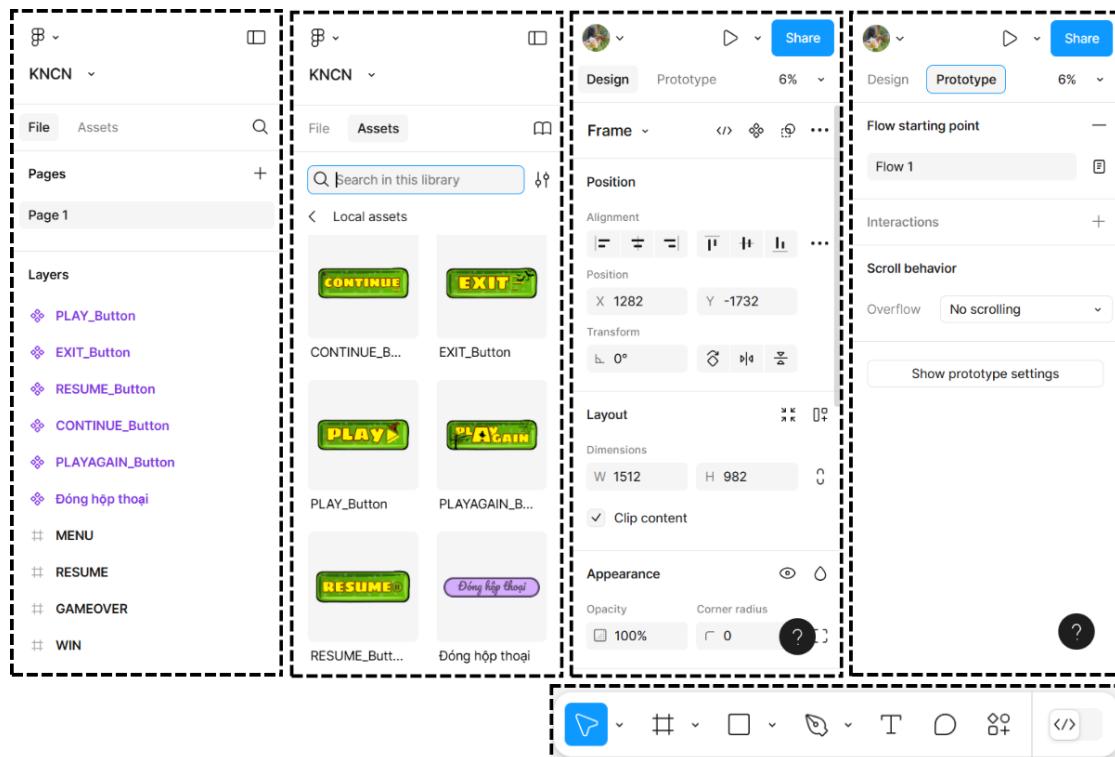
Link Figma dùng để thiết kế mô phỏng trò chơi platformer cổ điển "Loopy's Adventure": <https://bit.ly/Figma-KNCN-Loopy-Adventure>



4.1 Cấu trúc để mô phỏng game bằng Figma.

1. Các thành phần trong Figma.

- **File:** Các Pages và Layers (trong đó Layers chứa các thiết kế Prototype và Frames).
- **Assets:** Local assets chứa các Prototype.
- **Design:** Các tools hỗ trợ thiết kế giao diện.
- **Prototype:** Hỗ trợ tạo prototype.
- **Tools:** Tạo frame, vẽ hình, tạo text, comment, tìm kiếm và chuyển figma sang code.



Hình 15: Các thành phần trong Figma.



2. Các prototype để thiết kế button của trò chơi.

- **PLAY_Button:** Nút bắt đầu trò chơi (thứ thách 1).
- **EXIT_Button:** Nút thoát khỏi trò chơi.
- **RESUME_Button:** Nút tạm dừng trò chơi.
- **CONTINUE_Button:** Nút tiếp tục trò chơi sau khi đã ấn nút tạm dừng.
- **PLAYAGAIN_Button:** Nút để chơi lại trò chơi (tại THỦ THÁCH đang chơi).
- **Đóng hộp thoại:** Nút để đóng bảng thông báo/ hộp thoại trong quá trình chơi.



