

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------------------------

HỌ TÊN SINH VIÊN: NGUYỄN HỒNG PHÚC

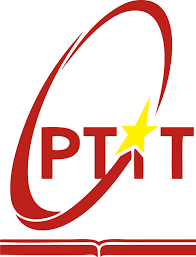
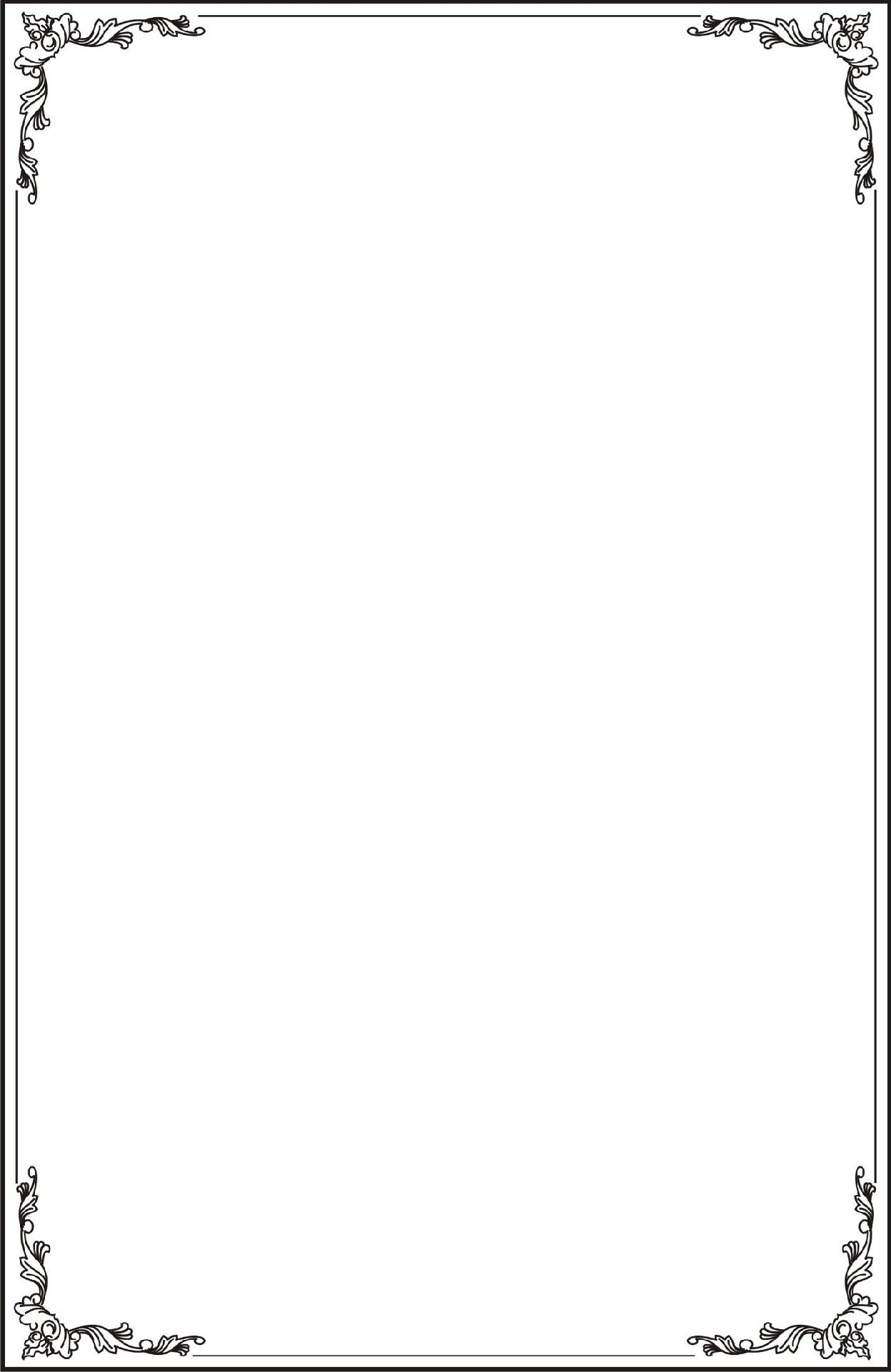
ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG ANDROID MÔ PHỎNG HỆ MẶT TRỜI**

LỚP: D19CNPM01

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên thực hiện:** | Nguyễn Hồng Phúc |
| **Mã sinh viên:** | B19DCCN505 |
| **Lớp:** | D19CNPM1 |
| **Niên khóa:** | 2019-2024 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2023**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------------

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG ANDROID MÔ PHỎNG HỆ MẶT TRỜI**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên:** | Nguyễn Hồng Phúc |
| **Mã sinh viên:** | B19DCCN505 |
| **Lớp:** | D19CNPM1 |
| **Niên khóa:** | 2019-2024 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2022**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20

**Giảng viên hướng dẫn**

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20

**Giảng viên phản biện**

**LỜI CẢM ƠN**

Đầu tiên, em xin cảm ơn các thầy, cô giáo của Học Viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô trong khoa Công nghệ thông tin 1 đã dạy dỗ và chỉ bảo em trong suốt thời gian học tập tại trường.

Em xin chân thành cảm ơn ThS. Đinh Xuân Trường đã quan tâm, hướng dẫn và tận tình chỉ dạy em trong suốt quá trình thực hiện đề tài và viết báo cáo đồ án này.

Mặc dù đã cố gắng hoàn thành bài báo cáo nhưng trong quá trình thực hiện không thể tránh được những thiếu sót, em kính mong nhận được những đánh giá , nhận xét của các thầy cô để có thể hoàn thiện sản phẩm của mình hơn nữa trong tương lai.

Em xin chân thành cảm ơn!

**Hà Nội, tháng 12 năm 2023**

**Sinh viên**

**Nguyễn Hồng Phúc****TÓM TẮT NỘI DUNG ĐỒ ÁN**

Phần mềm mô phỏng hệ Mặt Trời là một công cụ cung cấp thông tin về các  
thiên thể trong hệ Mặt Trời, giả lập chuyển động quỹ đạo của các hành tinh, giả  
lập các tương tác vật lý cơ bản giữa chúng, xây dựng bản đồ sao nhằm hỗ trợ người  
dùng xác định vị trí của các thiên thể. Phần mền được xây dựng trên công cụ Unity  
Engine để có thể tạo ra các mô hình 3D modeling cho Mặt Trời và các hành tinh  
đồng thời tận dụng được khả năng xử lý vật lý mà engine mang lại, bên cạnh đó,  
phần mềm cũng cung cấp cho phép người dùng thực hiện một số tương tác thú vị  
trên các thiên thể. Em đã chọn đề tài này vì sự quan tâm lớn đối với thiên văn học  
và việc hiểu về hệ Mặt Trời là một phần quan trọng của việc nghiên cứu và giảng  
dạy trong lĩnh vực này. Ứng dụng giả lập hệ Mặt Trời sẽ giúp người dùng khám phá và hiểu rõ hơn về cấu trúc và vận hành của hệ Mặt Trời một cách thú vị thông qua các tương tác trực quan.

**MỤC LỤC**

[LỜI MỞ ĐẦU 13](#_Toc123246387)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 13](#_Toc123246388)

[1.1 Khảo sát hiện trạng của ngành lập trình ở Việt Nam 13](#_Toc123246389)

[1.2 Xác định mục tiêu đề tài 13](#_Toc123246390)

[1.3 Mục tiêu nghiên cứu 14](#_Toc123246391)

[1.4 Phương pháp thực hiện 14](#_Toc123246392)

[1.5 Các công nghệ được sử dụng 14](#_Toc123246393)

[1.5.1 Tổng quan về Javascript 14](#_Toc123246394)

[1.5.2 Tổng quan về Node.js 15](#_Toc123246395)

[1.5.3 Tổng quan về ReactJs 16](#_Toc123246396)

[1.5.4 Tổng quan về REST và RESTful API 17](#_Toc123246397)

[1.5.5 Tổng quan về NoSQL và MongoDB 19](#_Toc123246398)

[1.6 Các công cụ 19](#_Toc123246399)

[1.6.1 Visual Studio Code 19](#_Toc123246400)

[1.6.2 Postman 20](#_Toc123246401)

[1.6.3 Chrome DevTools 20](#_Toc123246402)

[1.6.4 Mongo Atlas 21](#_Toc123246403)

[1.6.5 Cloudinary 21](#_Toc123246404)

[1.7 Tổng kết chương 21](#_Toc123246405)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 22](#_Toc123246406)

[2.1 Phân tích hệ thống 22](#_Toc123246407)

[2.1.1 Tên hệ thống 22](#_Toc123246408)

[2.1.2 Mục tiêu của hệ thống: 22](#_Toc123246409)

[2.1.3 Các tác nhân của hệ thống 22](#_Toc123246410)

[2.1.4 Yêu cầu của hệ thống: 22](#_Toc123246411)

[2.1.5 Phạm vi của hệ thống 22](#_Toc123246412)

[2.2 Thiết kế hệ thống 23](#_Toc123246413)

[2.2.1 Usecase tổng quan 23](#_Toc123246414)

[2.2.2 Biểu đồ lớp thực thể 24](#_Toc123246415)

[2.3 Phân rã các module 24](#_Toc123246416)

[2.3.1 Module đăng ký 24](#_Toc123246417)

[2.3.2 Module đăng nhập 27](#_Toc123246418)

[2.3.3 Module đăng xuất 29](#_Toc123246419)

[2.3.4 Module đổi mật khẩu 30](#_Toc123246420)

[2.3.5 Module xem danh sách bài toán 32](#_Toc123246421)

[2.3.6 Module lọc bài toán theo tiêu chí 34](#_Toc123246422)

[2.3.7 Module xem chi tiết bài toán 35](#_Toc123246423)

[2.3.8 Module xem xếp hạng thành viên 37](#_Toc123246424)

[2.3.9 Module chạy test 38](#_Toc123246425)

[2.3.10 Module submit lời giải 40](#_Toc123246426)

[2.3.11 Module xem lịch sử submit 42](#_Toc123246427)

[2.3.12 Bình luận về bài toán 43](#_Toc123246428)

[2.3.13 Module trả lời bình luận 45](#_Toc123246429)

[2.3.14 Module thích bình luận 46](#_Toc123246430)

[2.3.15 Module xem contest 48](#_Toc123246431)

[2.3.16 Đăng ký tham gia contest 49](#_Toc123246432)

[2.3.17 Tham gia contest 51](#_Toc123246433)

[2.3.18 Xem contest đã làm 53](#_Toc123246434)

[2.3.19 Module thêm danh mục 54](#_Toc123246435)

[2.3.20 Module sửa danh mục 56](#_Toc123246436)

[2.3.21 Module xóa danh mục 58](#_Toc123246437)

[2.3.22 Module thêm bài toán 60](#_Toc123246438)

[2.3.23 Module sửa bài toán 62](#_Toc123246439)

[2.3.24 Module thêm contest 63](#_Toc123246440)

[2.4 Thiết kế cơ sở dữ liệu 66](#_Toc123246441)

[2.4.1 Lược đồ cơ sở dữ liệu 66](#_Toc123246442)

[2.4.2 Mô tả cơ sở dữ liệu 66](#_Toc123246443)

[2.5 Tổng kết 71](#_Toc123246444)

[CHƯƠNG 3. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG 72](#_Toc123246445)

[3.1 Yêu cầu hệ thống 72](#_Toc123246446)

[3.2 Một số công cụ, thư viện hỗ trợ 72](#_Toc123246447)

[3.3 Cài đặt 72](#_Toc123246448)

[3.3.1 Cài đặt Nodejs 72](#_Toc123246449)

[3.3.2 Đăng ký Mongo Atlas 73](#_Toc123246450)

[3.3.3 Đăng ký Cloudinary 74](#_Toc123246451)

[3.3.4 Cài đặt mã nguồn phía backend 75](#_Toc123246452)

[3.3.5 Cài đặt mã nguồn phía frontend 75](#_Toc123246453)

[3.4 Kết quả cài đặt 76](#_Toc123246454)

[3.4.1 Kết quả cài đặt database 76](#_Toc123246455)

[3.4.2 Kết quả cài đặt website 81](#_Toc123246456)

[3.5 Kết luận 88](#_Toc123246457)

[PHẦN TỔNG KẾT 89](#_Toc123246458)

[1. Đánh giá kết quả của đồ án 89](#_Toc123246459)

[2. Phương hướng phát triển 89](#_Toc123246460)

[TÀI LIỆU THAM KHẢO 90](#_Toc123246461)

[PHỤ LỤC 91](#_Toc123246462)

[A. Chi tiết sơ đồ mô phỏng 91](#_Toc123246463)

**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Diễn giải** |
| 1 | API | Application Programming Interface |
| 2 | REST | REpresentational State Transfer |
| 3 | ECMA | European Computer Manufacturers Association |
| 4 | CPU | Central Processing Unit |
| 5 | HTML | HyperText Markup Language |
| 6 | JSX | Javascript Extensible Markup Language |
| 7 | MVC | Model-View-Controller |
| 8 | HATEOAS | Hypermedia As The Engine Of Application State |
| 9 | HTTP | HyperText Transfer Protocol |
| 10 | SQL | Structured Query Language |
| 11 | noSQL | Not-only SQL or No-relation SQL |
| 12 | IDE | Integrated Development Environment |

**DANH MỤC HÌNH VẼ**

[Hình 1.1 Khảo sát về công nghệ được sử dụng rộng rãi nhất năm 2022 [1] 15](#_Toc123246464)

[Hình 1.2 Tổng quan về nodejs [2] 16](#_Toc123246465)

[Hình 1.3 Danh sách thư viện được sử dụng rộng rãi nhất năm 2022 [1] 17](#_Toc123246466)

[Hình 1.4 Tổng quan về REST [3] 18](#_Toc123246467)

[Hình 1.5 Visual Studio Code 19](#_Toc123246468)

[Hình 1.6 Postman 20](#_Toc123246469)

[Hình 1.7 Chrome DevTools 20](#_Toc123246470)

[Hình 1.8 Mongo Atlas Site 21](#_Toc123246471)

[Hình 1.9 Cloudinary Site 21](#_Toc123246472)

[Hình 2.1 Usecase tổng quan của hệ thống 23](#_Toc123246473)

[Hình 2.2 Biểu đồ lớp thực thể của hệ thống 24](#_Toc123246474)

[Hình 2.3 Usecase chức năng đăng ký 24](#_Toc123246475)

[Hình 2.4 Biểu đồ tuần tự chức năng đăng ký (1) 26](#_Toc123246476)

[Hình 2.5 Usecase chi tiết module đăng nhập 27](#_Toc123246477)

[Hình 2.6 Biểu đồ tuần tự module đăng nhập 28](#_Toc123246478)

[Hình 2.7 Biểu đồ tuần tự module đăng xuất 30](#_Toc123246479)

[Hình 2.8 Usecase chi tiết module đổi mật khẩu 30](#_Toc123246480)

[Hình 2.9 Biểu đồ tuần tự module đổi mật khẩu 32](#_Toc123246481)

[Hình 2.10 Usecase chi tiết module xem danh sách bài toán 32](#_Toc123246482)

[Hình 2.11 Biểu đồ tuần tự module xem danh sách bài toán 33](#_Toc123246483)

[Hình 2.12 Usecase chi tiết module lọc bài toán 34](#_Toc123246484)

[Hình 2.13 Biểu đồ tuần tự module lọc bài toán 35](#_Toc123246485)

[Hình 2.14 Usecase tiết module xem chi tiết bài toán 35](#_Toc123246486)

[Hình 2.15 Biểu đồ tuần tự module xem chi tiết bài toán 36](#_Toc123246487)

[Hình 2.16 Usecase chi tiết chức năng xem xếp hạng thành viên 37](#_Toc123246488)

[Hình 2.17 Biểu đồ tuần tự module xem xếp hạng thành viên 38](#_Toc123246489)

[Hình 2.18 Usecase chi tiết module chạy test 38](#_Toc123246490)

[Hình 2.19 Biểu đồ tuần tự module chạy test 39](#_Toc123246491)

[Hình 2.20 Usecase chi tiết module submit lời giải 40](#_Toc123246492)

[Hình 2.21 Biểu đồ tuần tự module submit lời giải (2) 41](#_Toc123246493)

[Hình 2.22 Usecase chi tiết module xem lịch sử submit 42](#_Toc123246494)

[Hình 2.23 Biểu đồ tuần tự module xem lịch sử submit 43](#_Toc123246495)

[Hình 2.24 Usecase chi tiết module bình luận về bài toán 43](#_Toc123246496)

[Hình 2.25 Biểu đồ tuần tự module bình luận về bài toán 44](#_Toc123246497)

[Hình 2.26 Usecase chi tiết module trả lời bình luận 45](#_Toc123246498)

[Hình 2.27 Biểu đồ tuần tự module trả lời bình luận 46](#_Toc123246499)

[Hình 2.28 Usecase chi tiết module thích bình luận 46](#_Toc123246500)

[Hình 2.29 Biểu đồ tuần tự module thích bình luận 48](#_Toc123246501)

[Hình 2.30 Usecase chi tiết module xem contest 48](#_Toc123246502)

[Hình 2.31 Biểu đồ tuần tự module xem contest 49](#_Toc123246503)

[Hình 2.32 Usecase chi tiết module đăng ký tham gia contest 49](#_Toc123246504)

[Hình 2.33 Biểu đồ tuần tự module đăng ký tham gia contest 50](#_Toc123246505)

[Hình 2.34 Usecase chi tiết module tham gia contest 51](#_Toc123246506)

[Hình 2.35 Biểu đồ tuần tự module tham gia contest (3) 52](#_Toc123246507)

[Hình 2.36 Usecase chi tiết module xem contest đã làm 53](#_Toc123246508)

[Hình 2.37 Biểu đồ tuần tự module xem contest đã làm) 54](#_Toc123246509)

[Hình 2.38 Usecase chi tiết module thêm danh mục 54](#_Toc123246510)

[Hình 2.39 Biểu đồ tuần tự module thêm danh mục (6) 56](#_Toc123246511)

[Hình 2.40 Usecase chi tiết module sửa danh mục 56](#_Toc123246512)

[Hình 2.41 Biểu đồ tuần tự module sửa danh mục 58](#_Toc123246513)

[Hình 2.42 Usecase chi tiết module xóa danh mục 58](#_Toc123246514)

[Hình 2.43 Biểu đồ tuần tự module xóa danh mục 59](#_Toc123246515)

[Hình 2.44 Usecase chi tiết module thêm bài toán 60](#_Toc123246516)

[Hình 2.45 Biểu đồ tuần tự module thêm bài toán 61](#_Toc123246517)

[Hình 2.46 Usecase chi tiết module sửa bài toán 62](#_Toc123246518)

[Hình 2.47 Biểu đồ tuần tự module sửa bài toán 63](#_Toc123246519)

[Hình 2.48 Usecase chi tiết module thêm contest 63](#_Toc123246520)

[Hình 2.49 Biểu đồ tuần tự module thêm contest  (4) 65](#_Toc123246521)

[Hình 2.50 Lược đồ cơ sở dữ liệu 66](#_Toc123246522)

[Hình 3.1 Hướng dẫn tải Nodejs 72](#_Toc123246523)

[Hình 3.2 Màn hình cài đặt Nodejs 73](#_Toc123246524)

[Hình 3.3 Giao diện trang đăng ký Mongo Atlas 73](#_Toc123246525)

[Hình 3.4 Giao diện quản lý trang Mongo Atlas 74](#_Toc123246526)

[Hình 3.5 Giao diện trang hướng dẫn kết nối 74](#_Toc123246527)

[Hình 3.6 Giao diện đăng ký cloudinary 74](#_Toc123246528)

[Hình 3.7 Cấu trúc môi trường của dự án backend 75](#_Toc123246529)

[Hình 3.8 Kết quả cài đặt database bằng Mongo Atlas 76](#_Toc123246530)

[Hình 3.9 Kết quả cài đặt bảng users 76](#_Toc123246531)

[Hình 3.10 Kết quả cài đặt bảng uploads 77](#_Toc123246532)

[Hình 3.11 Kết quả cài đặt bảng testcases 77](#_Toc123246533)

[Hình 3.12 Kết quả cài đặt bảng submissions 78](#_Toc123246534)

[Hình 3.13 Kết quả cài đặt bảng Replies 78](#_Toc123246535)

[Hình 3.14 Kết quả cài đặt bảng problems 79](#_Toc123246536)

[Hình 3.15 Kết quả cài đặt bảng contests 79](#_Toc123246537)

[Hình 3.16 Kết quả cài đặt bảng contest\_histories 80](#_Toc123246538)

[Hình 3.17 Kết quả cài đặt bảng comments 80](#_Toc123246539)

[Hình 3.18 Kết quả cài đặt bảng categories 81](#_Toc123246540)

[Hình 3.19 Giao diện trang xem bài toán 81](#_Toc123246541)

[Hình 3.20 Giao diện xem chi tiết bài toán – chạy test 82](#_Toc123246542)

[Hình 3.21 Giao diện xem chi tiết bài toán 82](#_Toc123246543)

[Hình 3.22 Giao diện xem contest 83](#_Toc123246544)

[Hình 3.23 Giao diện làm contest (1) 83](#_Toc123246545)

[Hình 3.24 Giao diện làm contest (2) 84](#_Toc123246546)

[Hình 3.25 Giao diện xem lịch sử contest 84](#_Toc123246547)

[Hình 3.26 Giao diện xem lịch sử submit 85](#_Toc123246548)

[Hình 3.27 Giao diện xem xếp hạng 86](#_Toc123246549)

[Hình 3.28 Giao diện quản lý thông tin cá nhân 86](#_Toc123246550)

[Hình 3.29 Giao diện quản lý contest 86](#_Toc123246551)

[Hình 3.30 Giao diện quản lý danh mục 87](#_Toc123246552)

[Hình 3.31 Giao diện thêm danh mục 87](#_Toc123246553)

[Hình 3.32 Giao diện quản lý submissions 87](#_Toc123246554)

[Hình 3.33 Giao diện thêm contest 88](#_Toc123246555)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1 Kịch bản module đăng ký 25](#_Toc123246556)

[Bảng 2.2 Kịch bản module đăng nhập 27](#_Toc123246557)

[Bảng 2.3 Kịch bản module đăng xuất 29](#_Toc123246558)

[Bảng 2.4 Kịch bản module đổi mật khẩu 31](#_Toc123246559)

[Bảng 2.5 Kịch bản module xem danh sách bài toán 33](#_Toc123246560)

[Bảng 2.6 Kịch bản module lọc bài toán theo tiêu chí 34](#_Toc123246561)

[Bảng 2.7 Kịch bản module xem chi tiết bài toán 35](#_Toc123246562)

[Bảng 2.8 Kịch bản module xem xếp hạng thành viên 37](#_Toc123246563)

[Bảng 2.9 Kịch bản module chạy test 38](#_Toc123246564)