Hình 16: Các prototype để thiết kế button của trò chơi.

3. Các giao diện của trò chơi.

- Giao diện MENU, RESUME, GAMEOVER, WIN.
- Giao diện Stage 1, Stage 2, Stage 3, Stage BONUS.



Hình 17: Các giao diện của trò chơi.

4. Kiểm thử mô phỏng trò chơi bằng Figma.



Hình 18: Kiểm thử mô phỏng trò chơi bằng Figma.

4.2 Giao diện Menu



Hình 19: Giao diện MENU game "Loopy's Adventure".



4.3 Giao diện Resume (Tạm dừng) và GameOver (Thua)



Hình 20: Giao diện RESUME và GAMEOVER

4.4 Giao diện các thử thách trong trò chơi

1. Giao diện THỬ THÁCH 1: Kẻ địch mù phù thủy độc ác Gray Rose. (Loopy là player).

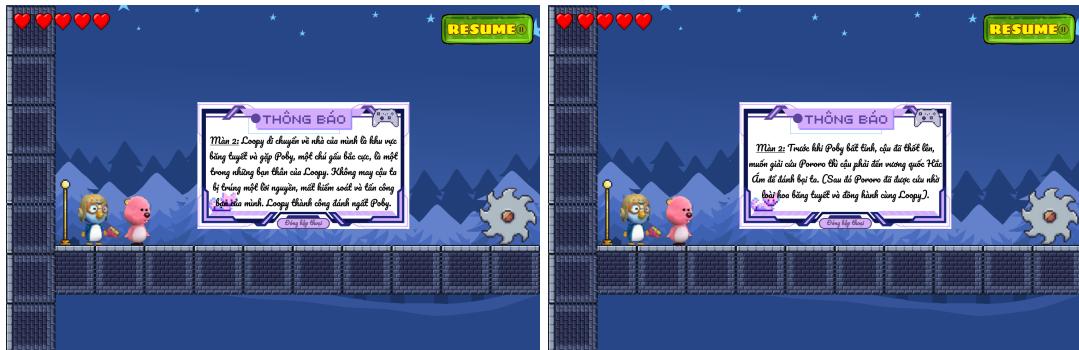


Hình 21: Giao diện GIỚI THIỆU THỬ THÁCH 1: Các hộp thoại trò chơi dẫn truyền màn 1.



Hình 22: Giao diện THỬ THÁCH 1 và VỀ DÍCH: Dánh bại Gray Rose và cứu Pororo.

2. Giao diện THỦ THÁCH 2: Kẻ địch Poby (Loopy là player).



Hình 23: Giao diện GIỚI THIỆU THỦ THÁCH 2: Các hộp thoại trò chơi dẫn truyện màn 2.



Hình 24: Giao diện THỦ THÁCH 2 và VỀ ĐÍCH: Dánh bại Poby và cứu Pororo nhờ hoa băng tuyêt.

3. Giao diện THỦ THÁCH 3: Kẻ địch Eddy Hắc Ám (Loopy là player và Pororo support).



Hình 25: Giao diện GIỚI THIỆU và THỦ THÁCH 3: Hộp thoại dẫn truyện và thử thách.

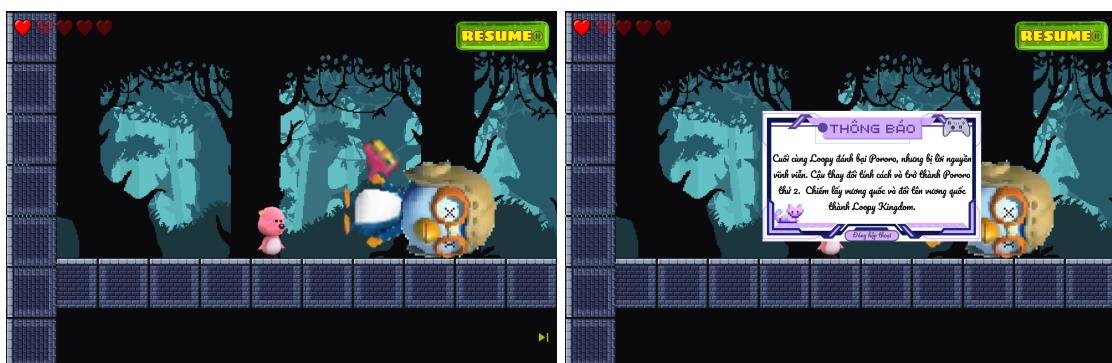


Hình 26: Giao diện VỀ DÍCH THỦ THÁCH 3 và THỦ THÁCH BONUS: *Hộp thoại dẫn truyện thủ thách phụ.*