[Bảng 2.10 Kịch bản module submit lời giải 40](#_Toc123246565)

[Bảng 2.11 Kịch bản module xem lịch sử submit 42](#_Toc123246566)

[Bảng 2.12 Kịch bản module bình luận về bài toán 43](#_Toc123246567)

[Bảng 2.13 Kịch bản module trả lời bình luận 45](#_Toc123246568)

[Bảng 2.14 Kịch bản module thích bình luận 47](#_Toc123246569)

[Bảng 2.15 Kịch bản module xem contest 48](#_Toc123246570)

[Bảng 2.16 Kịch bản module đăng ký tham gia contest 50](#_Toc123246571)

[Bảng 2.17 Kịch bản module tham gia contest 51](#_Toc123246572)

[Bảng 2.18 Kịch bản module xem contest đã làm 53](#_Toc123246573)

[Bảng 2.19 Kịch bản module thêm danh mục 55](#_Toc123246574)

[Bảng 2.20 Kịch bản module sửa danh mục 57](#_Toc123246575)

[Bảng 2.21 Kịch bản module xóa danh mục 59](#_Toc123246576)

[Bảng 2.22 Kịch bản module thêm bài toán 60](#_Toc123246577)

[Bảng 2.23 Kịch bản module sửa bài toán 62](#_Toc123246578)

[Bảng 2.24 Kịch bản module thêm contest 64](#_Toc123246579)

[Bảng 2.25 Cơ sở dữ liệu bảng users 66](#_Toc123246580)

[Bảng 2.26 Cơ sở dữ liệu bảng upload 67](#_Toc123246581)

[Bảng 2.27 Cơ sở dữ liệu bảng category 67](#_Toc123246582)

[Bảng 2.28 Cơ sở dữ liệu bảng comment 68](#_Toc123246583)

[Bảng 2.29 Cơ sở dữ liệu bảng reaction 68](#_Toc123246584)

[Bảng 2.30 Cơ sở dữ liệu bảng contest 68](#_Toc123246585)

[Bảng 2.31 Cơ sở dữ liệu bảng problem 69](#_Toc123246586)

[Bảng 2.32 Cơ sở dữ liệu bảng ContestHistory 70](#_Toc123246587)

[Bảng 2.33 Cơ sở dữ liệu bảng testcase 70](#_Toc123246588)

[Bảng 2.34 Cơ sở dữ liệu bảng submission 70](#_Toc123246589)

# LỜI MỞ ĐẦU

Ngành thiên văn học bắt đầu với sự quan sát thiên nhiên và vũ trụ từ thời kỳ cổ đại. Mục đích nghiên cứu thiên văn trong thời kỳ này là để dự đoán các hiện tượng thiên văn như chấm dứt của một chuỗi ngày đêm, các mùa trong năm và các sự kiện thiên văn khác. Ngày này ngành thiên văn học sử dụng các kỹ thuật tiên tiến để quan sát, sử dụng máy tính mô phỏng để nghiên cứu về nguồn gốc, cấu trúc của vũ trụ và khả năng tồn tại sự sống ngoài vũ trụ. Thiên văn học là một lĩnh vực khoa học có nhiều ứng dụng quan trọng trong cuộc sống hiện đại:

* Định vị và định hướng:Thiên văn học đã giúp phát triển các hệ thống định vị toàn cầu như GPS (Global Positioning System), cho phép chúng ta xác định vị trí và định hướng với độ chính xác cao.
* Dự báo thời tiết: theo dõi các hiện tượng thiên văn như cấu trúc mây và biểu đồ nhiệt độ của các hành tinh trong hệ Mặt Trời có thể cung cấp thông tin hữu ích cho dự báo thời tiết.
* Công nghiệp hàng không và không gian: hỗ trợ việc thiết kế, dự đoán, và điều khiển các vệ tinh và tàu vũ trụ.
* Giáo dục và tạo đam mê: Thiên văn học là nguồn cảm hứng không giới hạn cho giáo dục và tạo đam mê đối với khoa học trong các thế hệ trẻ. Các chương trình giảng dạy và sự kiện thiên văn học thú vị giúp truyền đạt kiến thức về vũ trụ và khoa học thiên văn cho mọi người.

Tuy nhiên tại Việt Nam, ngành thiên văn học còn chưa được chú trọng và quantâm. Chính vì thế em muốn xây dựng một ứng dụng cung cấp nhưng kiến thức, thông tin cơ bản về thiên văn(cụ thể ở đây là hệ Mặt Trời) nhằm giúp mọi người có cái nhìn mới về một ngành khoa học còn non trẻ tại Việt Nam, đồng thời giúp những người có sở thích về thiên văn có thêm một công cụ để tìm hiểu nhiều hơn về các thiên thể trong hệ Mặt Trời.

Đồ án sẽ tập trung trình bày một số nội dung chính như sau:

* **Chương 1: Giới thiệu:**

Nội dung chương 1 sẽ mô tả khái quát về đề tài, giới thiệu chung về các công nghệ, ngôn ngữ cũng như engine sử dụng: C#, OpenGL, Unity và các frame work liên quan.

* **Chương 2: Phân tích và thiết kế hệ thống:**

Nội dung của chương 2 sẽ trình bày về phân tích thiết kế các chức năng của hệ thống.

* **Chương 3: Cài đặt hệ thống:**

Nội dung của chương 3 sẽ tập trung vào việc hướng dẫn cài đặt các công cụ liên quan và chi tiết các bước cài đặt source code của hệ thống và những kết quả đạt được sau khi cài đặt thành công hệ thống.

* **Phần tổng kết:**

Tổng kết bài toán, tóm tắt những kết quả đã đạt được và còn chưa đạt được. Từ đó đề xuất mục tiêu hướng tới cũng như hướng nghiên cứu, phát triển tiếp theo.

# GIỚI THIỆU CHUNG

Trong chương này, đồ án sẽ trình bày về hiện trạng, mức độ quan tâm tới ngành vật lý thiên văn trên thế giới nói chung và tại Việt Nam nói riêng đồng thời giới thiệu tổng quan về đề tài: Xây dựng ứng dụng Android mô phỏng hệ Mặt Trời.

## Khảo sát hiện trạng của ngành lập trình ở Việt Nam

Theo báo cáo thị trường của *TopDev*, năm 2021 Việt Nam cần 450.000 nhân lực ngành công nghệ thông tin. Hiện tại tổng số lập trình viên ở Việt Nam là vào khoảng 430.000 nhân lực, tương đương với việc đang thiếu hụt khoảng 20.000 nhân lực. Năm 2022, Việt Nam sẽ thiếu đến 150.000 nhân lực khi nhu cầu thị trường tăng lên đến 530.000 người.

Theo *Jobstreet.com,* lập trình viên đang nằm trong top 10 những ngành nghề có mức lương cao nhất tại Việt Nam.  Dải lương phổ biến của lập trình viên từ 8,4 triệu đồng đến 27,4 triệu đồng. Với những sinh viên mới tốt nghiệp chuyên ngành công nghệ thông tin, lương cơ bản nằm trong khoảng 8 - 10 triệu đồng. Những người 1 - 2 năm kinh nghiệm sẽ nhận lương mỗi tháng khoảng 10 - 15 triệu đồng. Con số này có thể lên hơn 20 triệu đồng nếu có tối thiểu 3 năm kinh nghiệm và kỹ năng chuyên sâu, đa dạng.

Theo báo *Thanh Niên*, số liệu xét tuyển năm nay cho thấy Công nghệ thông tin là một trong số những ngành thu hút nhiều thí sinh đăng ký ở nhiều trường. Kết quả này không quá bất ngờ bởi trên thực tế, ngành công nghệ thông tin đang có sức hút lớn với mức lương hấp dẫn và nhu cầu về nhân lực cao.

Tuy nhiên, theo *career.gpo.vn* chỉ có khoảng 16.500 sinh viên trong tổng số 55.000 sinh viên chuyên ngành công nghệ thông tin tốt nghiệp mỗi năm có khả năng đáp ứng được nhu cầu của doanh nghiệp (khoảng 30%).

Nguyên nhân thiếu hụt được cho là xuất phát từ sự chênh lệch giữa trình độ của lập trình viên và yêu cầu của doanh nghiệp về những kỹ năng cần thiết như khả năng lập trình và giải quyết các vấn đề liên quan đến các thuật toán trong thực tế.

## Xác định mục tiêu đề tài

* Xây dựng mô hình hệ Mặt Trời cung cấp một cách trực quan về thông tin thông qua các hình ảnh và model 3D.
* Cung cấp hệ thống giả lập quỹ đạo cho các hành tinh trong hệ Mặt Trời.
* Cung cấp hệ thống bản đồ sao chi tiết.
* Thực hiện bản địa hóa cho các ngôn ngữ khác nhau.
* Đối tượng hướng đến: những người quan tâm và yêu thích tới vật lý thiên văn.

## Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của đồ án bao gồm các mục tiêu chính:

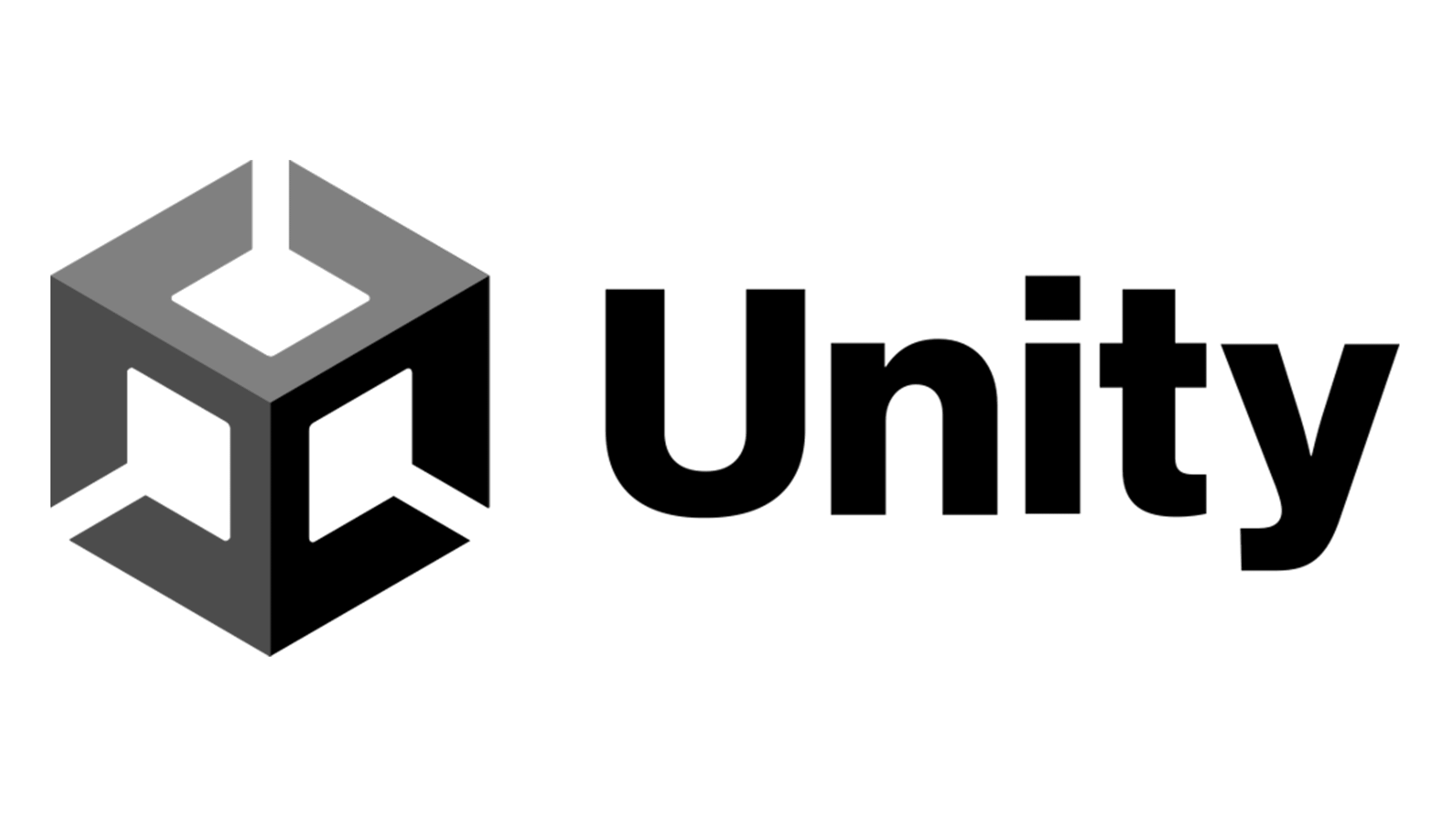
* Tìm hiểu về kiến trúc và các design partem trong việc thiết kế ứng dụng Android.
* Tìm hiểu về ngôn ngữ lập trình C# thông qua việc xây dựng ứng dụng Android.
* Tìm hiểu OpenGL(một API trừu tượng về đồ họa 2D và 3D) thông qua Shader và Shader Graph được tích hợp sẵn trong Unity.
* Áp dụng những kiến thức đã tìm hiểu để xây dựng một hệ thống giúp người sử dụng tìm hiểu về lĩnh vực vật lý thiên văn.

## Phương pháp thực hiện

* Khảo sát, tìm hiểu và xác định những thông tin cần thiết để hình thành ý tưởng, phạm vi, yêu cầu cần đạt của đề tài.
* Dựa vào phạm vi, yêu cầu để thực hiện phân tích thiết kế hệ thống.
* Tìm hiểu các công nghệ, kỹ thuật để đáp ứng được yêu cầu của đề tài.
* Dựa trên những kiến thức tìm hiểu được, tiến hành xây dựng và triển khai đề tài.
* Đánh giá kết quả đạt được để từ đó đưa ra phương hướng phát triển trong tương lai cho đề tài.

## Các công nghệ được sử dụng

### Tổng quan về Unity Engine



Unity là một game engine [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) được phát triển bởi Unity Technologies, mà chủ yếu để phát triển video game cho máy tính, consoles và điện thoại. Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng.

Unity hỗ trợ đồ họa 2D và 3D, các chức năng được viết chủ yếu qua ngôn ngữ [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng). Hai ngôn ngữ lập trình khác cũng được hỗ trợ: Boo, đã bị loại cùng với việc phát triển Unity 5 và UnityScript bị loại vào tháng 8 năm 2017 sau khi phát hành Unity 2017.1UnityScript là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh_k%E1%BB%8Bch_b%E1%BA%A3n) độc quyền có cú pháp tương tự [JavaScript](https://vi.wikipedia.org/wiki/JavaScript).

Phần mềm nhắm mục tiêu các đồ họa APIs sau: Direct3D trên Windows và Xbox One; [OpenGL](https://vi.wikipedia.org/wiki/OpenGL) trên Linux, macOS, và Windows; OpenGL ES trên Android và iOS; WebGL trên web; và APIs độc quyền trên các [máy chơi video game](https://vi.wikipedia.org/wiki/Video_game_console).

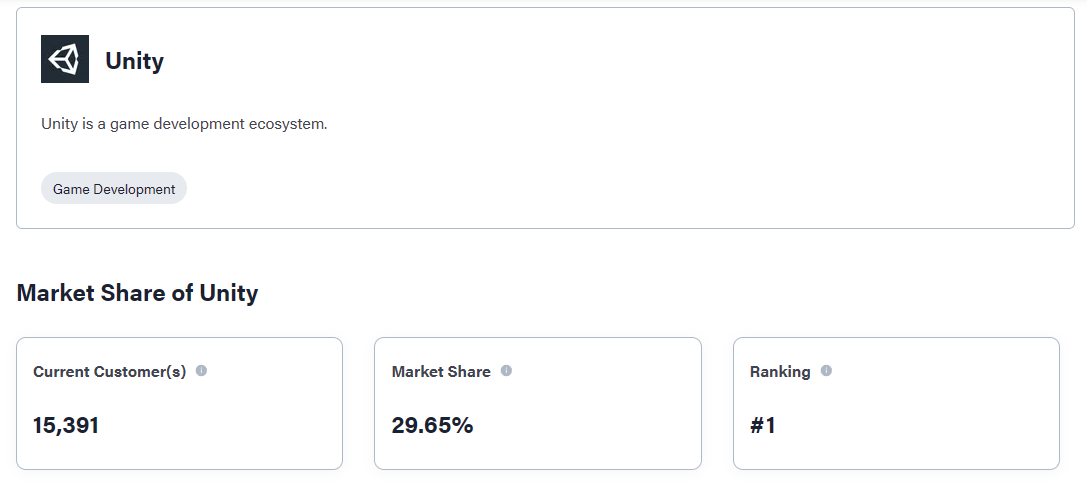
Ngoài ra, Unity hỗ trợ APIs cấp thấp như Metal trên iOS và macOS và Vulkan trên Android, Linux, và Windows, cũng như Direct3D 12 trên Windows và Xbox One.

Trong 2D games, Unity cho phép nhập sprites và một renderer thế giới 2D tiên tiến. Đối với 3D games, Unity cho phép thiết lập các đập điểm kỹ thuật của các kết cấu và độ phân giải mà công cụ trò chơi hỗ trợ, cung cấp các hỗ trợ cho bump mapping, reflection mapping, parallax mapping, cảnh không gian ambient occlusion (SSAO), hiệu ứng bóng đổ bằng cách sử dụng shadow maps, render thiết lập toàn cảnh đến hiệu ứng.

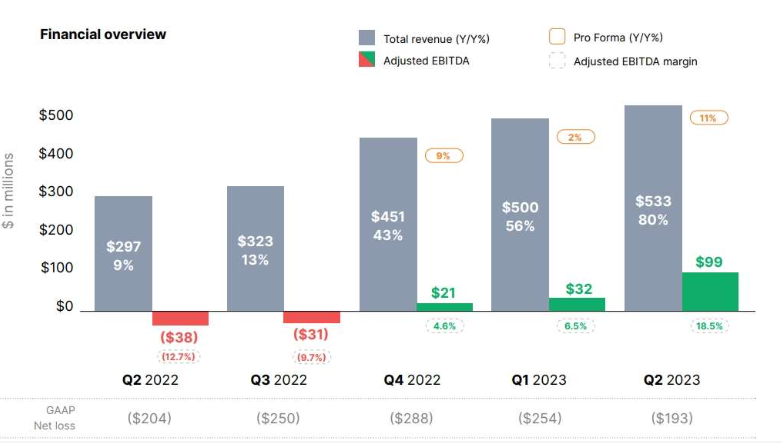
Unity cũng cung cấp các dịch vụ cho nhà phát triển, bao gồm: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity API, Unity Multiplayer, Unity Performance Reporting and Unity Collaborate.

Một số thống kê về Unity

* Tính đến quý 3 năm 2016 đã có hơn 5 tỉ lượt tải các ứng dụng được phát triển bởi Unity.
* 2,4 tỉ thiết bị di động đã từng cài ít nhất một ứng dụng bởi Unity.
* Trong top 1000 các sản phẩm game mobiles miễn phí thì số lượng game được tạo ra bởi Unity chiếm tới 34%.



* Theo báo cáo quý 2, Unity đạt doanh thu 533 triệu đô la trong quý, tăng 80% so với năm trước. Tạo giải pháp (công cụ, đăng ký, v.v.) đạt doanh thu 193 triệu đô la, tăng 17% so với năm trước (do hoạt động kinh doanh đăng ký của Unity thúc đẩy). Grow Solutions (quảng cáo, kiếm tiền) đạt doanh thu 340 triệu USD, tăng 157% so với năm trước.



Ưu điểm của Unity:

* Cộng đồng lớn và hỗ trợ mạnh mẽ: Có một cộng đồng lớn của những người phát triển sử dụng Unity và UnityEngine, cung cấp tài nguyên học tập, diễn đàn, và hỗ trợ.
* Tiết kiệm thời gian và tài nguyên: Unity cung cấp nhiều công cụ và tính năng giúp giảm thời gian phát triển và tối ưu hóa quy trình làm việc.
* Asset Store: Unity Asset Store cung cấp một cách thuận tiện để mua và bán tài nguyên, công cụ, và giao diện người dùng, giúp những người phát triển tiết kiệm thời gian và công sức.
* Đồ họa và hiệu ứng: Unity hỗ trợ đồ họa 2D và 3D mạnh mẽ, cung cấp khả năng tạo ra trải nghiệm đồ họa và hiệu ứng đa dạng.
* Độ Linh Hoạt Cao: Unity hỗ trợ nhiều ngôn ngữ lập trình như C#, JavaScript, và Boo, cho phép lập trình viên lựa chọn ngôn ngữ mà họ thoải mái nhất.
* Hỗ trợ VR/AR: Unity hỗ trợ phát triển trò chơi và ứng dụng thực tế ảo (VR) và thực tế ảo (AR), giúp nhà phát triển tham gia vào lĩnh vực công nghiệp này.

### Tổng quan về C#

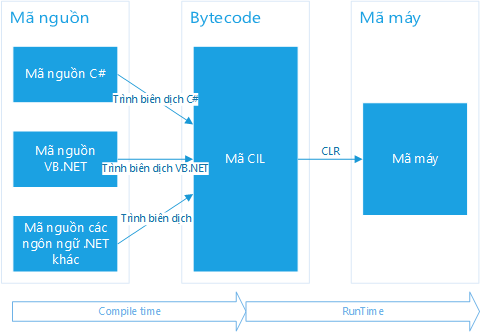
C# (hay còn gọi là C Sharp, là ngôn ngữ lập trình hướng đối tượng, hiện đại do Microsoft phát triển như một phần của nền tảng .NET. Nó được giới thiệu lần đầu tiên vào năm 2000 và kể từ đó đã trở thành một trong những ngôn ngữ lập trình được sử dụng rộng rãi nhất trong ngành.

C# có thiết kế đơn giản, mạnh mẽ và linh hoạt. Sở hữu một cú pháp tương tự như các ngôn ngữ lập trình phổ biến, bao gồm Java và C++, giúp các nhà phát triển đã quen thuộc với các ngôn ngữ này dễ làm quen.

Một trong những tính năng chính của C# là an toàn kiểu của nó, giúp ngăn ngừa các lỗi lập trình có thể xảy ra với các ngôn ngữ khác. Không những vậy, nó còn hỗ trợ quản lý bộ nhớ tự động thông qua trình thu gom rác, giúp dễ dàng viết mã đáng tin cậy mà không làm rò rỉ bộ nhớ hay gây ra bất kỳ sự cố nào.

Nhờ những điều đó, nên C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows dành cho máy tính để bàn, game, ứng dụng web và các ứng dụng dành cho thiết bị di động. Ngoài ra, nó còn được sử dụng trong phát triển phần mềm doanh nghiệp, với nhiều công ty lớn sử dụng C# để xây dựng các ứng dụng kinh doanh của họ.