4. Giao diện THỦ THÁCH BONUS.



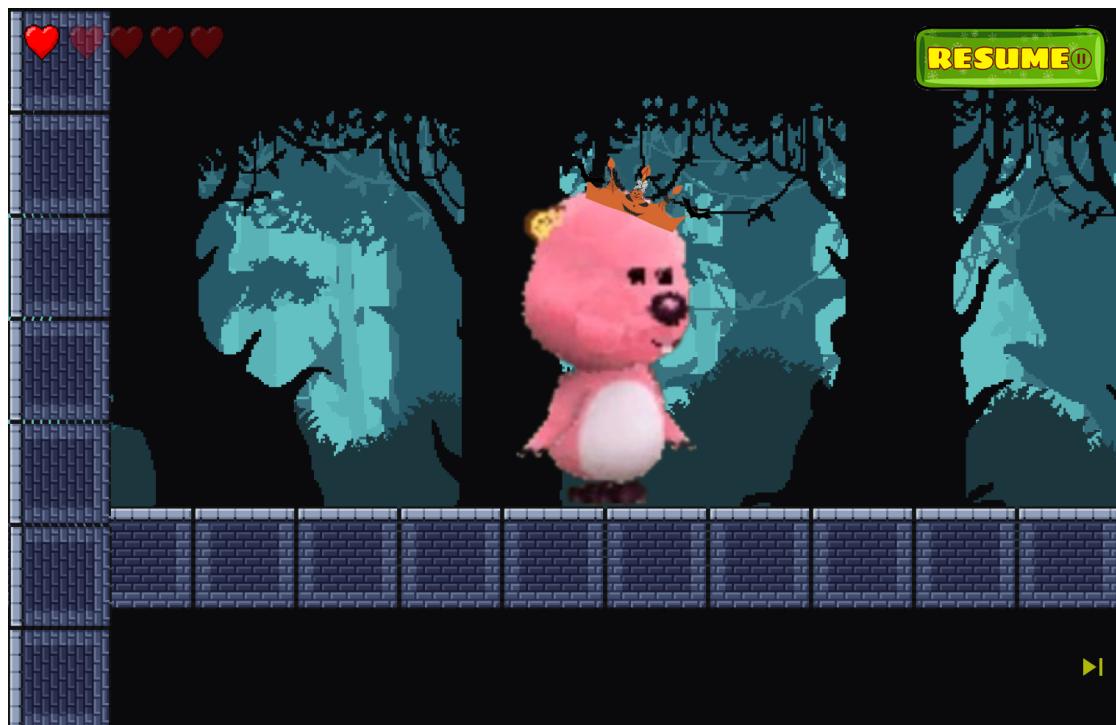
Hình 27: Giao diện THỦ THÁCH BONUS: *Hộp thoại dẫn truyện thủ thách phụ.*



Hình 28: Giao diện CHIẾN THẮNG THỦ THÁCH BONUS.



4.5 Giao diện Kết truyện của Game



Hình 29: Giao diện Kết truyện của game "Loopy's Adventure".



5 Hiện thực trò chơi

5.1 Sơ lược về cốt truyện trò chơi

Lấy cảm hứng từ dòng game platformer cổ điển Mario, **Loopy's Adventure** được xây dựng với lối chơi tương tự nhưng có cốt truyện và các nhân vật riêng.

Bối cảnh trò chơi tại Vương quốc Pororo - Pororo Kingdom, một thế giới giả tưởng đầy màu sắc và phong phú. Trong thế giới này, có nhiều loại sinh vật dễ thương, các vương quốc và những khu vực khác nhau. Mỗi khu vực lại có những phong cách riêng biệt như Vùng đất Băng Giá là quê hương của các nhân vật chính, khu vực Đầm Lầy với những động vật lưỡng cư nguy hiểm hay Khu Rừng Ma Ám là nơi sinh vật hắc ám cư ngụ.