Với tổng quan những điều trên, thì chúng ta dễ dàng nhận thấy C# là một ngôn ngữ mạnh mẽ và linh hoạt, phù hợp với nhiều nhiệm vụ lập trình. Cho dù bạn mới bắt đầu làm quen với lập trình hay là một nhà phát triển có kinh nghiệm, thì C# luôn là một ngôn ngữ tuyệt vời để học và sử dụng khi muốn xây dựng các ứng dụng mạnh mẽ, đáng tin cậy và có thể mở rộng.



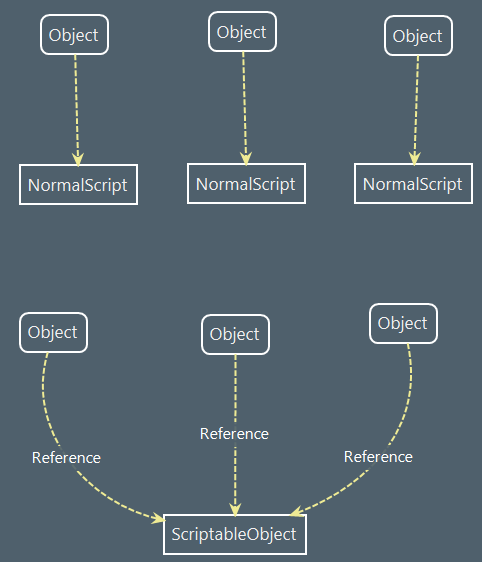
C# có những đặc điểm nổi bật như sau:

* Ngôn ngữ lập trình C# (C Sharp) là một ngôn ngữ lập trình đa mô hình (multi-paradigm) được phát triển bởi Microsoft. Dưới đây là một số đặc điểm nổi bật của ngôn ngữ C#:
* Tích hợp với Hệ điều hành Windows và Microsoft.NET: C# được thiết kế chủ yếu để phát triển ứng dụng chạy trên nền tảng Windows và tích hợp tốt với .NET Framework và .NET Core.
* Ngôn ngữ lập trình hướng đối tượng (OOP): C# hỗ trợ các nguyên tắc lập trình hướng đối tượng như kế thừa, đa hình, và đóng gói, giúp tổ chức mã nguồn và tăng tính tái sử dụng.
* Cú pháp rõ ràng và đơn giản: Cú pháp của C# được thiết kế để dễ đọc, dễ hiểu và giảm thiểu lỗi.
* Quản lý bộ nhớ tự động: C# sử dụng mô hình quản lý bộ nhớ tự động thông qua Garbage Collector, giảm gánh nặng cho lập trình viên khi phải quản lý bộ nhớ thủ công.
* Thư viện chuẩn rộng lớn: C# đi kèm với một số lượng lớn các thư viện chuẩn, cùng với sự hỗ trợ mạnh mẽ từ .NET Framework, giúp lập trình viên phát triển ứng dụng nhanh chóng và hiệu quả.
* Hỗ trợ đa nền tảng: C# đã phát triển để hỗ trợ đa nền tảng thông qua .NET Core và sau này là .NET 5 và các phiên bản mới hơn, cho phép việc phát triển ứng dụng chạy trên nhiều hệ điều hành như Windows, Linux, và macOS.
* Kiểu dữ liệu an toàn: C# hỗ trợ kiểu dữ liệu an toàn, giúp giảm thiểu lỗi liên quan đến kiểu dữ liệu trong quá trình thực thi.
* Delegates và Events: Delegates và Events là tính năng quan trọng của C#, giúp xử lý sự kiện và triển khai các mô hình lập trình không đồng bộ (asynchronous programming).
* LINQ (Language Integrated Query): LINQ là một tính năng quan trọng giúp thực hiện các truy vấn trực tiếp trong mã nguồn C#, giúp giảm độ phức tạp và tăng tính tổ chức của mã nguồn.
* ASP.NET: C# được sử dụng chủ yếu để phát triển ứng dụng web thông qua ASP.NET, một framework mạnh mẽ cho việc xây dựng các ứng dụng web và dịch vụ web.

Những đặc điểm này làm cho C# trở thành một ngôn ngữ lập trình phổ biến trong cộng đồng phần mềm và phát triển ứng dụng.

### Tổng quan về Scriptable Object

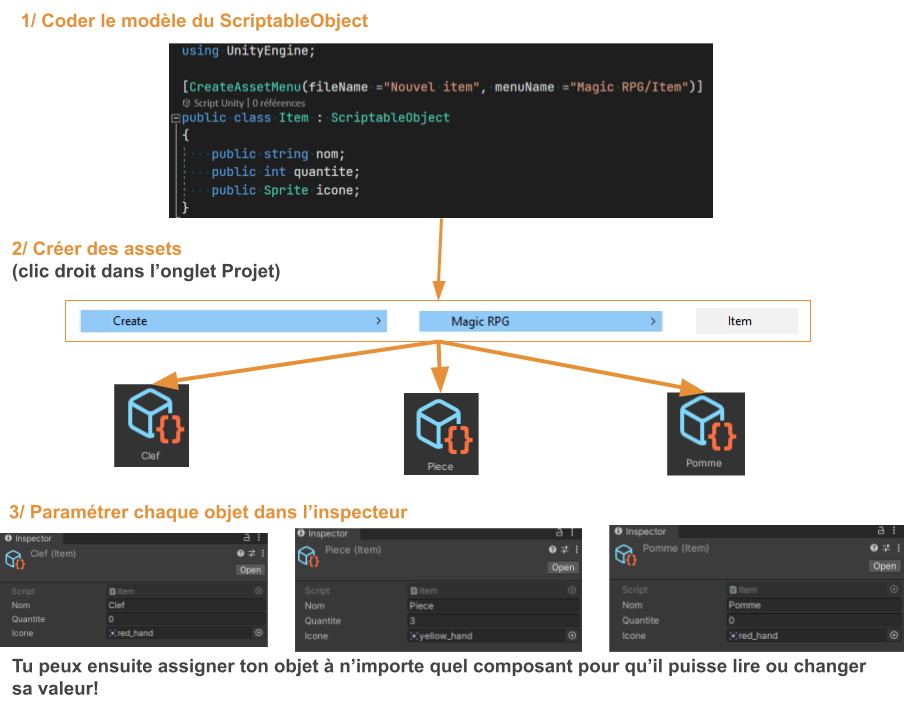
ScriptableObject là một bộ chứa dữ liệu có thể được sử dụng để lưu trữ lượng lớn dữ liệu, độc lập với các class. Một trong những trường hợp sử dụng chính của ScriptableObjects là giảm bớt việc sử dụng bộ nhớ trong Dự án bằng cách tránh sao chép các giá trị. Điều này hữu ích nếu Dự án có một Prefab lưu trữ dữ liệu không thay đổi trong các script MonoBehaviour đính kèm.



Mỗi khi thể hiện Prefab đó, nó sẽ có bản sao riêng của dữ liệu đó. Thay vì sử dụng phương thức này và lưu trữ dữ liệu trùng lặp, có thể sử dụng một ScriptableObject để lưu trữ dữ liệu và sau đó truy cập nó theo tham chiếu từ tất cả các Prefabs. Điều này có nghĩa là có một bản sao duy nhất của dữ liệu trong bộ nhớ

Tương tự như MonoBehaviours, ScriptableObjects được dẫn xuất từ đối tượng cơ sở Unity, nhưng khác với MonoBehaviours, bạn không thể đính kèm một ScriptableObject vào một GameObject. Thay vào đó, bạn cần lưu chúng dưới dạng Tài nguyên trong Dự án của bạn.

Dữ liệu lưu từ Unity engine vào ScriptableObjects như một tài nguyên được ghi vào đĩa và nó bền vững giữa các phiên làm việc.



Có nhiều cách để lưu trữ dữ liệu trong xây dựng ứng dụng Android, nhưng với Unity, Scriptable Object là một phương thức dữ liệu offine tối ưu.Việc lưu trữ bằng ScriptableObject trong Unity có nhiều ưu điểm so với việc lưu trữ dưới dạng các định dạng khác như JSON. Dưới đây là một số lợi ích chính:

* Tích hợp Tốt với Unity: ScriptableObject được tích hợp chặt chẽ với hệ thống Asset và Editor của Unity. Điều này giúp dễ dàng tạo, chỉnh sửa và quản lý các đối tượng ScriptableObject trong môi trường phát triển của Unity.
* Dễ Dàng Sử Dụng Trong Editor: Các đối tượng ScriptableObject có thể được tạo và chỉnh sửa trong Editor, giúp nhà phát triển thực hiện điều chỉnh nhanh chóng mà không cần chạy trò chơi.
* Khả Năng Quản Lý Mã Nguồn: Dữ liệu trong ScriptableObject được tổ chức tốt và có thể dễ dàng quản lý từ mã nguồn. Điều này làm cho mã nguồn của bạn trở nên dễ đọc và bảo trì.
* Tái Sử Dụng Dữ Liệu: Dữ liệu trong ScriptableObject có thể được chia sẻ và tái sử dụng giữa nhiều đối tượng khác nhau mà không cần tạo bản sao của dữ liệu. Điều này giúp giảm bớt lượng bộ nhớ được sử dụng.
* Khả Năng Kế Thừa và Mở Rộng: Bạn có thể kế thừa từ ScriptableObject để tạo các lớp con mới, giúp bạn mở rộng chức năng và cấu trúc của dữ liệu dễ dàng hơn.
* Hiệu Suất: Truy cập dữ liệu từ ScriptableObject có thể nhanh chóng hơn so với việc đọc và phân tích cú pháp các định dạng như JSON hoặc XML. ScriptableObject được tối ưu hóa cho việc sử dụng trong Unity.
* Hỗ Trợ Tốt Cho Prefab và Serialization: ScriptableObject rất phù hợp với việc lưu trữ dữ liệu cho Prefabs và hỗ trợ tốt cho quá trình serialization của Unity.

## Các công cụ

### JetBrains Rider

JetBrains Rider là một môi trường phát triển tích hợp (IDE) do JetBrains phát triển, chuyên dành cho lập trình viên sử dụng các ngôn ngữ lập trình liên quan đến Microsoft .NET. Được phát hành lần đầu vào năm 2016, Rider nhanh chóng trở thành một lựa chọn phổ biến cho các nhà phát triển .NET. Dưới đây là một số điểm đặc sắc của JetBrains Rider:



* Hỗ Trợ Nhiều Ngôn Ngữ: Rider không chỉ hỗ trợ C# mà còn hỗ trợ một loạt các ngôn ngữ lập trình khác như F#, VB.NET, HTML, CSS, JavaScript, TypeScript, và SQL.
* Tích Hợp Với .NET và ASP.NET: Rider được thiết kế để hoạt động chặt chẽ với .NET Framework và .NET Core, giúp bạn phát triển ứng dụng Windows, web, và dịch vụ điện toán đám mây dễ dàng. Cross-Platform:
* Rider có khả năng chạy trên nhiều hệ điều hành, bao gồm Windows, macOS và Linux, giúp nhà phát triển linh hoạt trong việc chọn lựa nền tảng phát triển.
* Đầy Đủ Tính Năng: Rider cung cấp một loạt các tính năng hỗ trợ mã nguồn, gỡ lỗi, kiểm thử, quản lý phiên bản, và nhiều tính năng khác để tối ưu hóa quy trình phát triển.
* Code Analysis và Refactoring: Rider tích hợp các công cụ phân tích mã nguồn mạnh mẽ, cung cấp gợi ý và khuyến nghị để cải thiện chất lượng mã nguồn. Nó cũng hỗ trợ nhiều thao tác tái cấu trúc mã nguồn (refactoring).
* Hỗ Trợ Unity: JetBrains Rider hỗ trợ phát triển trò chơi và ứng dụng sử dụng Unity Game Engine, cung cấp tính năng làm việc với C# trong môi trường phát triển Unity.
* Tích Hợp Git và VCS: Rider tích hợp tốt với các hệ thống quản lý mã nguồn như Git và hệ thống kiểm soát phiên bản khác, giúp quản lý và theo dõi sự thay đổi mã nguồn dễ dàng.
* UI Thân Thiện và Tùy Chỉnh: Giao diện người dùng của Rider được thiết kế để tối ưu hóa trải nghiệm người dùng, và người dùng có thể tùy chỉnh môi trường làm việc theo ý muốn.
* Cập Nhật Thường Xuyên: JetBrains thường xuyên cập nhật Rider để hỗ trợ các tính năng mới của .NET và cải thiện hiệu suất và trải nghiệm người dùng.
* Đồng Tích Hợp với JetBrains Toolbox: Rider có thể tích hợp chặt chẽ với JetBrains Toolbox, giúp quản lý cài đặt và cập nhật IDE cũng như các công cụ khác của JetBrains.

## Tổng kết chương

Như vậy, ở chương 1 đồ án đã giới thiệu tổng quan về đề tài bao gồm những nội dung sau:

* Khảo sát về hiện trạng thực tế của ngành thiên văn học tại Việt Nam nói riêng và thế giới nói chung
* Xác định mục tiêu và mục tiêu nghiên cứu của đề tài.
* Xác định phương pháp để thực hiện đề tài.
* Giới thiệu về các công nghệ và công cụ được sử dụng để thực hiện đề tài.

Dựa vào bài toán đã được đặt ra ở chương này, chương tiếp theo đồ án sẽ thực hiện xây dựng phân tích và thiết kế cho hệ thống.

# CƠ SỞ LÝ THUYẾT

**Lý thuyết quỹ đạo của các hành tinh**

Trên thực tế, việc tính toán quỹ đạo của các hành tinh trong hệ Mặt Trời một cách chính xác là việc vô cùng khó, phức tạp đòi hỏi sự chính xác cao về toán học và vật lý. Hệ Mặt Trời bao gồm nhiều hành tinh chuyển động quanh Mặt Trời và ảnh hưởng lẫn nhau dưới tác động của lực hấp dẫn.

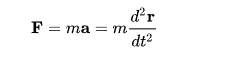
Các yếu tố cần phải xem xét khi tính toán quỹ đạo của hành tinh bao gồm:

* Lực hấp dẫn
* Khối lượng và kích thước của hành tinh
* Tác động từ các vật thể khác
* Điều kiện ban đầu
* Sự biến đổi theo thời gian

Tuy nhiên trong khuôn khổ của dự án, để đơn giản hóa việc tính quỹ đạo cho các hành tinh trong hệ Mặt Trời, chúng tả chỉ để cập tới các lực cơ bản nhất là lực hấp hẫn và một số lực tương tác trực tiếp tới chúng.

**Lực hấp dẫn**

Khoảng thời gian từ năm 1665 đến 1666, Issac Newton đã phát triển một số khái niệm liên quan đến chuyển động, lực hấp dẫn và phép vi phân. Trong đó, ông đã trình bày về các định luật về chuyển động và định luật vạn vật hấp dẫn. Định luật thứ hai trong ba định luật của ông nêu rõ: gia tốc của vật song song và tỉ lệ thuận với tổng lực tác dụng lên vật, cùng hướng với tổng hợp lực và tỉ lệ nghịch với khổi lượng của vật:

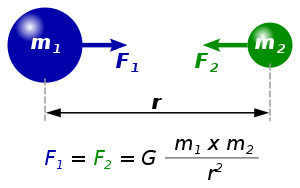


Trong đó F là lực, m là khối lượng của vật, a là gia tốc.

Định luật này được đơn giản hóa như sau: mọi khối lượng chất điểm đều hút mọi khối lượng chất điểm khác bằng một lực dọc theo giao điểm đường giao nhau của hai điểm. Lực tỉ lệ thuận với tích khối lượng hai vật và tỉ lệ nghịch với bình phương khoảng cách của hai vật

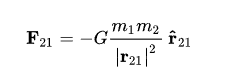
****

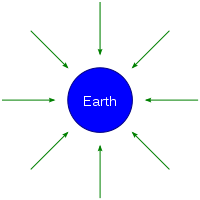
Trong đó F là lực hấp dẫn giữa hai vật, G là hằng số hấp dẫn, m1, m2 là khối lượng hai vật, r là khoảng cách giữa hai vật.



Dạng thức vector trong không gian:

Định luật vạn vật hấp dẫn có thể được viết dưới dạng phương trình vector để tính hướng của lực hấp dẫn cũng như độ lớn của nó:

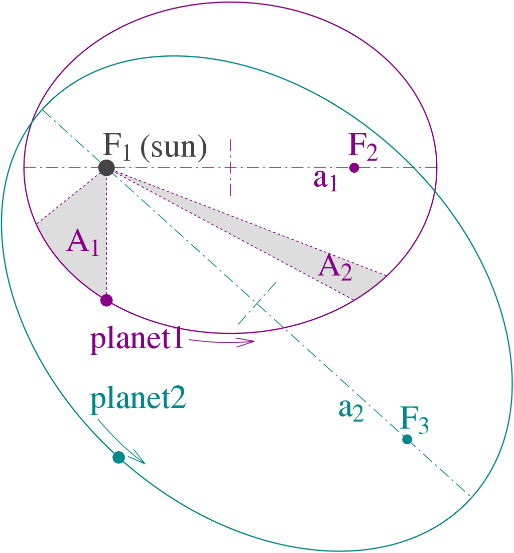




**Quỹ đạo Kepler**

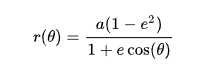
Một trong những đóng góp sớm nhất và quan trọng nhất của cơ học Newton là việc xác minh ba định luật về chuyển động hành tinh của Kepler. Trong thiên văn học, những định luật của Kepler về chuyển động thiên thể là ba định luật khoa học miêu tả chuyển động trên quỹ đạo của các vật thể, ban đầu chúng dùng để miêu tả chuyển động của các hành tinh trên quỹ đạo quay quanh Mặt Trời. Các định luật của Kepler bao gồm:

* Các hành tinh chuyển động quanh Mặt Trời theo quỹ đạo hình elip với Mặt Trời là một tiêu điểm.
* Đường nối một hành tinh với hệ Mặt Trời quét qua những diện tích bằng nhau trong khoảng thời gian bằng nhau.
* Bình phương chu kỳ quỹ đạo của một hành tinh tỉ lệ với lập phương bán kính trục lớn của quỹ đạo elip của hành tinh đó.



Trong cơ học vũ trụ, một **quỹ đạo Kepler** (hay **quỹ đạo Keplerian**, được đặt theo tên nhà thiên văn học người Đức Johannes Kepler) là chuyển động của một vật thể này so với vật thể khác, dưới dạng **ellipse**, **parabol** hoặc **hyperbol**, tạo thành một **mặt phẳng quỹ đạo hai chiều** trong không gian ba chiều. Quỹ đạo Kepler cũng có thể hình thành một **đường thẳng**.

Nó chỉ xét đến lực hấp dẫn giống như điểm của hai vật thể, bỏ qua các **nhiễu loạn** do tương tác hấp dẫn với các vật thể khác, **lực cản khí quyển**, **áp lực bức xạ mặt trời**, một vật thể trung tâm không hình cầu, v.v. Do đó, nó được cho là một giải pháp cho một trường hợp đặc biệt của **bài toán hai vật**, được gọi là **bài toán Kepler**. Là một lý thuyết trong cơ học cổ điển, nó cũng không tính đến các tác động của **thuyết tương đối tổng quát**. Về mặt toán học, khoảng cách giữa một vật thể đang quay trên quỹ đạo Kepler có thể được thể hiện bằng công thức.



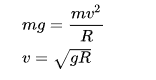
Trong đó: r là khoảng cách, a là trục chính bán kính, e là độ lệch, theta là goc giữa vị trí hiện tại của vật thể quay và vị trí trong quỹ đạo mà nó gần nhất với tiêu điểm

**Tốc độ vũ trụ**

Vận tốc vũ trụ hay tốc độ vũ trụ hay tốc độ thoát ly cũng như là vận tốc thoát ly là tốc độ của một vật cần có để nó chuyển động theo quỹ đạo tròn gần bề mặt của một vật thể khác hoặc thoát ra khỏi lực hấp dẫn của vật thể khác. Có bốn cấp độ của tốc độ vũ trụ như sau:

* Tốc độ vũ trụ cấp 1 nếu vận tốc đủ lớn để trở thành vệ tinh của một hành tinh nào đó, với Trái Đất là khoảng 7,9 km/s.
* Tốc độ vũ trụ cấp 2 nếu đủ lớn để trở thành vật thể bay xung quanh Mặt Trời: khoảng 11,2 km/s.
* Tốc độ vũ trụ cấp 3 nếu đủ lớn để thoát khỏi lực hấp dẫn của Mặt Trời: khoảng 16,6 km/s.
* Tốc độ vũ trụ cấp 4 nếu đủ lớn để thoát ra khỏi lực hấp dẫn của dải Ngân Hà: 525 km/s.

Trong giả lập quỹ đạo của hành tinh, việc tính tốc độ vũ trụ là cần thiết để giúp cho hành tinh có quỹ đạo tròn quanh Mặt Trời hay nói chính xác hơn là hành tinh quanh quanh thiên thể chủ. Lực mà tốc độ vũ trụ ra giúp cho vệ tinh đó tiếp tục chuyển động quanh quỹ đạo mà không bị rơi vào bề mặt của hành tinh, nó là sự cân bằng giữa lực hấp dẫn và lực quán tính ly tâm do vật chuyển động tròn có được. Từ điều kiện lực hấp dẫn bằng lực quán tính ly tâm có thể suy ra



Trong đó v là tốc độ bay trên quỹ đạo của vật thể, m là khối lượng vật thể, g là gia tốc trọng trường, R là bán kính thiên thể chủ.

Chú ý: gia tốc trọng trường g có thể được tính theo công thức định luật vạn vật hấp dẫn của Newton:



# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong chương này, đồ án sẽ tập trung vào việc trình bày phân tích thiết kế cho toàn bộ hệ thống và phân tích thiết kế chi tiết cho từng module có mặt trong trong hệ thống.

## Phân tích hệ thống

### Tên hệ thống

Ứng dụng Android mô phỏng hệ mặt trời.

### Mục tiêu của hệ thống:

Ứng dụng được xây dựng với mục tiêu cung cấp những kiến thức căn bản, cung cấp hệ thống hình ảnh và modle cho những người quan tâm và yêu thích với thiên văn học. Hệ thống giúp người dùng có cái nhìn trực quan hơn về hệ Mặt Trời cũng như các hành tinh trong hệ Mặt Trời.

### Tác nhân của hệ thống

Tác nhân chính của hệ thống là người dùng hệ thống.

### Yêu cầu của hệ thống:

Hệ thông đảm bảo chất lượng của những chức năng cung cấp tới người dùng, tương thích được với nhiều phiên bản máy của người dùng, tối ưu được dung lượng lưu trữ, hoạt động trơn tru trên các thế hệ máy cũ.