Khởi đầu trò chơi là khi thế lực hắc ám tiến hành xâm chiếm Vương quốc Pororo và chiếm đóng trong Đầm Lầy, Rừng Ma Ám. Một trong những người bạn thân nhất của Loopy là Pororo bị bắt cóc và giam giữ trong Đầm Lầy, để cứu bạn mình nhân vật Loopy đành phải lên đường dấn thân vào Đầm Lầy. Đây là mở đầu của một hành trình đầy lý thú.

5.2 Chi tiết cốt truyện các màn chơi

1. Màn 1: Loopy chiến đấu chống lại mụ phù thủy độc ác Gray Rose nhằm giải cứu Pororo.

Loopy vượt qua Đầm Lầy giải cứu chim cánh cụt Pororo khỏi mụ phù thủy độc ác Gray Rose. Nhưng không may, Pororo đã trúng loại kịch độc bởi mụ phù thủy độc ác. Loại độc này chỉ chữa khi ăn loài hoa băng tuyết ở chính quê nhà của mình - Vùng đất Băng Giá.

2. Màn 2: Loopy lên đường trở về Vùng đất Băng Giá tìm hoa băng tuyết và gặp phải Poby hắc hóa.

Loopy di chuyển về nhà của mình là khu vực băng tuyết và gặp Poby - gấu bắc cực, là một trong những bạn thân của Loopy. Không may cậu ta bị trúng một lời nguyền, mất kiểm soát và tấn công bạn của mình. Loopy phải đánh ngất Poby để ngăn công kích. Trước khi Poby bất tỉnh, cậu đã thốt lên: "Muốn giải cứu Poby thì cậu phải đến vương quốc Hắc Ám để đánh bại ta (thế lực hắc ám)".

Loop hóa giải kịch độc cho Pororo băng hoa băng tuyết và đành phải tiếp tục lên đường.

3. Màn 3: Loopy tiến vào Khu Rừng Ma Ám tìm cách đánh bại thế lực hắc ám bí ẩn và giải lời nguyền Poby đã mắc phải.

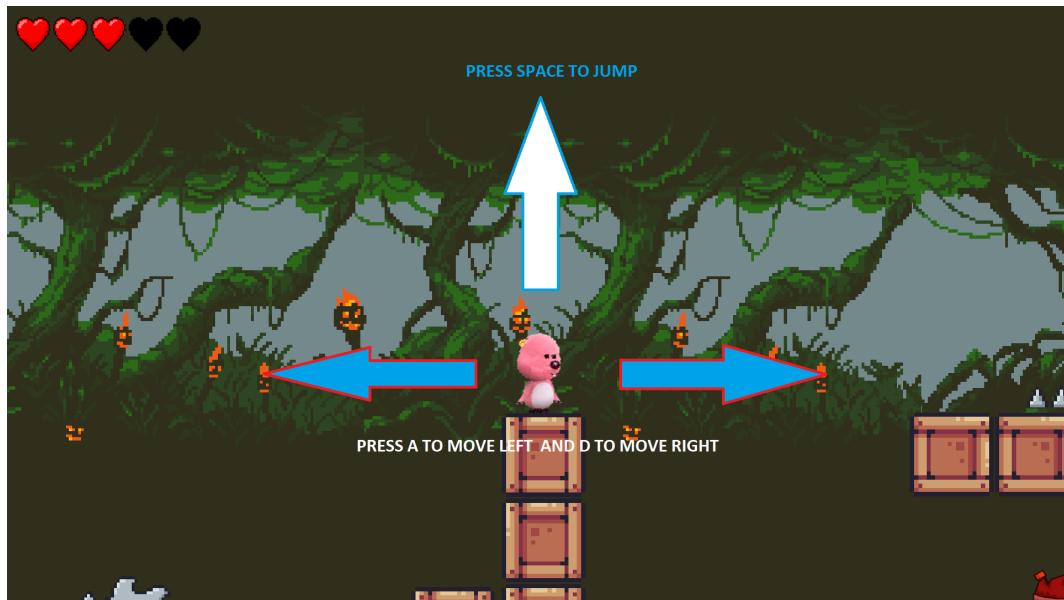
Loopy và Pororo cùng dấn thân vào Rừng Ma Ám, trải qua quá trình đầy thử thách, bị tấn công bởi các sinh vật hắc ám để tìm ra chân tướng thế lực bí ẩn sau màn. Cuối rừng, Loopy gặp Eddy, là một người bạn đã lâu không gặp, Eddy bây lâu nay bị thế lực hắc ám giam giữ và biến thành Eddy Hắc Ám.