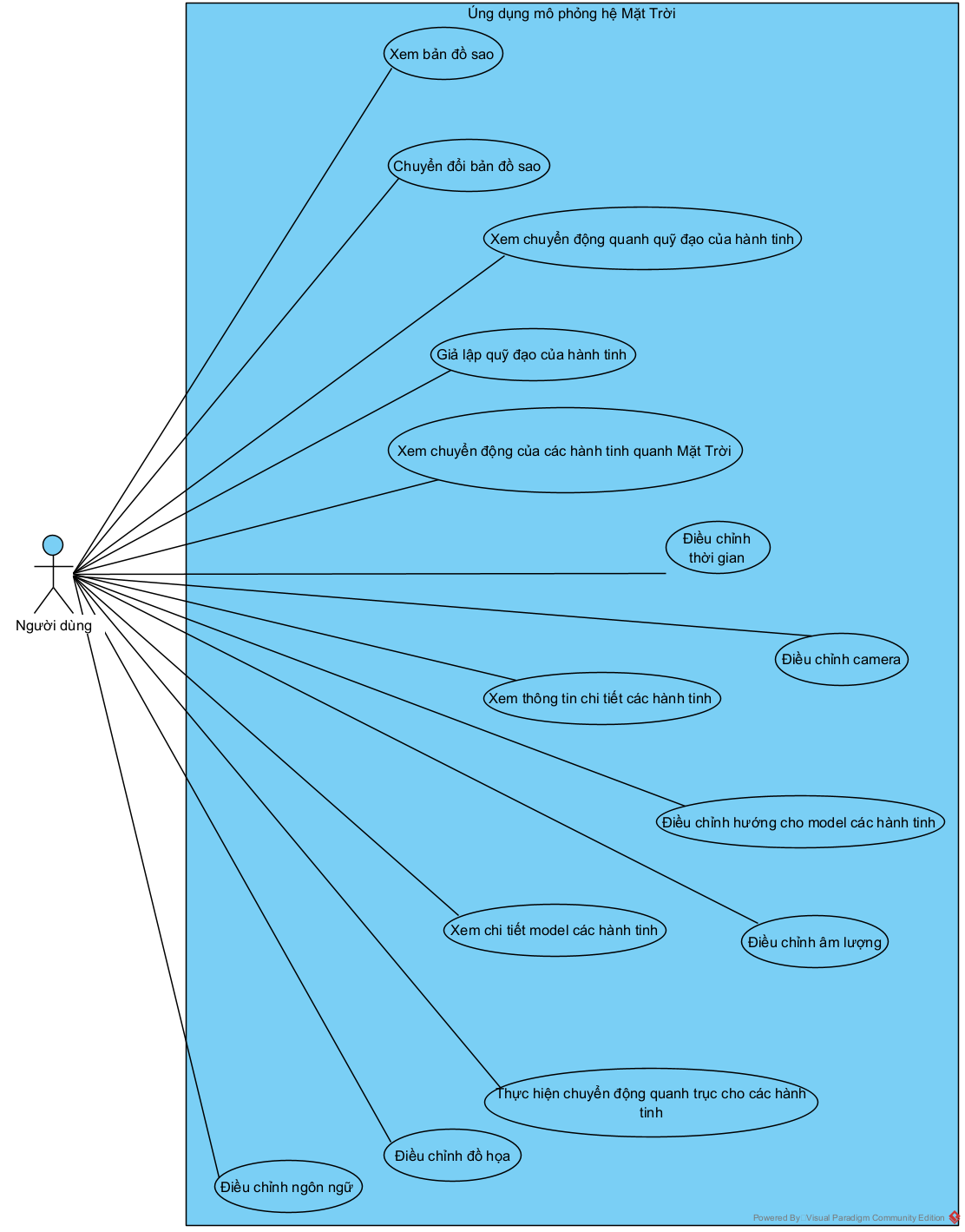
### Phạm vi của hệ thống

Người dùng của hệ thống có thể:

* Xem bản đồ sao.
* Chuyển đổi giữa các bản đồ.
* Xem chuyển động xoay quanh quỹ đạo của hành tinh.
* Thực hiện giả lập chuyển động quanh quỹ đạo của hành tinh.
* Điều chỉnh, giả lập thời gian trong chuyển động quỹ đạo.
* Xem chuyển động của các hành tinh trong hệ Mặt Trời.
* Điều chỉnh vị trí, góc quay của camera.
* Hiển thị thông tin của các hành tinh trong hệ Mặt Trời.
* Xem chi tiết model của các hành tinh trong hệ Mặt Trời.
* Thực hiện chuyển động quay quanh trục cho model của các hành tinh.
* Điều chỉnh hướng cho các model
* Điều chỉnh âm lượng cho toàn bộ hệ thống.
* Điều chỉnh đồ họa cho toàn bộ hệ thống.
* Điều chỉnh ngôn ngữ cho toàn bộ hệ thống.
* Chạy test: thực thi chương trình với đầu vào mong muốn

## Thiết kế hệ thống

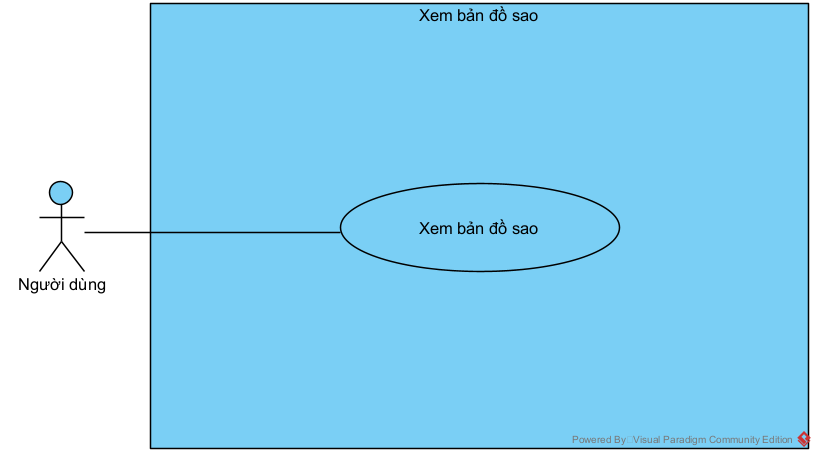
### Usecase tổng quan



## Phân rã các module

### Module xem bản đồ sao

#### Usecase chi tiết

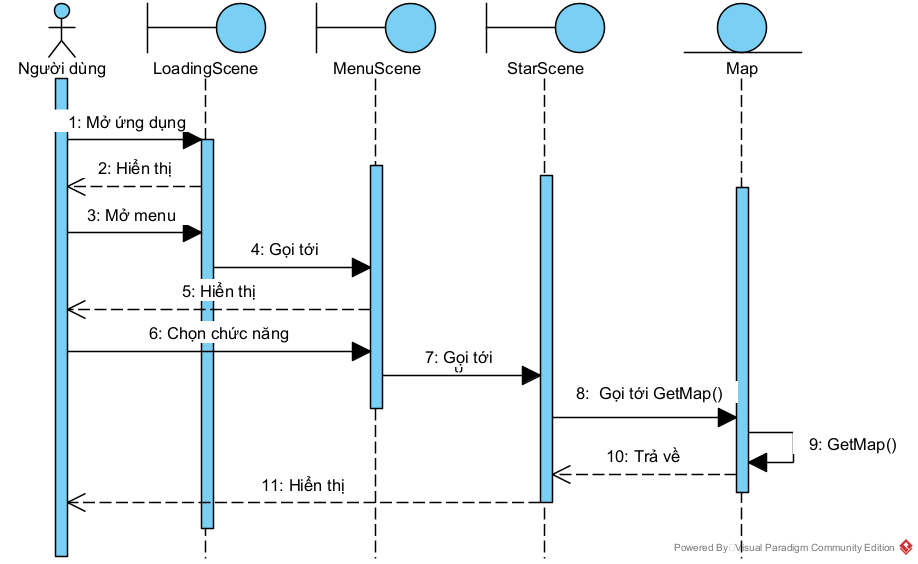


#### Kịch bản

Bảng . Kịch bản xem bản đồ sao

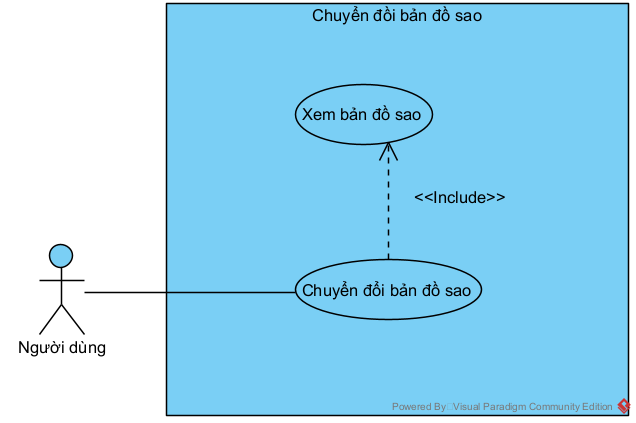
|  |  |
| --- | --- |
| Tên usecase | Xem bản đồ sao |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng lựa chọn chức năng xem bản đồ sao |
| Hậu điều kiện | Người dùng có thể sử dụng bản đồ sao |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem bản đồ sao.  6. Giao diện xem bản đồ sao hiện lên với bản đồ sao dưới dạng 3D cube map, button chuyển đổi bản đồ sao.  7.Người dùng dùng thao tác vuốt để di chuyển tầm nhìn và góc nhìn cho phù hợp để xem được các chòm sao. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module chuyển đồi bản đồ sao

#### Usecase chi tiết

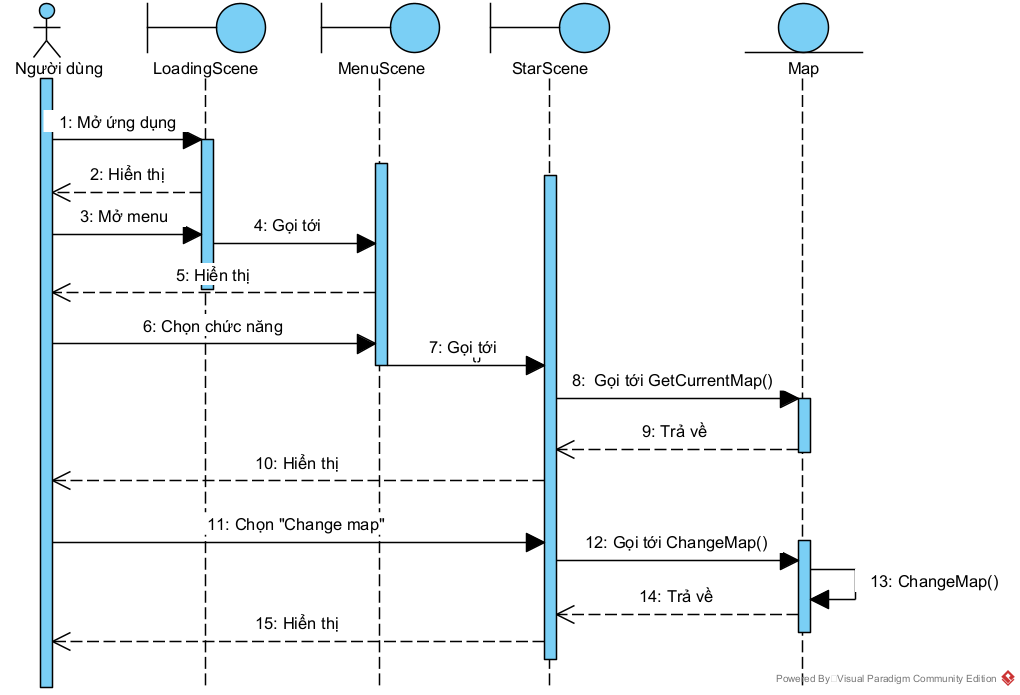


#### Kịch bản

Bảng . Kịch bản module chuyển đổi bản đồ sao

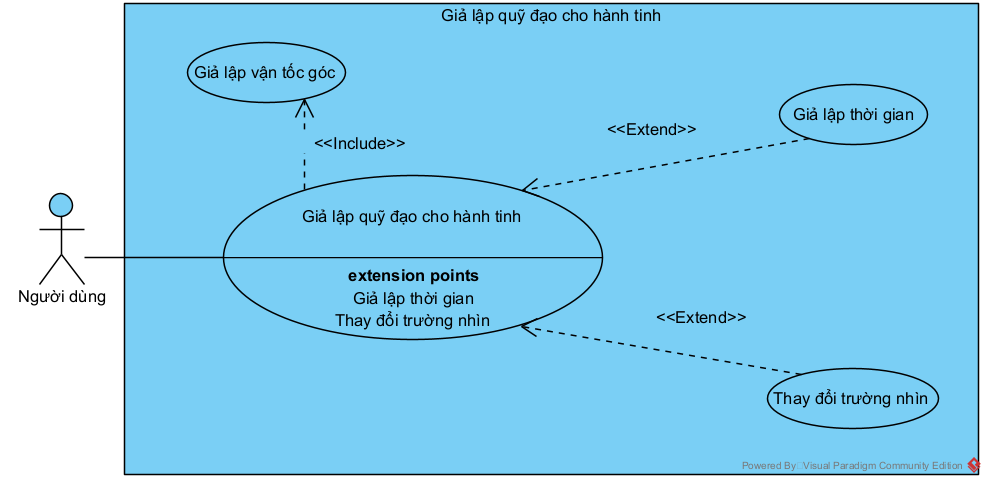
|  |  |
| --- | --- |
| Tên usecase | Đăng nhập |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sủ dụng chức năng xem bản đồ sao |
| Hậu điều kiện | Người dùng chuyển đổi bản đồ sao thành công |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem bản đồ sao.  6. Giao diện xem bản đồ sao hiện lên với bản đồ sao dưới dạng 3D cube map, button chuyển đổi bản đồ sao.  7. Người dùng chọn vào button thay đổi bản đồ.  8. Hệ thống hiển thị lại bản đồ mới thay thế cho bản đồ trước đó. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module giả lập quỹ đạo cho hành tinh

#### Usecase chi tiết

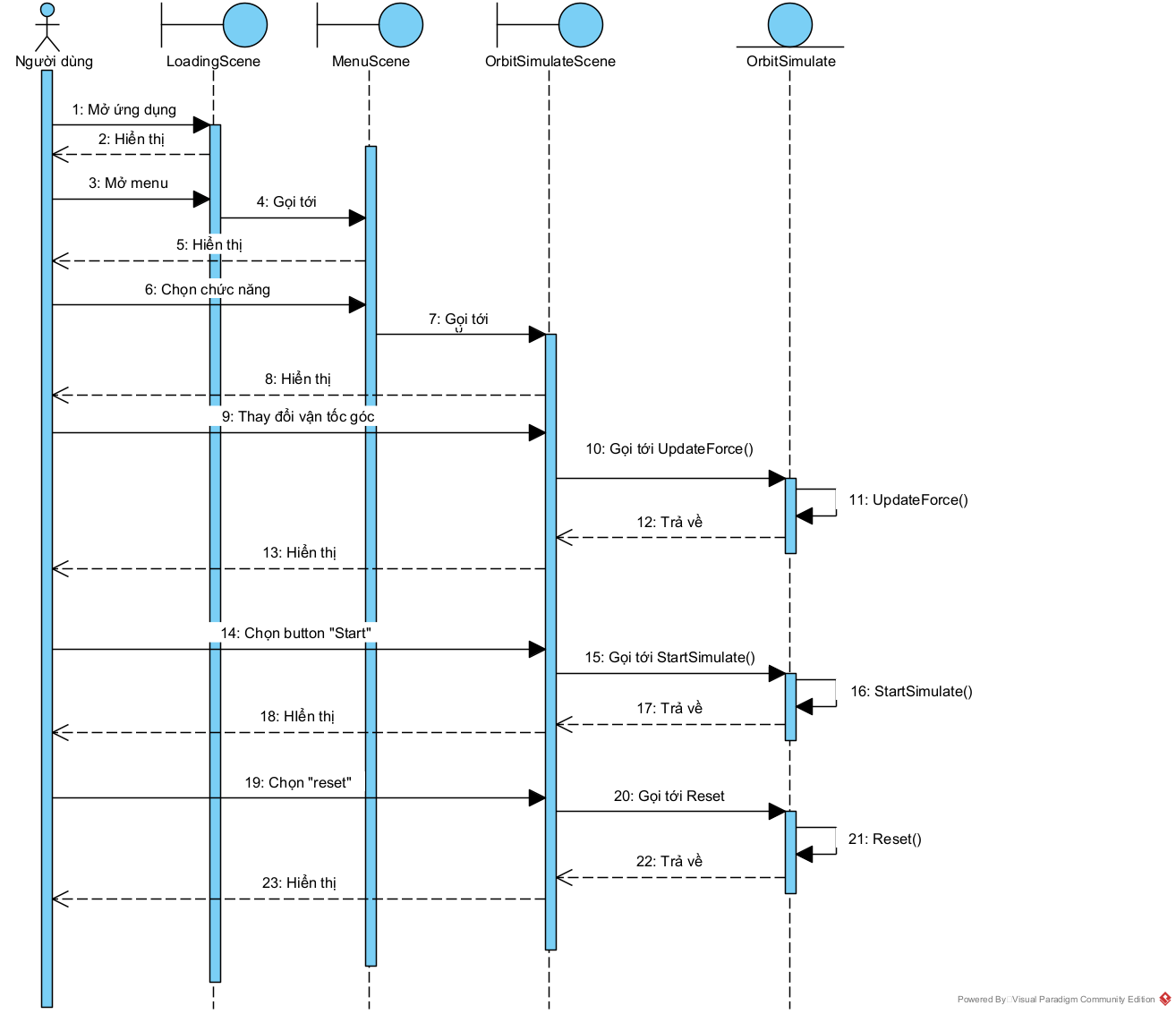


#### Kịch bản

Bảng . Kịch bản module giả lập quỹ đạo cho hành tinh

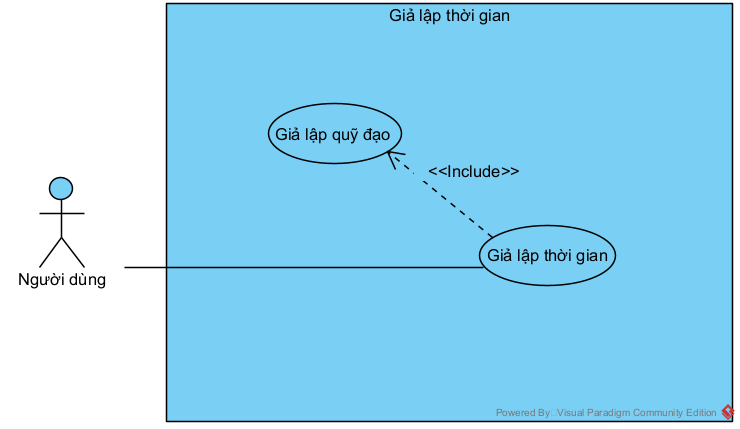
|  |  |
| --- | --- |
| Tên usecase | Giả lập quỹ đạo cho hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng lựa chọn chức năng giả lập quỹ đạo cho hành tinh |
| Hậu điều kiện | Quỹ đạo của hành tinh được thay đổi |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo thả để thay đổi quỹ đạo cho hành tinh.  8. Hệ thống liên tục hiển thị quỹ đạo mới cho hành tinh dựa trên hướng kéo và khoảng cách kéo mà người dùng tác độn lên hành tinh.  9. Người dùng kết thúc thao tác kéo thả để cố định quỹ đạo cho hành tinh.  10. Hệ thống hiển thị quỹ đao cho hành tinh dưới dạng lineRenderer.  11. Người dùng nhấn vào button bắt đầu để thực hiện chuyển độn quay quanh quỹ đạo cho hành tinh.  12. Hành tinh di chuyển trên quỹ đạo mới được tạo  13. Người dùng nhấn button đặt lại để đưa hành tinh về trạng thái ban đầu.  14. Thực hiện lặp lại các bước 8 đến 14 để điều chỉnh lại quỹ đạo cho hành tinh. |
| Ngoại lệ | 8. Lực mà người dùng tác động lên hành tinh quá lớn.  8.1 Hệ thống hiển thị lực tràn ra ngoài tầm nhìn khả dụng.  8.2 Người dùng nhấn reset.  8.3 Hệ thống thực hiện reset lực.  8.4 Người dùng tạo lại lực mới lên hành tinh. |

#### Biểu đồ tuần tự



### Module giả lập thời gian

#### Usecase chi tiết

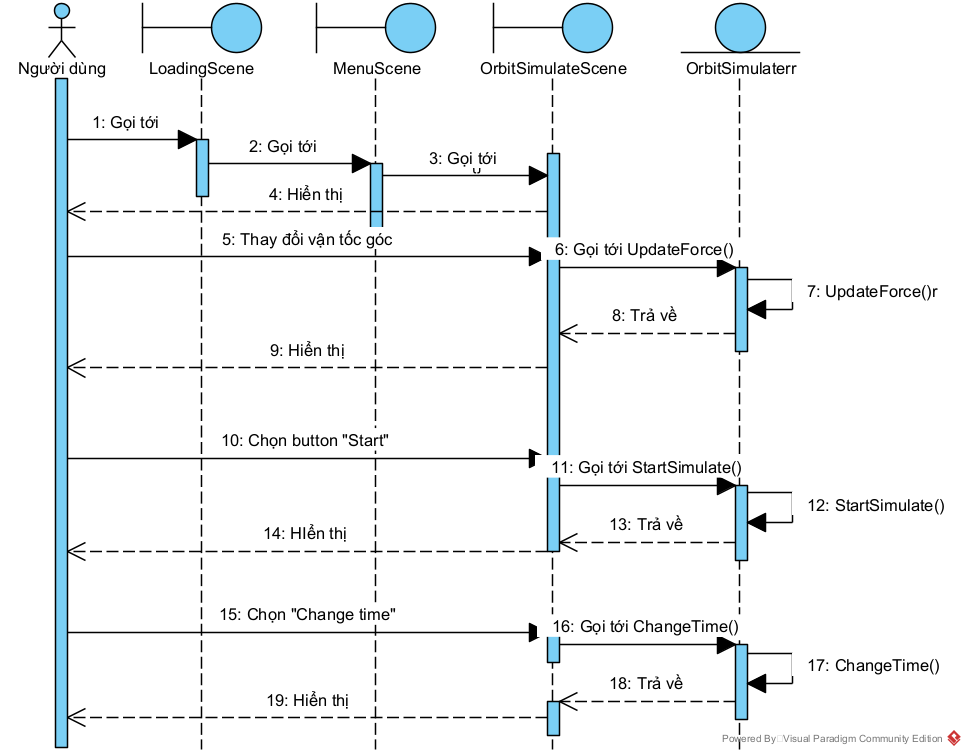


#### Kịch bản

Bảng . Kịch bản giả lập thời gian

|  |  |
| --- | --- |
| Tên usecase | Giả lập thời gian |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng giả lập quỹ đạo |
| Hậu điều kiện | Thay đổi thành công thời gian tương đối trong hệ thống |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”..  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo thả để thay đổi quỹ đạo cho hành tinh.  8. Hệ thống liên tục hiển thị quỹ đạo mới cho hành tinh dựa trên hướng kéo và khoảng cách kéo mà người dùng tác độn lên hành tinh.  9. Người dùng kết thúc thao tác kéo thả để cố định quỹ đạo cho hành tinh.  10. Hệ thống hiển thị quỹ đao cho hành tinh dưới dạng lineRenderer.  11. Người dùng nhấn vào button bắt đầu để thực hiện chuyển độn quay quanh quỹ đạo cho hành tinh.  12. Hành tinh di chuyển trên quỹ đạo mới được tạo  13. Người dùng nhấn vào button thay đổi thời gian cho tới khi đáp ứng được thời gian mong muốn.  14. Hành tinh chuyển động trên quỹ đạo với tốc đô tương ứng với gia tốc thời gian. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module thay đổi trường nhìn

#### Usecase chi tiết

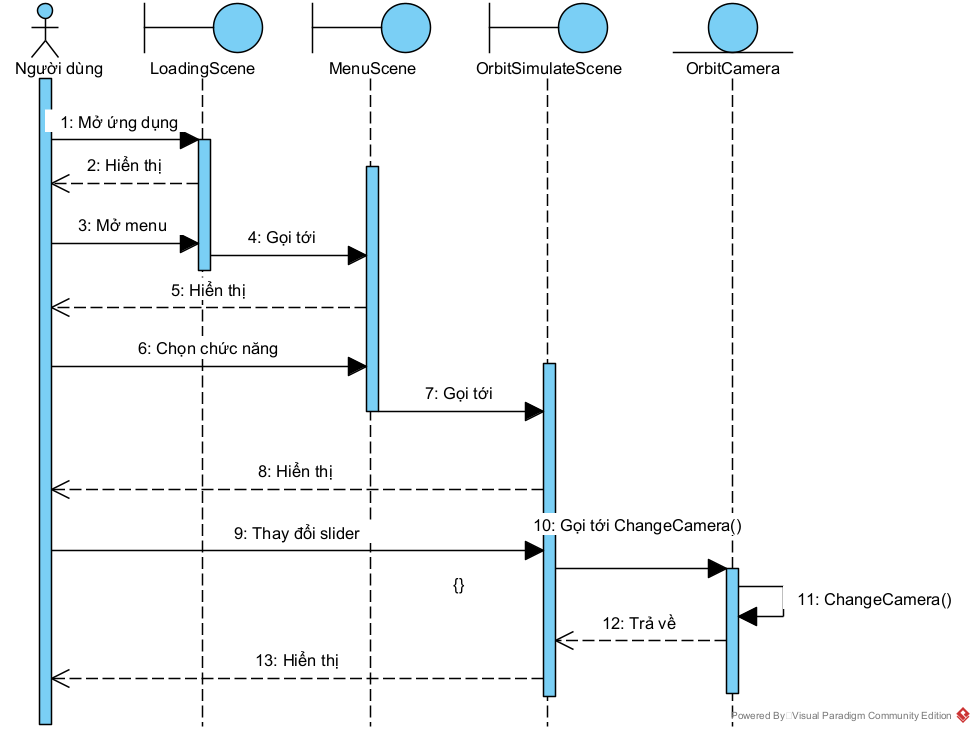


#### Kịch bản

Bảng . Kịch bản module thay đổi trường nhìn

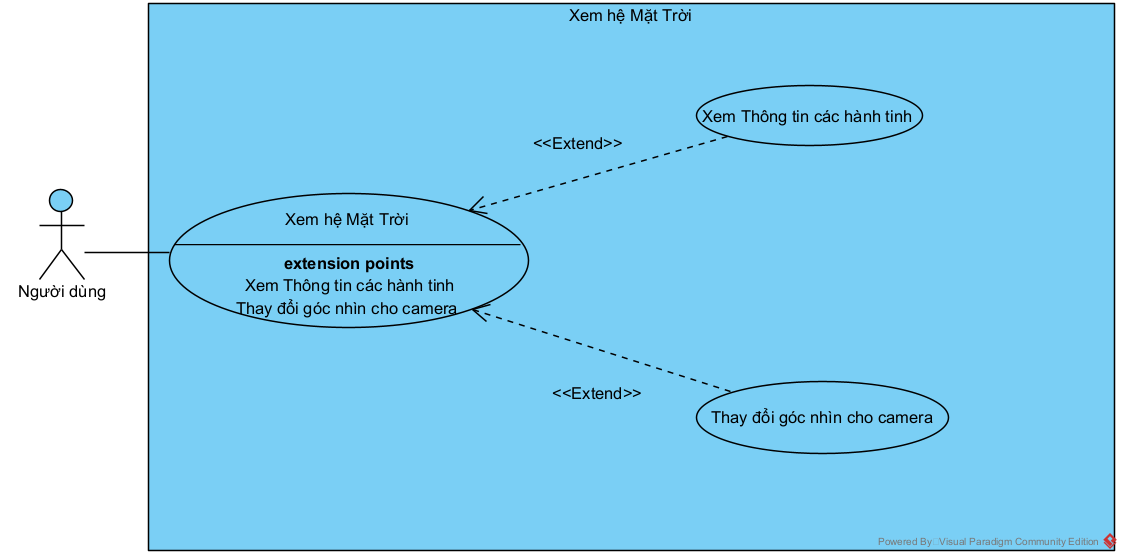
|  |  |
| --- | --- |
| Tên usecase | Thay đổi trường nhìn |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng mô phỏng quỹ đạo |
| Hậu điều kiện | Trường nhìn của camera được thay đổi |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo để thay đổi giá trị của slider “Fiew Of View” cho đến khi có trường nhìn phù hợp.  8. Hệ thống thay đổi trường nhìn liên tục theo giá trị của slider mà người dùng chọn. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module xem hệ Mặt Trời

#### Usecase chi tiết

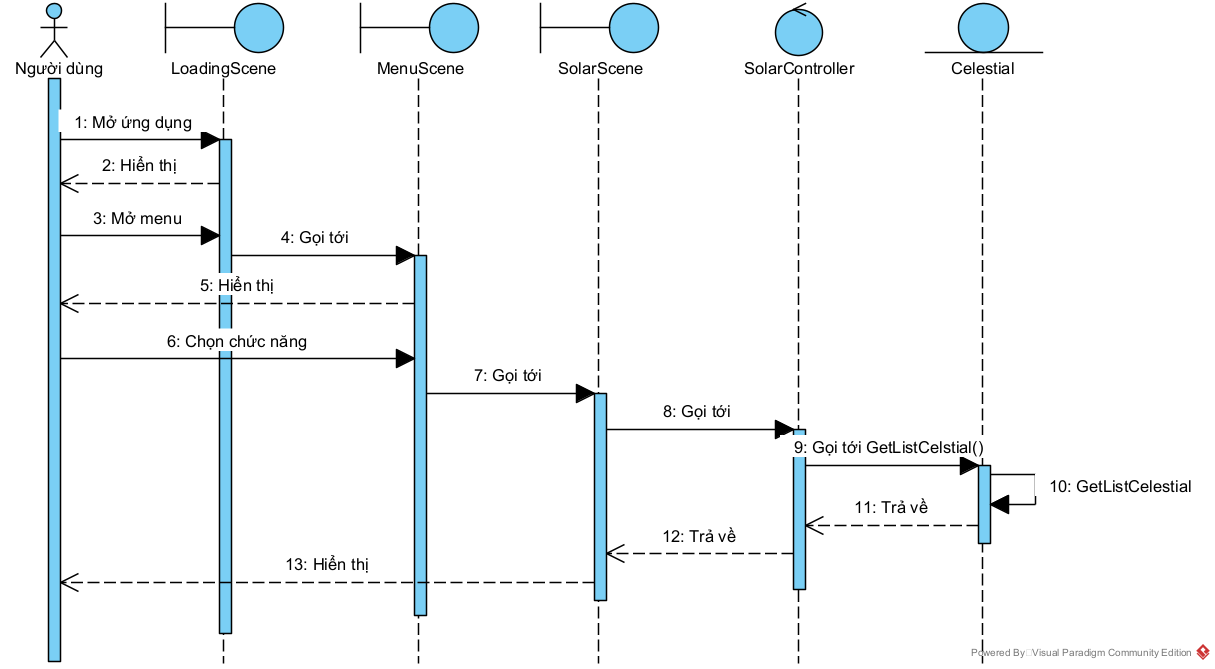


#### Kịch bản

Bảng . Kịch bản module xem hệ Mặt Trời

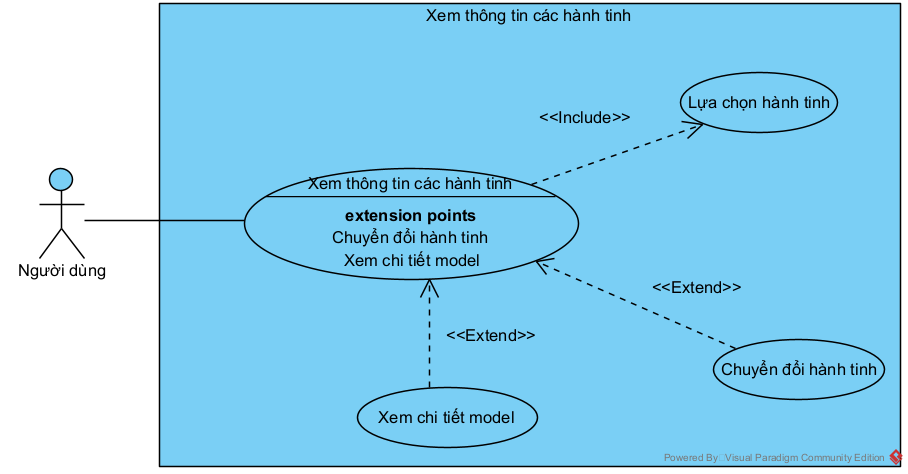
|  |  |
| --- | --- |
| Tên usecase | Xem hệ Mặt Trời |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem hệ Mặt Trời |
| Hậu điều kiện | Người dùng xem được các hành tinh trong hệ Mặt Trời |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module xem thông tin các hành tinh

#### Usecase chi tiết

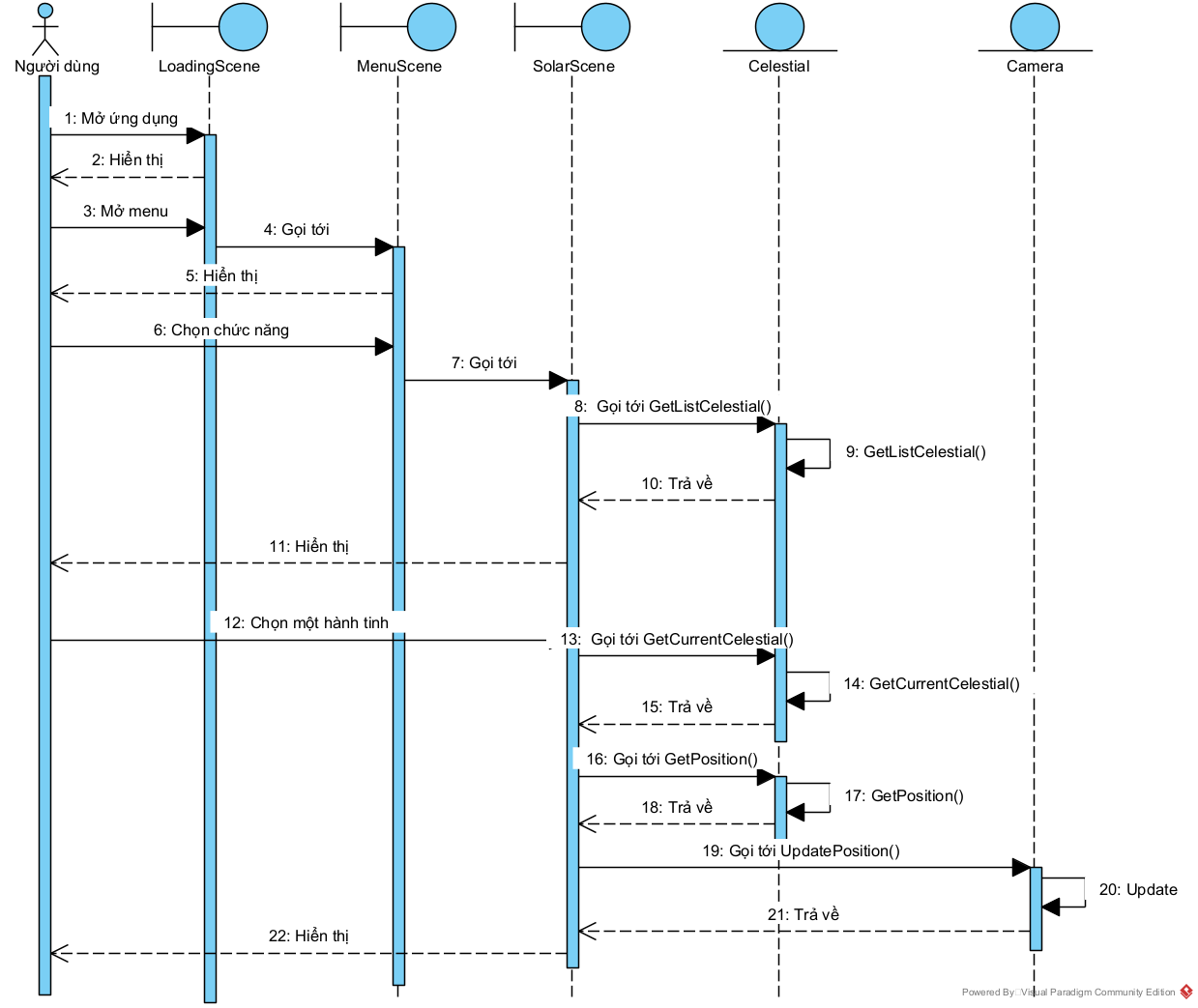


#### Kịch bản

Bảng . Kịch bản module xem thông tin các hành tinh

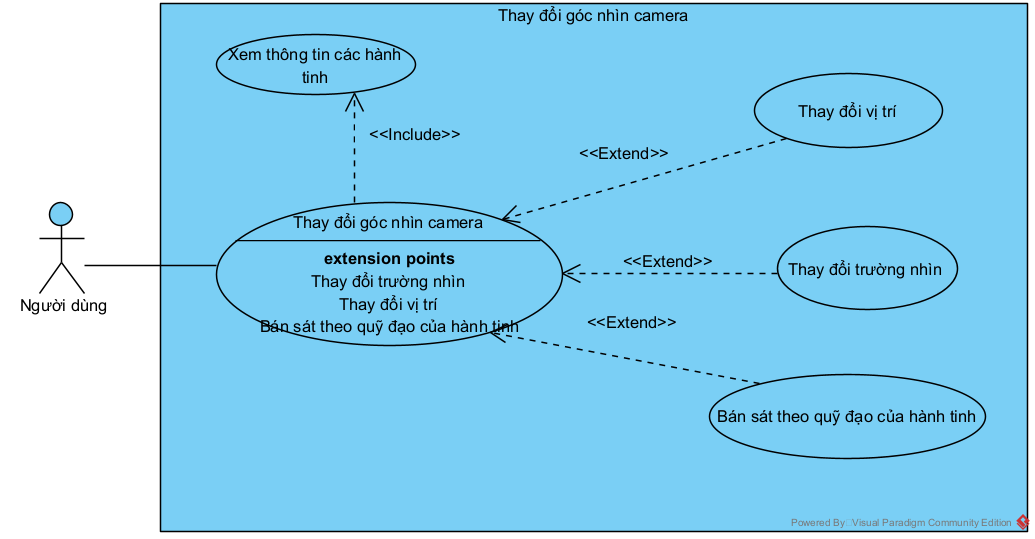
|  |  |
| --- | --- |
| Tên usecase | Xem thông tin các hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem hệ Mặt Trời |
| Hậu điều kiện | Người dùng xem được thông tin của các hành tinh |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời.  7. Người dùng chọn button tên một hành tinh đang được hiển thị.  8. Hệ thống tự động di chuyển camera đến hành tinh đó và di chuyển bám theo chuyển động của hành tinh đồng thời hiển thị lên màn hình các thông tin cơ bản của hành tinh.  9. Người dùng chọn thoát để trở về trạng thái ban đầu.  10. Lặp lại các bước từ 7 đến 9 để thay đổi thông tin của các hành tinh. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module thay đổi góc nhìn camera

#### Usecase chi tiết

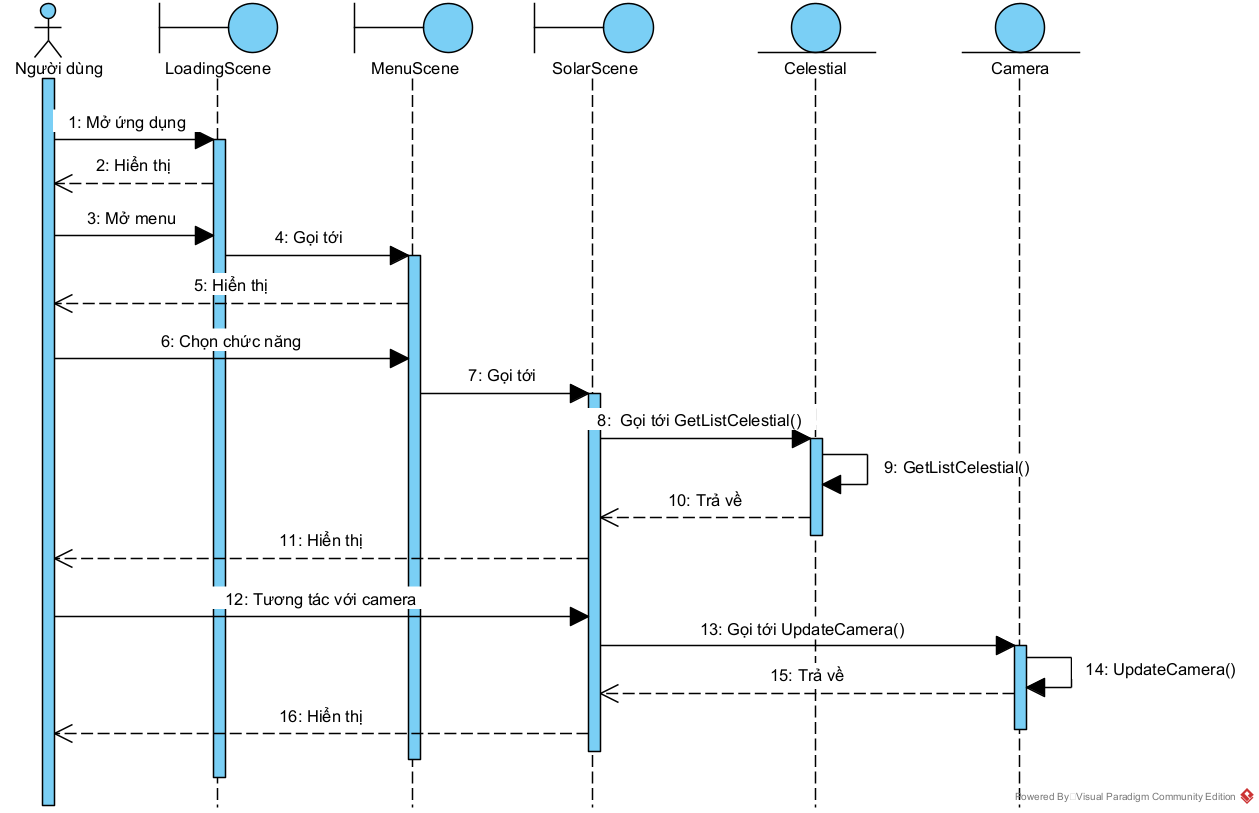


#### Kịch bản

Bảng . Kịch bản module

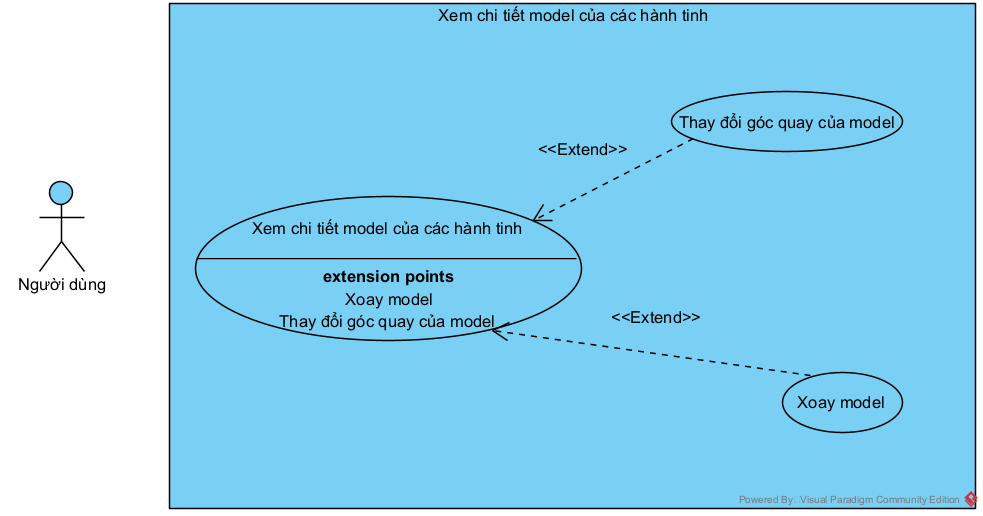
|  |  |
| --- | --- |
| Tên usecase | Xem xếp hạng thành viên |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng xem được xếp hạng người dùng |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời.  7. Người dùng sử dụng thao tác zoom in, zoom out để thay đổi khoảng cách tương đối của camera với Mặt Trời.  8. Hệ thống thực hiện di chuyển camera tương ứng với độ zoom mà người dùng thực hiện.  9. Người dùng thực hiện thao tác vuốt để thay đổi hướng nhìn của camera.  10. Hệ thống thực hiện thay đổi góc nhìn của camera dựa trên khoảng cách và hướng vuốt của người dùng. |
| Ngoại lệ | 8. Người dùng zoom quá trường nhìn khả dụng.  8.1 Hệ thống tự động ngừng thao tác zoom của người dùng.  8.2 Người dùng thực hiện lại thao tác zoom |