Loopy, Pororo chiến đấu chống lại Eddy, tưởng rằng đây đã là nút thắt cuối cùng của hành trình thì bỗng dung Pororo phản chiến, trở nên hắc ám và chống lại Loopy. Hóa ra Pororo đã bị thế lực hắc ám đồng hóa từ lâu và trở thành một Hắc Ám thực thụ. Pororo lên kế hoạch để lừa những người bạn cũng trở nên Hắc Ám giống mình.

Đến đây, để kết thúc hành trình, Loopy cần đánh bại hoàn toàn Pororo, nếu Loopy thất bại, sức mạnh đặc biệt của Loopy là Loop (vòng lặp) sẽ được kích hoạt và quay ngược thời gian về điểm bắt đầu. Lúc này, Loopy đã mang kí ức của tương lai và tin chắc rằng sẽ không còn bị Pororo tấn công nữa.

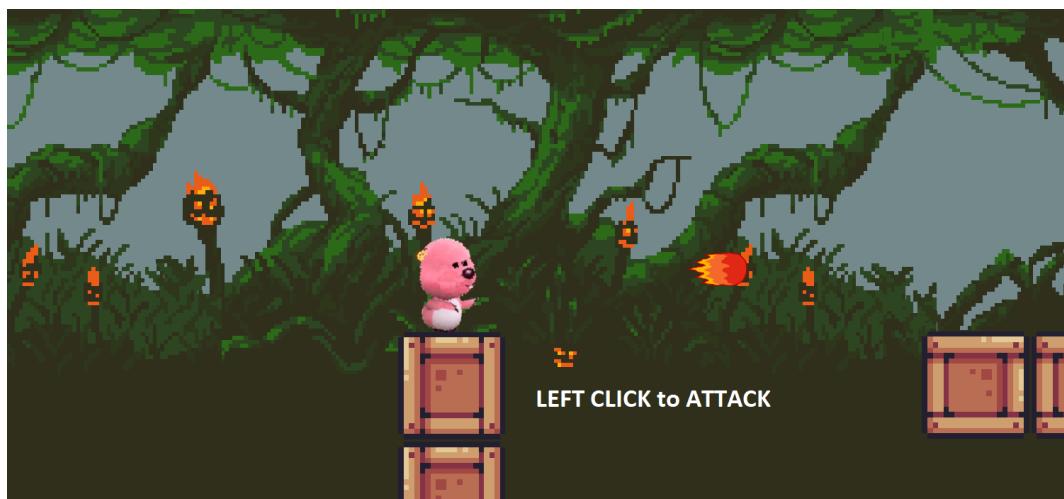
5.3 Demo sản phẩm

5.3.1 Thao tác di chuyển, tấn công



Hình 30: Thao tác di chuyển của nhân vật

Trong trò chơi, người chơi có thể di chuyển sang trái và phải bằng các phím A và D hoặc các phím mũi tên trái và phải tương ứng. Để nhảy, người chơi nhấn phím Space, và hệ thống hỗ trợ nhảy kép, cho phép người chơi nhấn Space lần thứ hai khi đang ở trên không để thực hiện một cú nhảy nữa.



Hình 31: Thao tác tấn công của nhân vật

Dể tấn công, người chơi nhấn chuột trái. Sau khi thực hiện cú tấn công đầu tiên, hệ thống sẽ chờ 0,5 giây trước khi cho phép tấn công lần thứ hai. Điều này giúp tạo ra một cơ chế tấn công theo nhịp, yêu cầu người chơi có sự chính xác và thời gian phản xạ hợp lý.

5.3.2 Chuồng ngai vật



Hình 32: Bẫy lưỡi cưa di chuyển và bẫy chông gai



Trong trò chơi, bẫy lưỡi cưa xoay tròn di chuyển qua lại, gây sát thương cho người chơi khi chạm phải. Bẫy này có thể khiến người chơi phải né tránh một cách chính xác để tránh bị tổn thương. Ngoài ra, bẫy chông sắt được đặt dưới đất, và người chơi nếu dẫm phải cũng sẽ bị sát thương, tạo ra thử thách trong việc di chuyển an toàn qua các khu vực nguy hiểm.



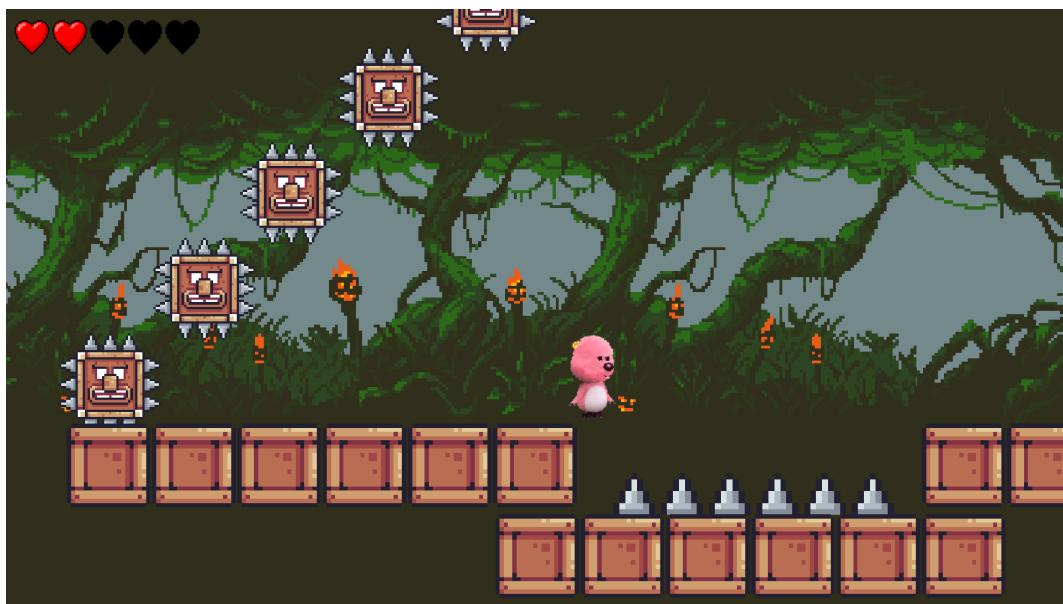
Hình 33: Bẫy mìn lửa và chạm mục tiêu gây cháy

Bẫy mìn phun lửa được kích hoạt khi người chơi đi qua. Sau khi kích hoạt từ 0,2 đến 0,4 giây, bẫy sẽ phun lửa gây sát thương cho người chơi nếu bị trúng phải. Người chơi cần nhanh chóng di chuyển để tránh bị thiệt hại từ những đợt phun lửa.



Hình 34: Kẻ địch cận chiến và kẻ địch bắn cung

Trong trò chơi, kẻ địch cầm kiếm tấn công tầm gần sẽ đứng im và chờ người chơi tiếp cận. Khi người chơi lại gần, kẻ địch sẽ lao vào tấn công với các đòn cận chiến, yêu cầu người chơi phải né tránh hoặc phản công kịp thời để tránh bị sát thương. Trong khi đó, kẻ địch bắn tên tấn công tầm xa sẽ đứng im và sử dụng cung tên để bắn vào người chơi khi người chơi tiếp cận trong phạm vi tấn công, tạo ra thử thách cho người chơi trong việc di chuyển và tránh các mũi tên từ xa.