#### Biểu đồ tuần tự



### Module xem chi tiết model

#### Usecase chi tiết

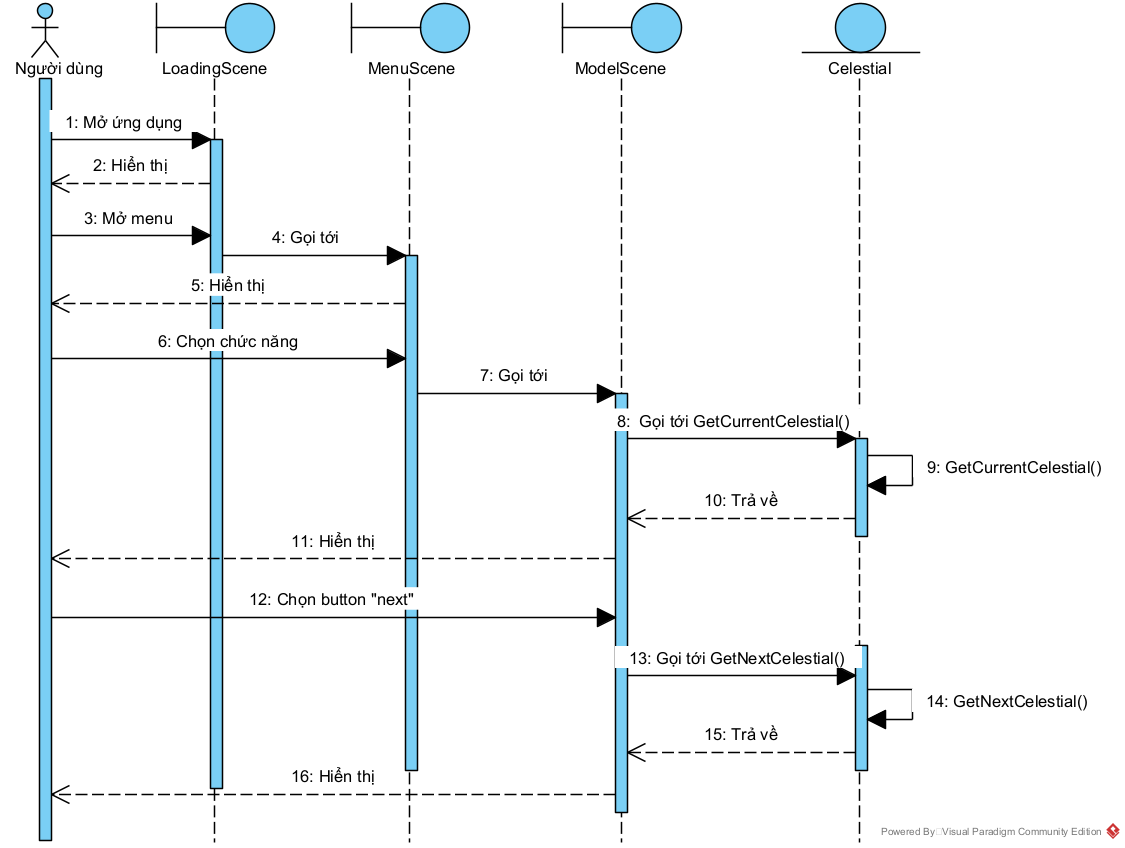


#### Kịch bản

Bảng . Kịch bản module xem chi tiết model

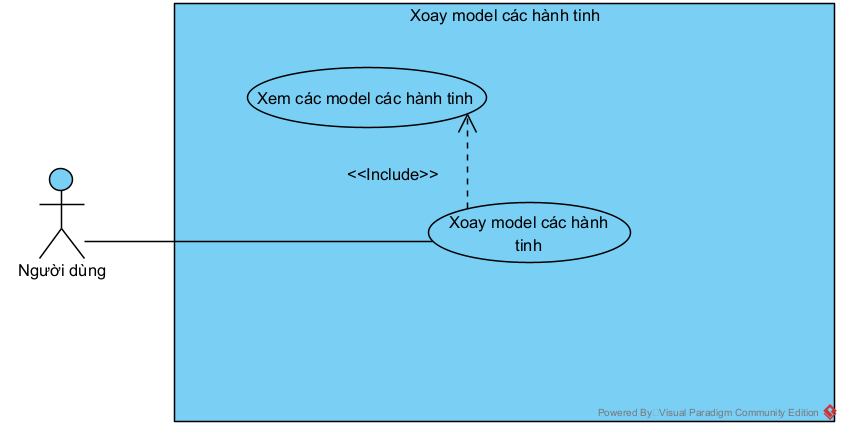
|  |  |
| --- | --- |
| Tên usecase | Xem chi tiết model |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem chi tiết model |
| Hậu điều kiện | Người dùng xem được model của các hành tinh trong hệ Mặt Trời |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem chi tiết model.  6. Hệ thống hiển thị giao diện xem chi tiết model với model mặc định là model của Mặt Trời.  7. Người dùng chọn các buttton trái phải để thay đổi giữa các model.  8. Hệ thống thực hiện thay đổi các model tương ứng dựa trên danh sách các hành tinh được cài đặt trước. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



### Module xoay model các hành tinh

#### Usecase chi tiết

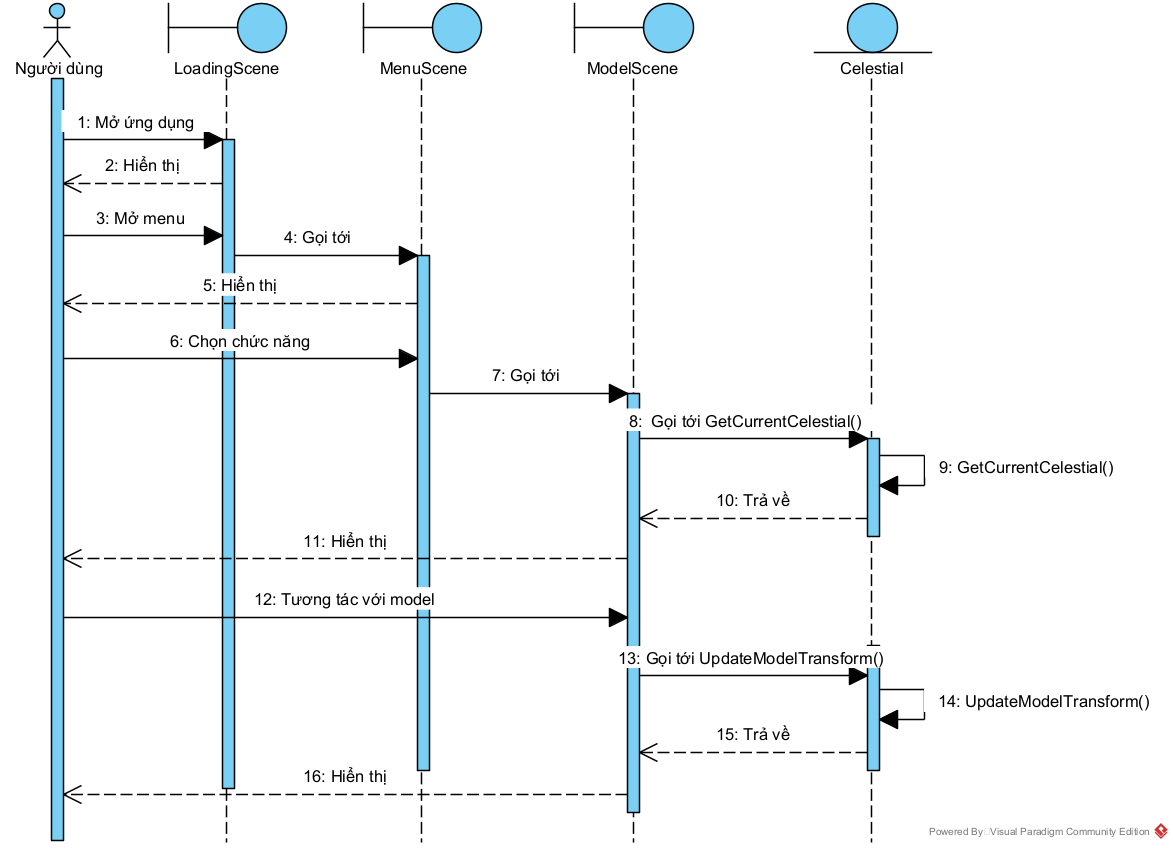


#### Kịch bản

Bảng . Kịch bản module xoay model các hành tinh

|  |  |
| --- | --- |
| Tên usecase | Xoay model các hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem model các hành tinh |
| Hậu điều kiện | Model của các hành tinh xoay theo hướng người dùng chọn |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem chi tiết model.  6. Hệ thống hiển thị giao diện xem chi tiết model với model mặc định là model của Mặt Trời.  7. người dùng thực hiện thao tác kéo thả trên model để quay model theo hướng chỉ định.  8. Hệ thống thực hiện xoay model theo hướng người dùng đã chỉ đinh. |
| Ngoại lệ |  |

#### Biểu đồ tuần tự



### Module thực hiện chuyển động quay quanh trục cho các hành tinh

#### Usecase chi tiết

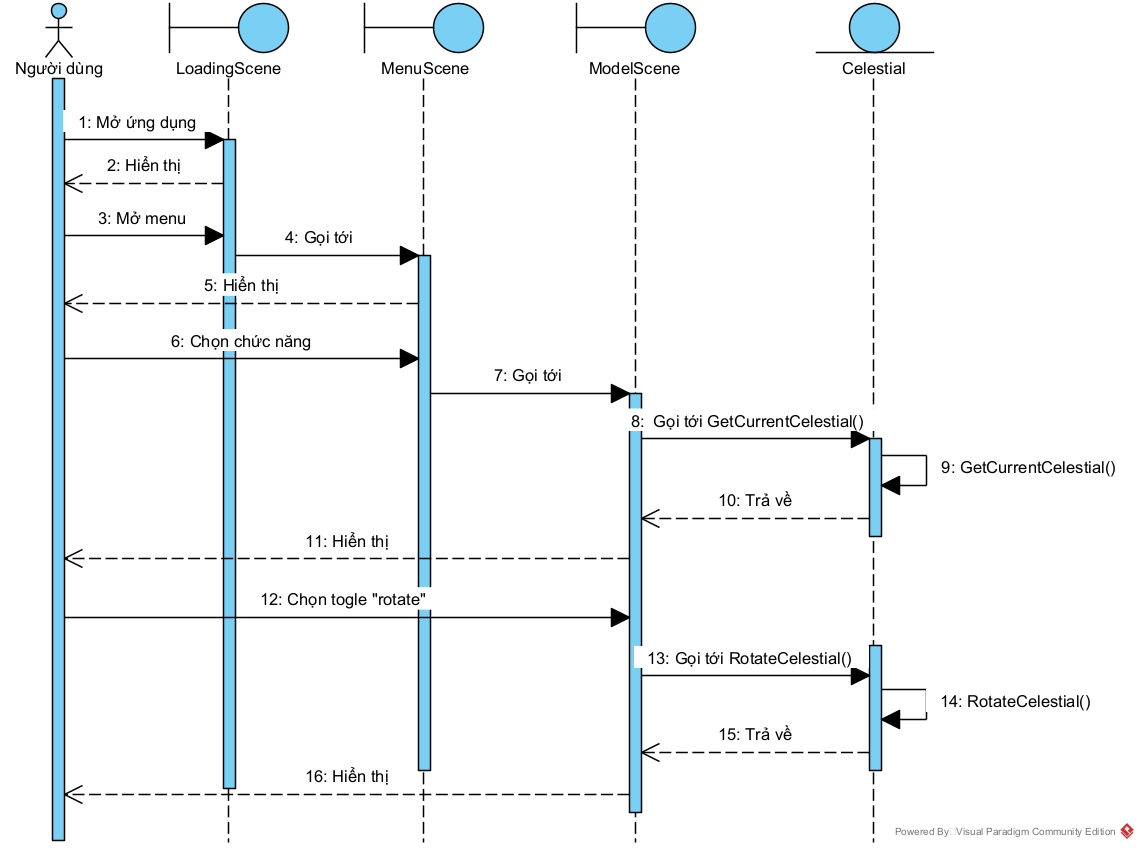


#### Kịch bản

Bảng . Kịch bản chuyển động quay quanh trục cho các hành tinh.

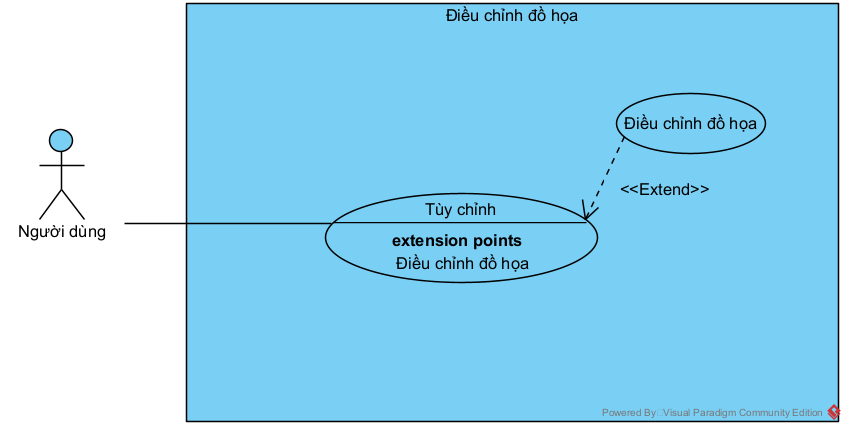
|  |  |
| --- | --- |
| Tên usecase | Thực hiện chuyển động quay quanh trục cho các hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem model của các hành tinh |
| Hậu điều kiện | Model của hành tinh thực hiện chuyển động quay quanh trục |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem chi tiết model.  6. Hệ thống hiển thị giao diện xem chi tiết model với model mặc định là model của Mặt Trời.  7. Người dùng chọn vào toggle “is rotate ”.  8. Hệ thống thực hiện chuyển động quay quanh trục cho hành tinh hiện tại.  9. Người dùng chọn chuyển đổi hành tinh và thực hiện lại các bước 7 đến 9 đển thực hiện tương tự với các hành tinh khác. |
| Ngoại lệ |  |

#### Biểu đồ tuần tự



### Module điều chỉnh đồ họa

#### Usecase chi tiết

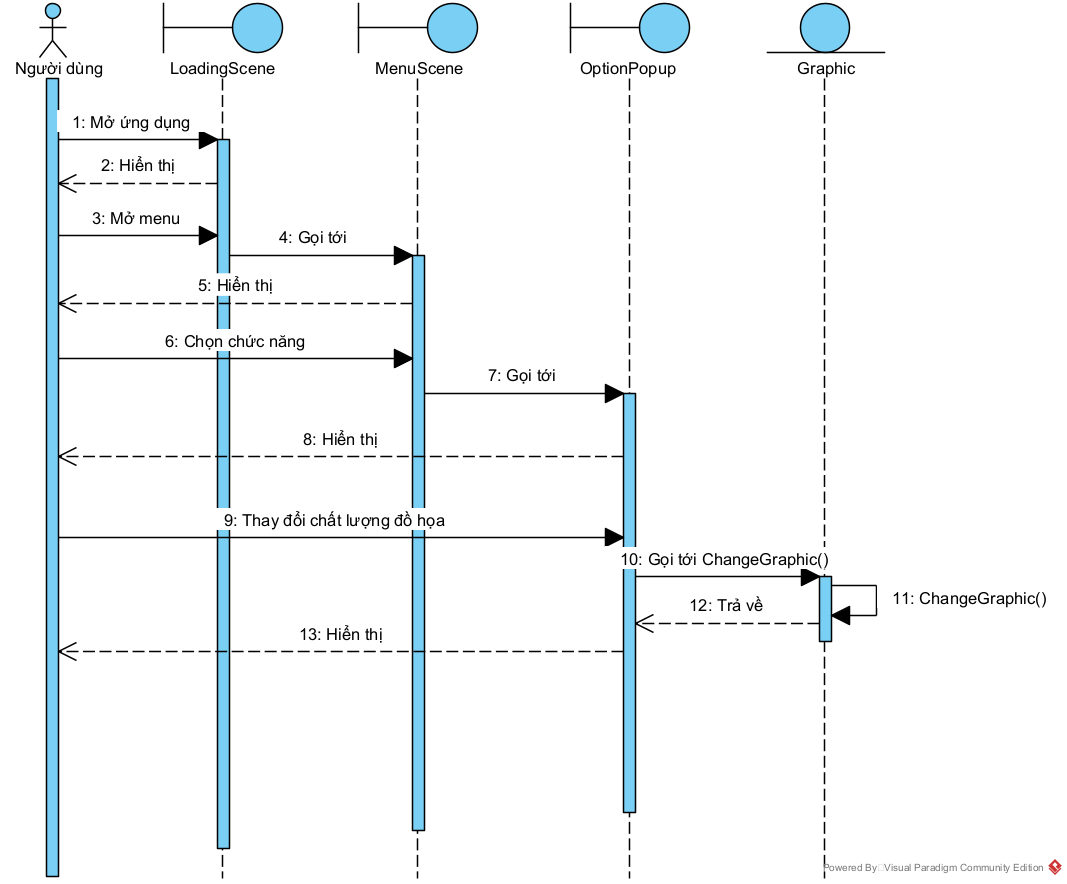


#### Kịch bản

Bảng . Kịch bản module điều chỉnh đồ họa

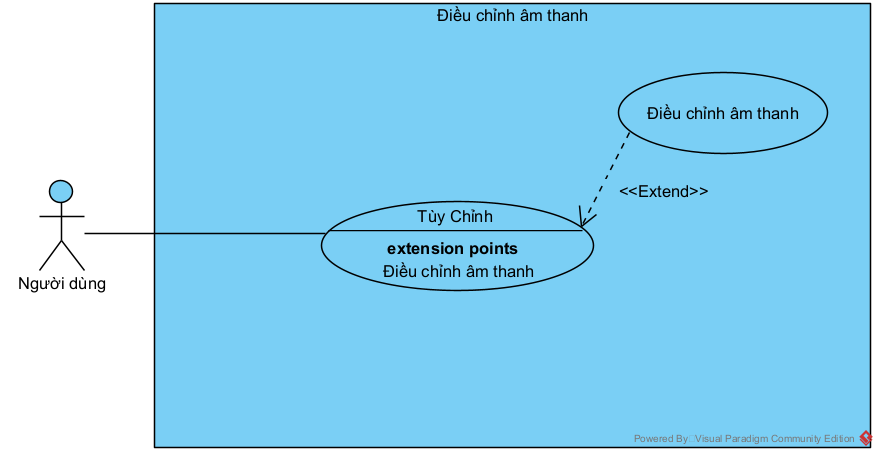
|  |  |
| --- | --- |
| Tên usecase | Điều chỉnh đồ họa |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng tùy chọn |
| Hậu điều kiện | Đồ họa được điều chỉnh tướng ứng với mức mà người dùng chọn |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn vào chức năng điều chỉnh.  6. Giao diện điều chỉnh hiện lên với 3 chức năng chính là điều chỉnh âm thanh, điều chỉnh đồ họa và điều chỉnh ngôn ngữ.  7. Người dùng chọn vào dropdown tùy chỉnh hình ảnh và chọn chất lượng hình ảnh.  8. Hệ thống thay đổi chất lượng ảnh tùy theo mốc mà người dùng chọn. |
| Ngoại lệ |  |

#### Biểu đồ tuần tự



### Module điều chỉnh âm lượng

#### Usecase chi tiết

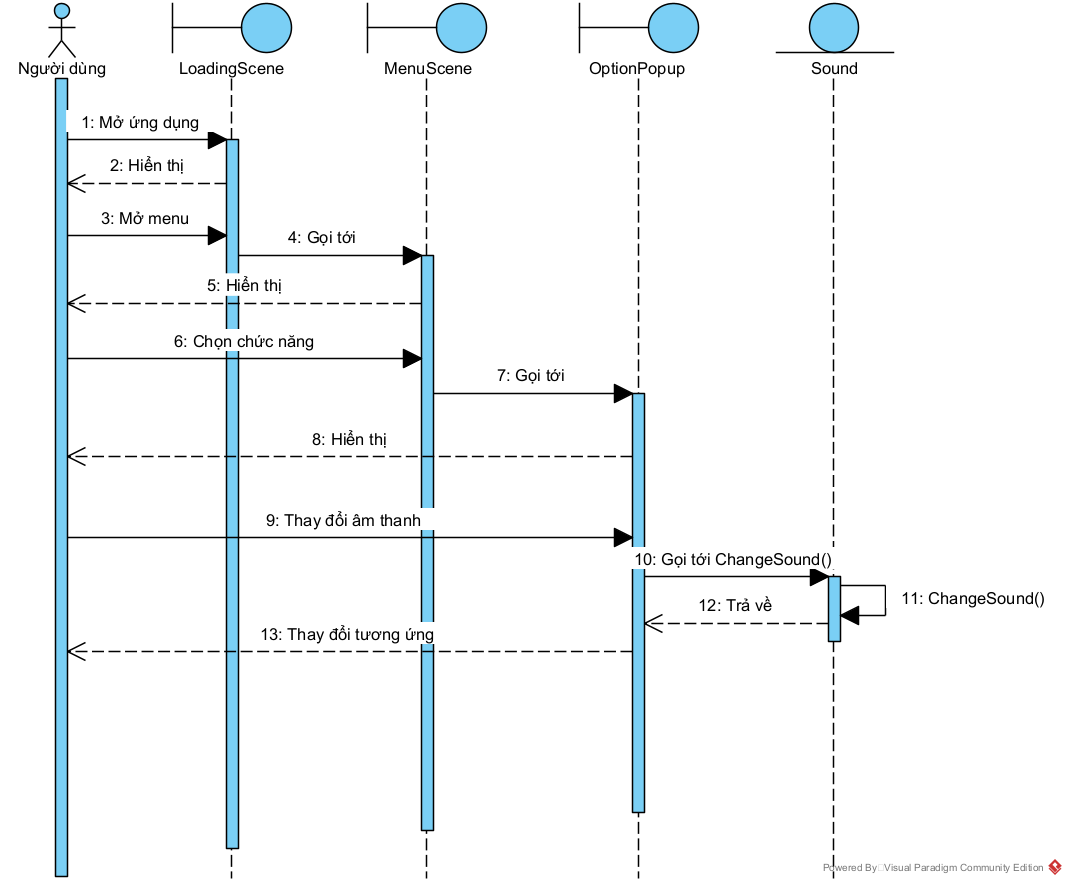


#### Kịch bản

Bảng . Kịch bản module điều chỉnh âm lượng

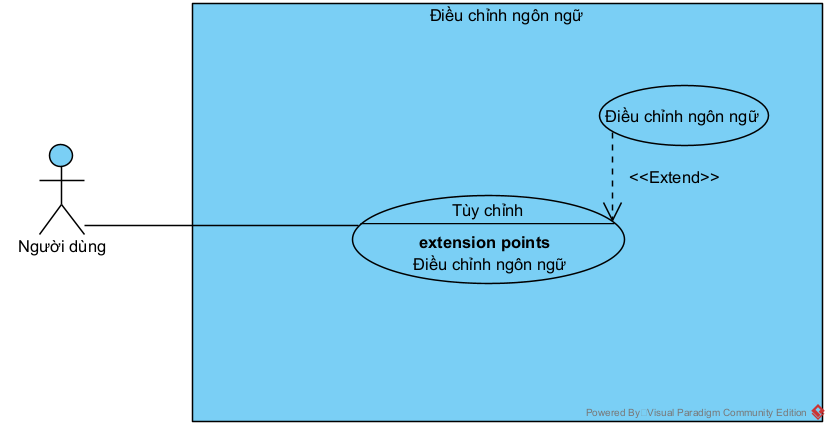
|  |  |
| --- | --- |
| Tên usecase | Điều chỉnh âm lượng |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng tùy chọn |
| Hậu điều kiện | Âm lượng được điều chỉnh |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn vào chức năng điều chỉnh.  6. Giao diện điều chỉnh hiện lên với 3 chức năng chính là điều chỉnh âm thanh, điều chỉnh đồ họa và điều chỉnh ngôn ngữ.  7. Người dùng sử dụng chức năng điều chỉnh âm lượng bằng cách kéo slider tương ứng với mức âm lượng.  8. Hệ thống điều chỉnh âm thanh theo mức mà người dùng đã chọn. |
| Ngoại lệ |  |