Hình 35: Bãy chông truy đuổi theo nhân vật ở gần

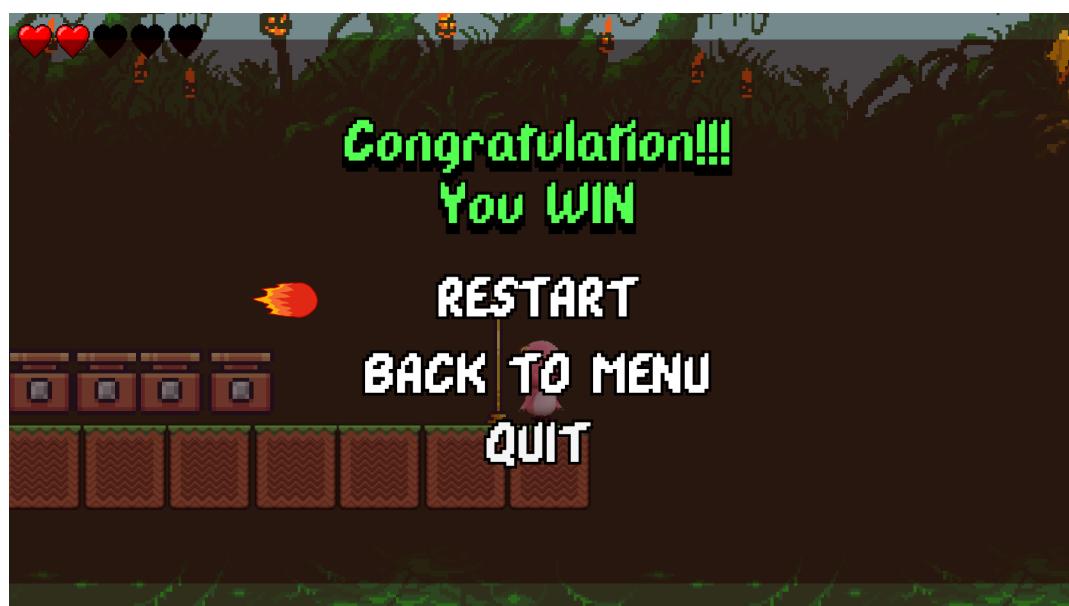
Bãy chông truy đuổi sẽ kích hoạt khi có người chơi ở gần, tự động di chuyển và dí theo người chơi theo một đường thẳng. Bãy này sẽ tiếp tục truy đuổi người chơi cho đến khi ra khỏi phạm vi tấn công hoặc bị dẫn dụ ra khỏi khu vực. Người chơi có thể lợi dụng điều này để dụ bãy bay ra khỏi khu vực, tránh gây phiền phức và bảo vệ mình khỏi sự tấn công của bãy. Cách chơi này đòi hỏi người chơi phải có chiến thuật di chuyển thông minh để thoát khỏi sự truy đuổi của bãy.

5.3.3 Kết thúc trò chơi



Hình 36: Kết thúc trò chơi khi nhân vật không đến được checkpoint

Khi người chơi không đến được checkpoint và mất hết toàn bộ máu, màn hình Game Over sẽ hiển thị. Giao diện này thông báo rằng người chơi đã thất bại và yêu cầu họ thử lại từ đầu hoặc tải lại cấp độ, tùy thuộc vào thiết lập của trò chơi. Các thông tin như số điểm, tiến độ, và lý do thất bại có thể được hiển thị để người chơi có thể đánh giá lại và cải thiện chiến lược chơi của mình.



Hình 37: Giao diện chiến thắng khi nhân vật đến được đích đến

Khi người chơi đến đích và hoàn thành cấp độ, màn hình Chiến Thắng sẽ hiển thị với thông báo Congratulations!. Người chơi có thể chọn chơi lại để thử thách bản thân hoặc kết thúc trò chơi.



6 Tài nguyên dự án

Dưới đây là các đường dẫn đến các trang liên quan đến dự án:

- GitHub Repository: [Link](#)
- Figma Design: [Link](#)

7 Tài liệu tham khảo

References

- [1] Pandemonium, [Unity 2D Platformer for Complete Beginners](#), 5/12/2020.
- [2] Aseprite, [Aseprite Tutorials](#).
- [3] Productive Dude, [Trello App: Full Trello Tutorial for Beginners in 2023!](#), 13/9/2022.
- [4] Bring Your Own Laptop, [Figma Tutorial: A Crash Course for Beginners](#), 17/3/2023.
- [5] Blackthornprod, [HOW TO MAKE 2D GAME ANIMATIONS IN UNITY - BEGINNER TUTORIAL](#), 25/02/2019.