#### Biểu đồ tuần tự



### Module điều chỉnh ngôn ngữ

#### Usecase chi tiết

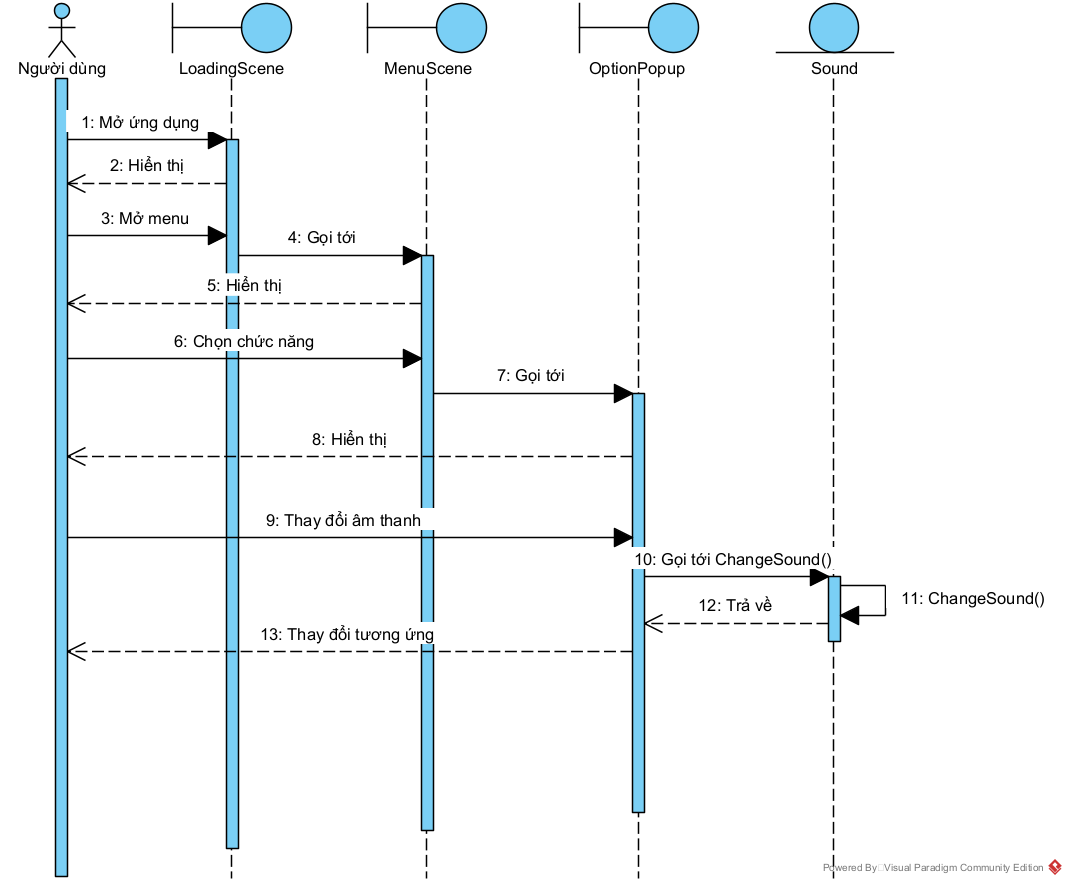


#### Kịch bản

Bảng . Kịch bản module điều chỉnh ngôn ngữ

|  |  |
| --- | --- |
| Tên usecase | Điều chỉnh ngôn ngữ |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng tùy chọn |
| Hậu điều kiện | Ngôn ngữ được điều chỉnh |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn vào chức năng điều chỉnh.  6. Giao diện điều chỉnh hiện lên với 3 chức năng chính là điều chỉnh âm thanh, điều chỉnh đồ họa và điều chỉnh ngôn ngữ.  7. Người dùng chọn vào thay đổi ngôn ngữ và lựa chọn ngôn ngữ phù hợp.  8. Hệ thống thực hiện thay đổi ngôn ngữ theo như ngôn ngữ mà người dùng đã chọn. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự



## Tổng kết

Chương 2 của đồ án đã trình bày phân tích thiết kế tổng quan của toàn bộ hệ thống bao gồm:

* Giới thiệu tổng quan về hệ thống
* Usecase tổng quan của hệ thống
* Biểu đồ lớp thực thể và thiết kế cơ sở dữ liệu cho hệ thống.

Chương 2 cũng tập trung vào phân tích thiết kế chi tiết từng module chức năng có mặt trong hệ thống bao gồm:

* Usecase chi tiết từng chức năng
* Kịch bản chuẩn cho module
* Biểu đồ tuần tự của module

Từ đó cung cấp cái nhìn cụ thể hơn về hệ thống và các chức năng trong hệ thống. Trên cơ sở phân tích thiết kế đó, trong chương tiếp theo đồ án sẽ trình bày về cách cài đặt, triển khai hệ thống và kết quả thu được khi triển khai hệ thống.

# CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

Trong chương này, đồ án sẽ trình bày về cách cài đặt và triển khai hệ thống và kết quả thu được sau khi cài đặt.

## Yêu cầu hệ thống

* Hệ thống cài đặt source code:
  + Hệ điều hành: Window 7 Windows 7 (SP1+), Windows 10 and Windows 11, chỉ dùng phiên bản 64-bit.
  + CPU: kiến trúc X64 có hỗ trợ tập lệnh SSE2.
  + GraphicAPI: DX10, DX11, và DX12-capable GPUs
  + Một số yêu cầu khác: phần cứng hỗ trợ drive cho các API của Unity.
  + Cài đặt các trình biên dịch: c++, .NET,...
  + Git: Quản lý source code.
* Hệ thống chạy ứng dụng:
  + Hệ điều hành: Android.
  + API level 22(Android 5.1 Lollipop) – API level 34(Android 11.0).
  + Các hệ máy hỗ trợ Vulkan.
  + Bộ nhớ tối thiểu: 250MB.

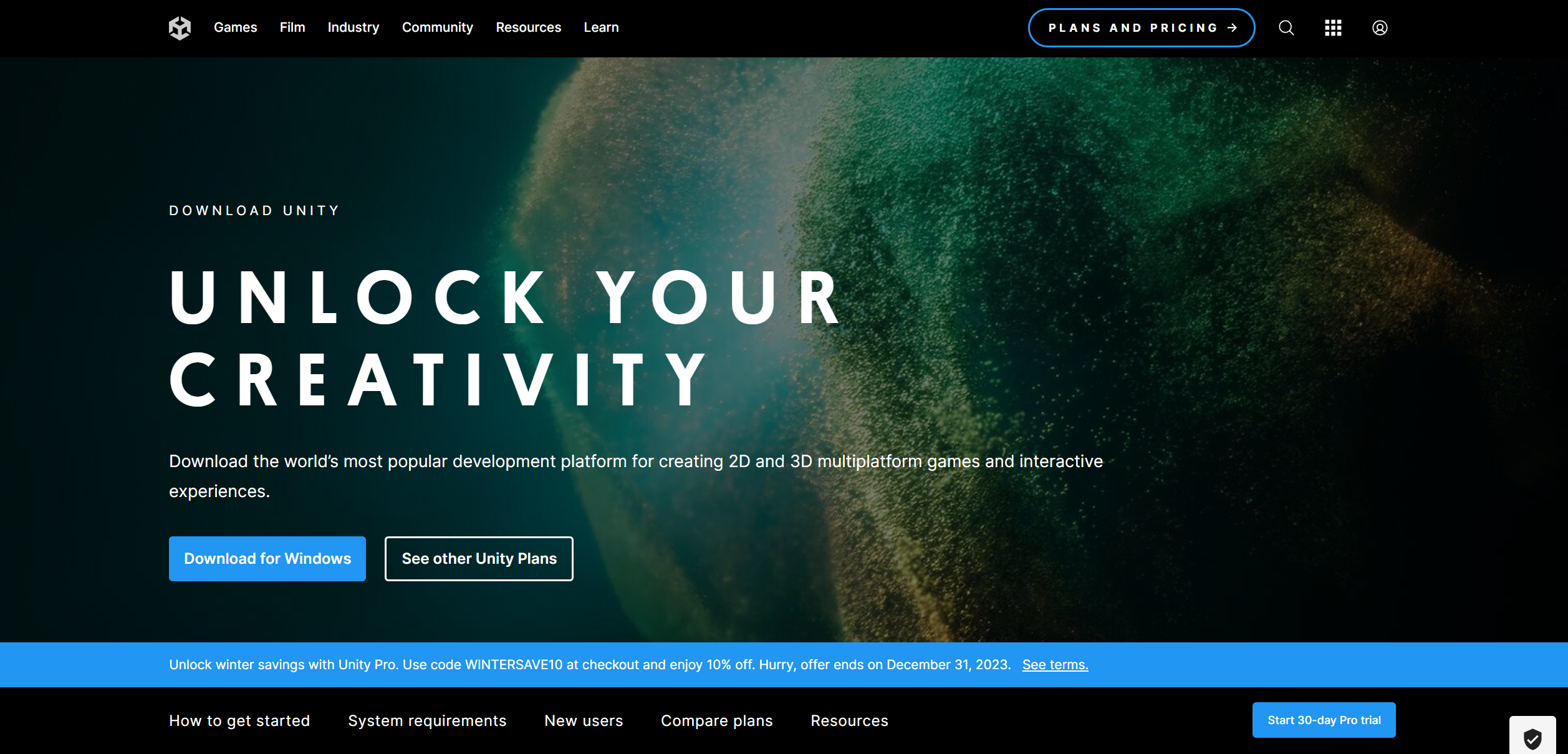
## Một số công cụ, thư viện hỗ trợ

* Jetbrain Rider
* Unity editer
* Unity hub
* Android studio

## Cài đặt

### 3.3.1 Cài đặt Unity

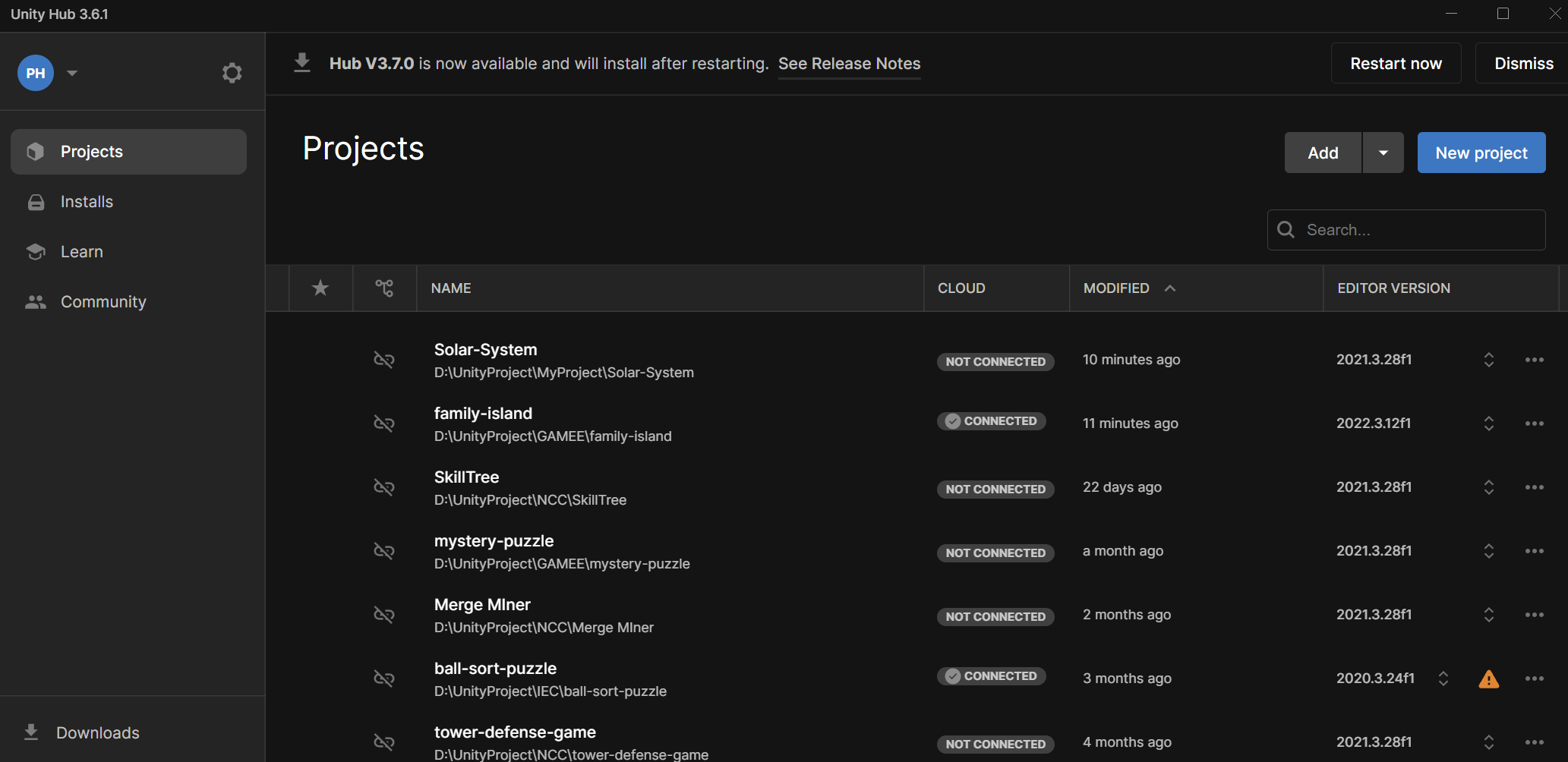
Bước 1: Tải Unity hub



Để tải Unity Hub trước hết truy cập vào trang chủ của Untiy: [*https://unity.com/download*](https://unity.com/download). Màn hình trang chủ hiện lên có các thông tin cũng như các trường giúp người dùng cài đặt phần mềm Unity Hub. Chọn vào ô “Download for Windows” để tải file cài đặt .exe về.

Bước 2: Cài đặt

Sau khi đã tải về, tiến hành cài đặt mặc định hoặc custom một số thông tin tùy thuộc vào nhu cầu của người dùng. Sau khi cài đặt hoàn tất, đăng nhập tài khoản Unity để tiếp tục sử dụng.



Bước 3: Cài đặt Unity Editer

Do dự án được xây dựng trên nền Unity Editer phiên bản 2021.3.28f1 nên để tối ưu thời gian cài đặt, chúng ta nên sử dụng phiên bản Unity 2021.3.28f1(có thể sử dụng các phiên bản khác mới hơn bản Unity hiện thời của dự án).

Từ giao diện chính của Unity Hub, chọn mục Install và chọn phiên bản Unity Editer muốn cài đặt.

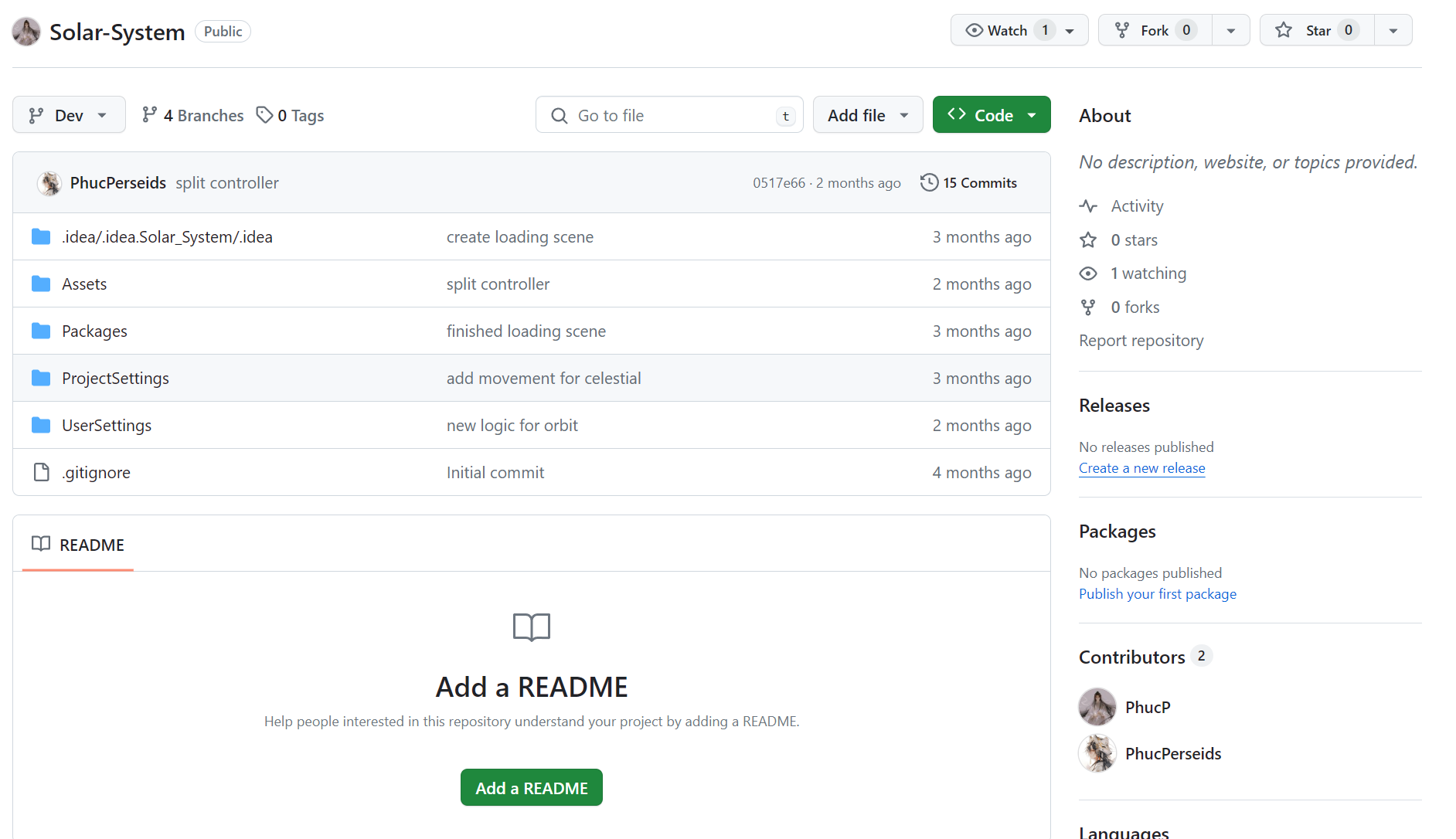


Sau khi cài đặt thành công Unity Editer, khởi chạy lại Unity Hub và có thể sử dụng.

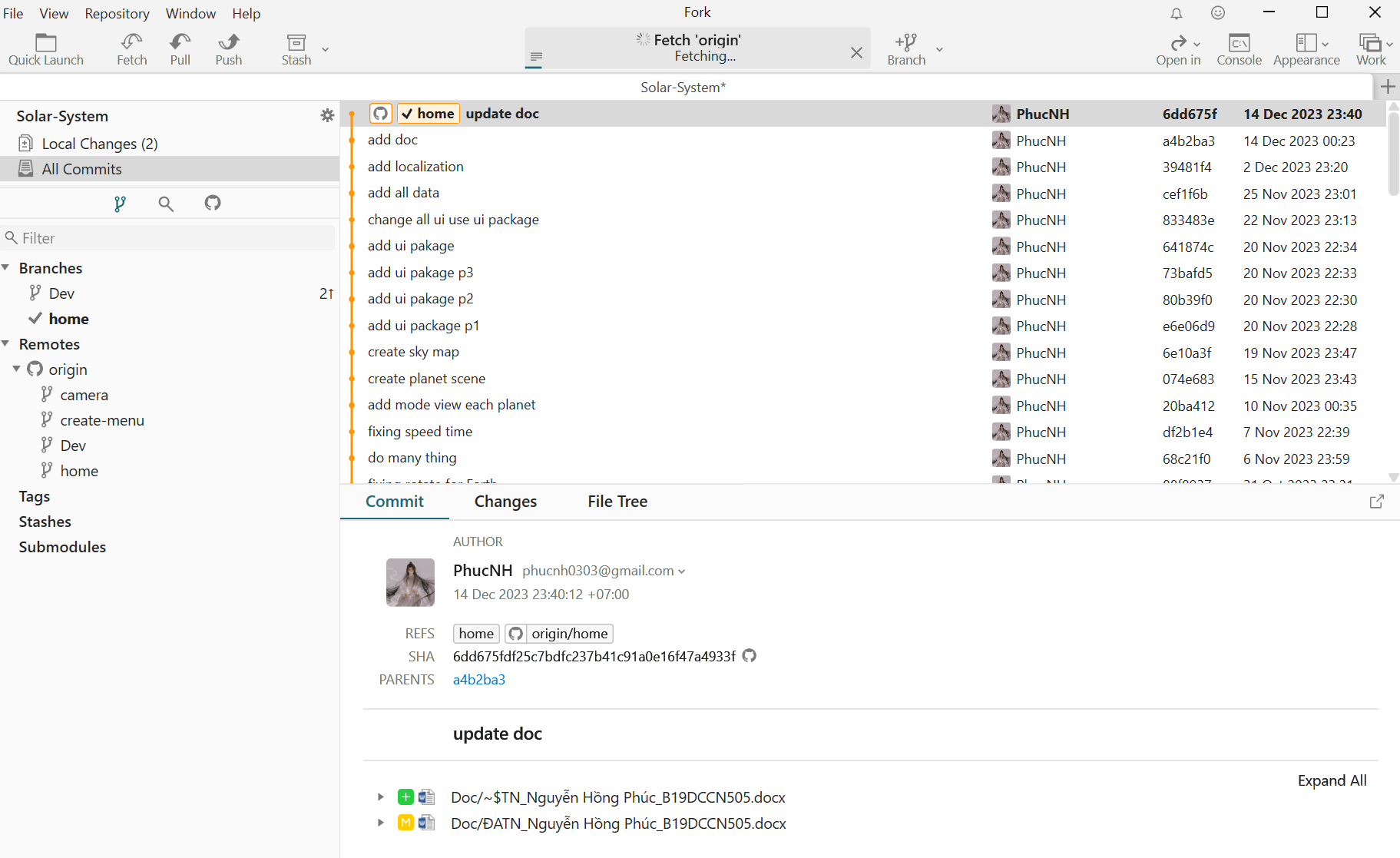
### 3.3.2 Khởi chạy dự án trên Untiy Editer

Bước 1: Tải xuống dự án

Sử dụng các công cụ quản lý Git để tải dự án từ trên Git Hub về, link tới Git Hub của dự án: [*https://github.com/PhucP/Solar-System.git*](https://github.com/PhucP/Solar-System.git)

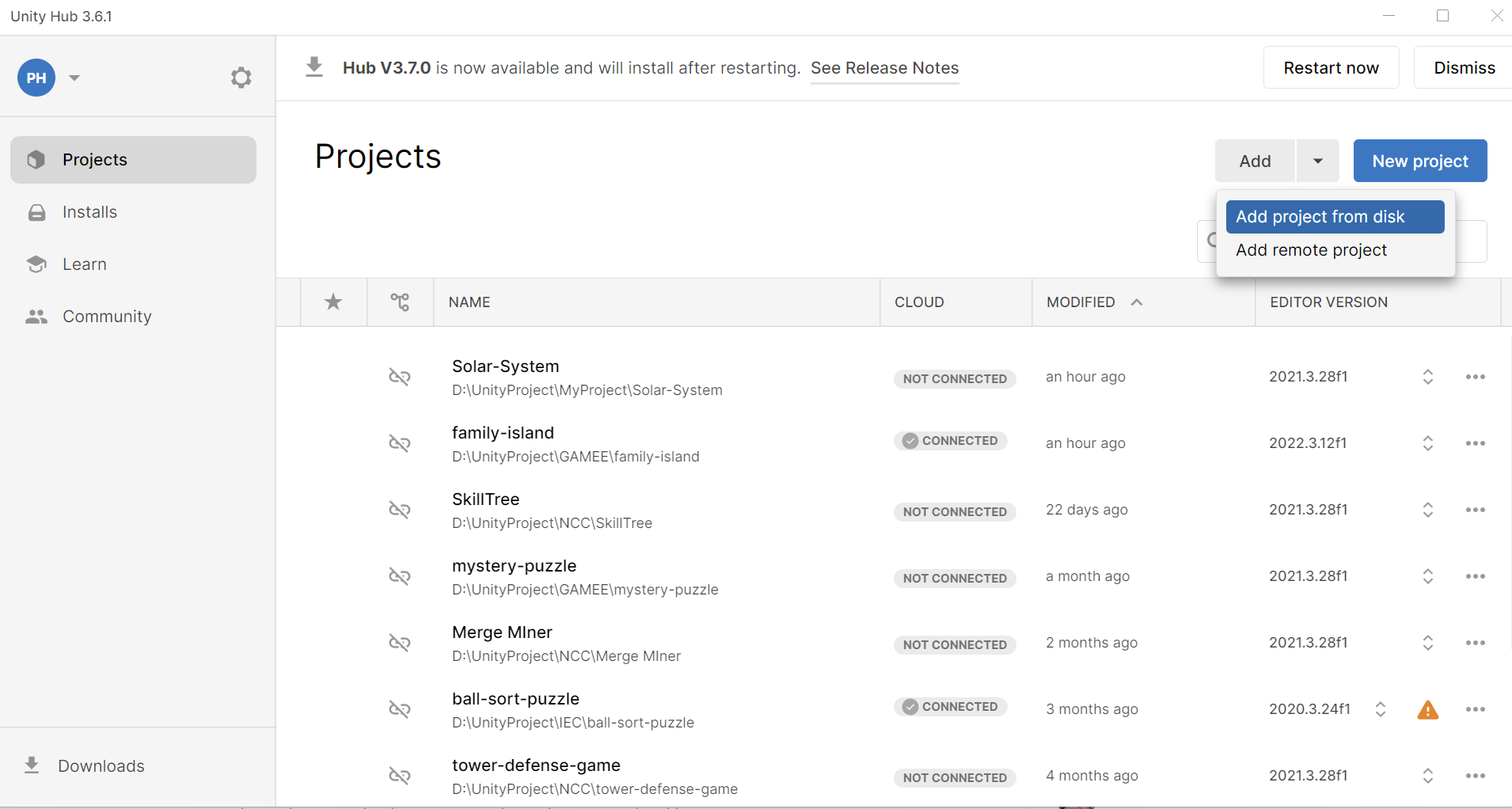


Tại đây sử dụng Fork như một công cụ quản lý mã nguồn của dự án.

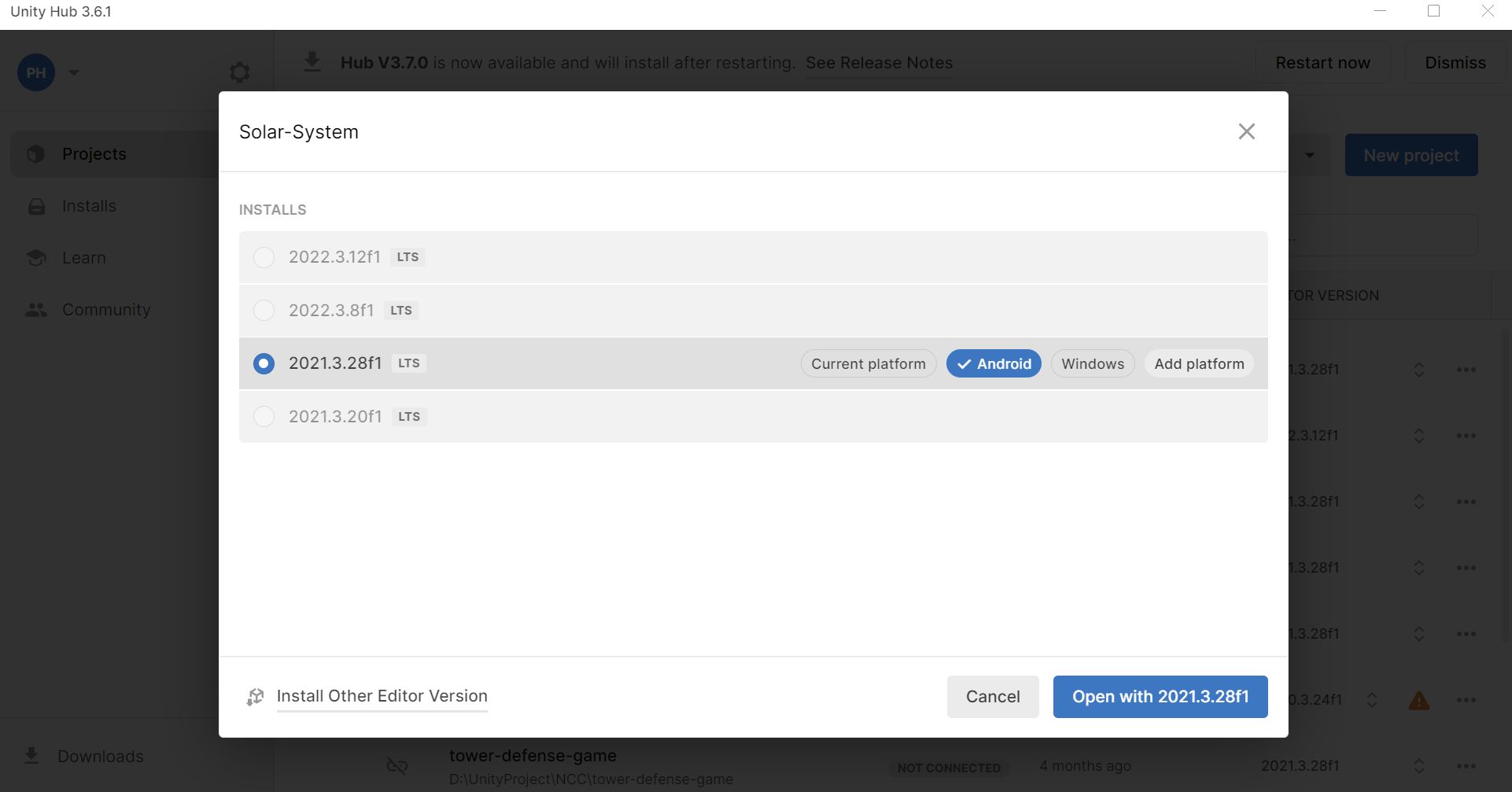


Bước 2: Liên kết source code với Unity Editer

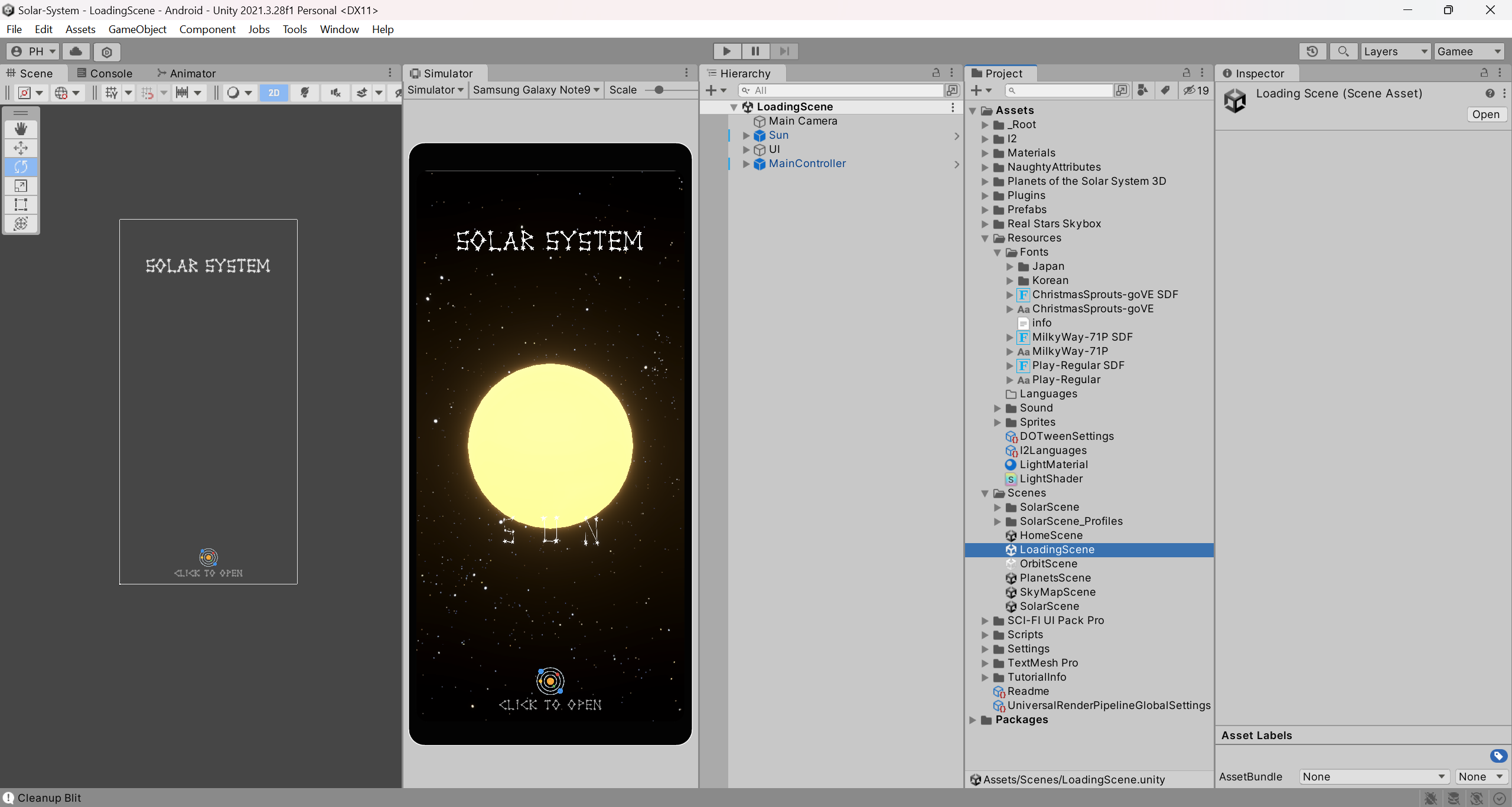
Mở Unity Hub, trong phần “Add” chọn “Add project from disk”, sau đó liên kết với đường dẫn tới project.



Chọn phiên bản Unity Editer phù hợp với phiên bản của dự án, mở nó trong Android platform.



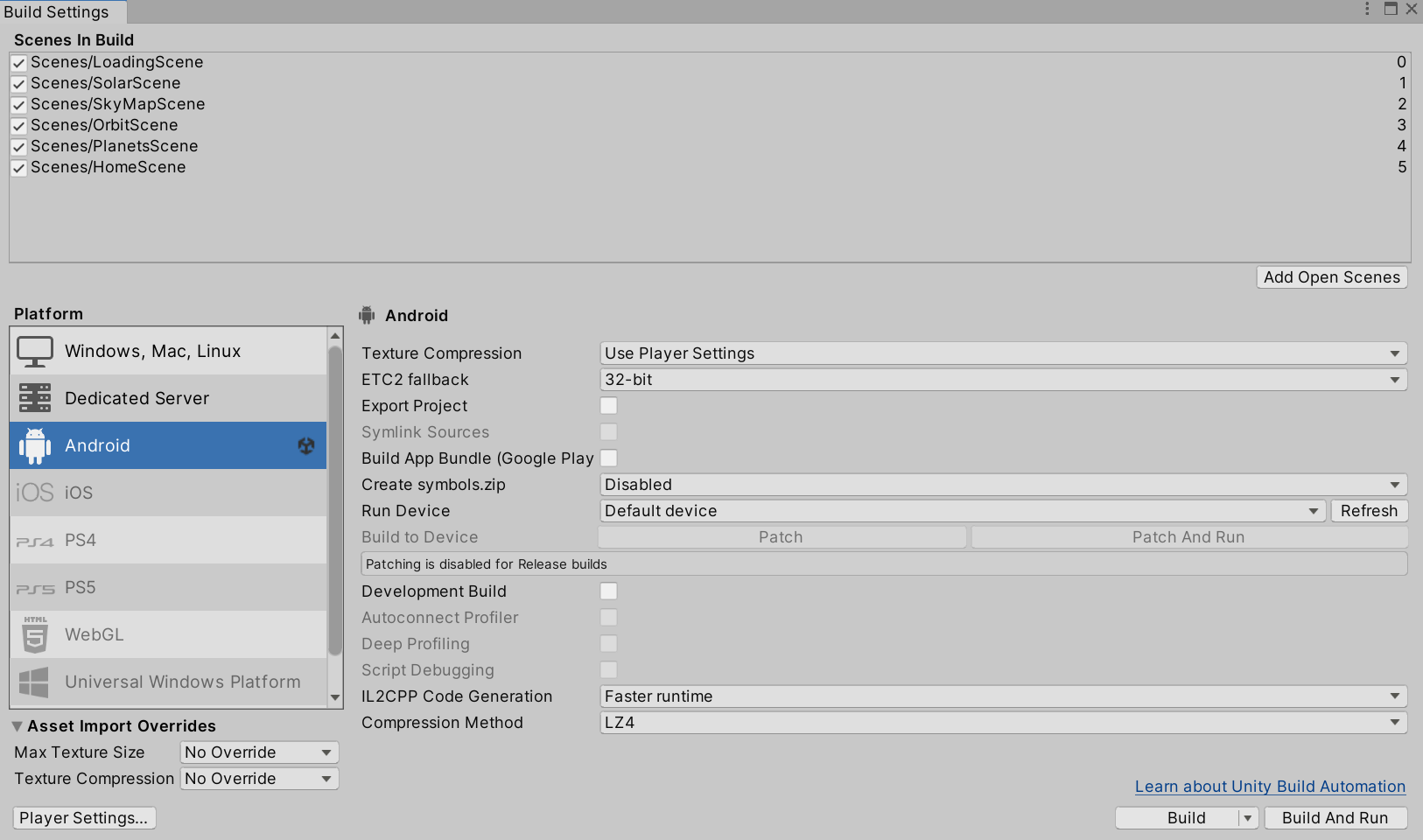
Sau khi hoàn thành các bước trên, dự án được mở trên Unity Editer.



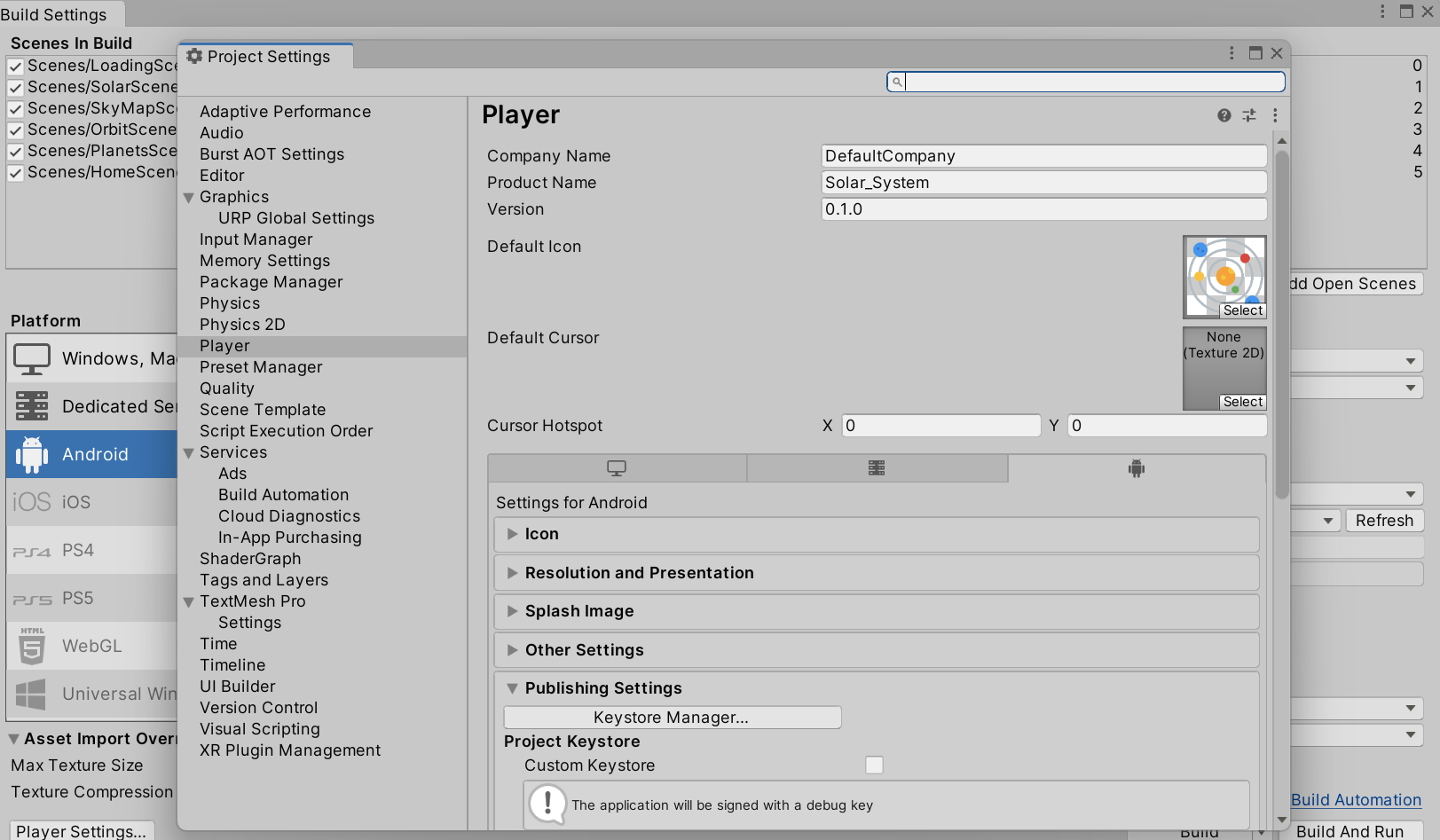
### 3.3.3 Thực hiện build ra APK và cài đặt trên điện thoại Android

Bước 1: Thực hiện build APK

Tại giao diện chính của Unity Editer trong mục “File” chọn “Build Setting”.

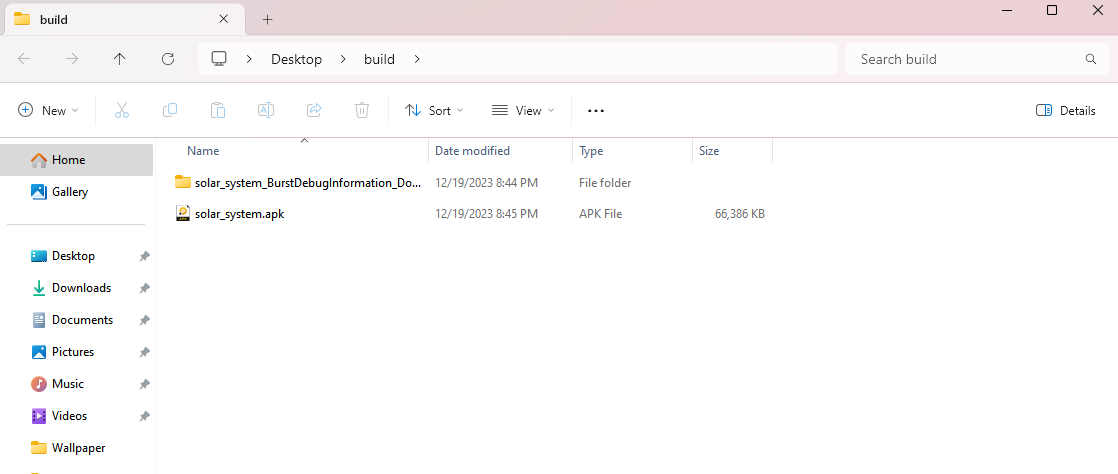


Tại bước này có thể trực tiếp build ra APK, tuy nhiên để bản build đáp ứng tốt cho cấu hình của máy di động, chúng ta chọn “Player setting” để tùy biến các thông số.



Một số mục quan trọng mà chúng ta cần lưu ý khi build là: phiên bản build “Version”, icon, Scripting backend(nên chọn IL2CPP để có chất lượng ảnh tốt hơn nhưng đánh đổi lại là dung lượng lớn), phiên bản Android tối thiểu và tối đa, …

Sau khi đã tinh chỉnh xong các thuộc tính cần thiết, bắt đầu thực hiện build bằng cách chọn build bên ngoài build setting. Kết quả của tất cả quá trình là file APK và file app bundle, burst debug in formation.



### 3.3.4 Khởi chạy ứng dụng trên thiết bị thực tế

Bước 1: Cài đặt trên thiết bị di động

Tải xuống file APK vừa build trển máy tính và chọn các bước cài đặt mặc định trên thiết bị di động.



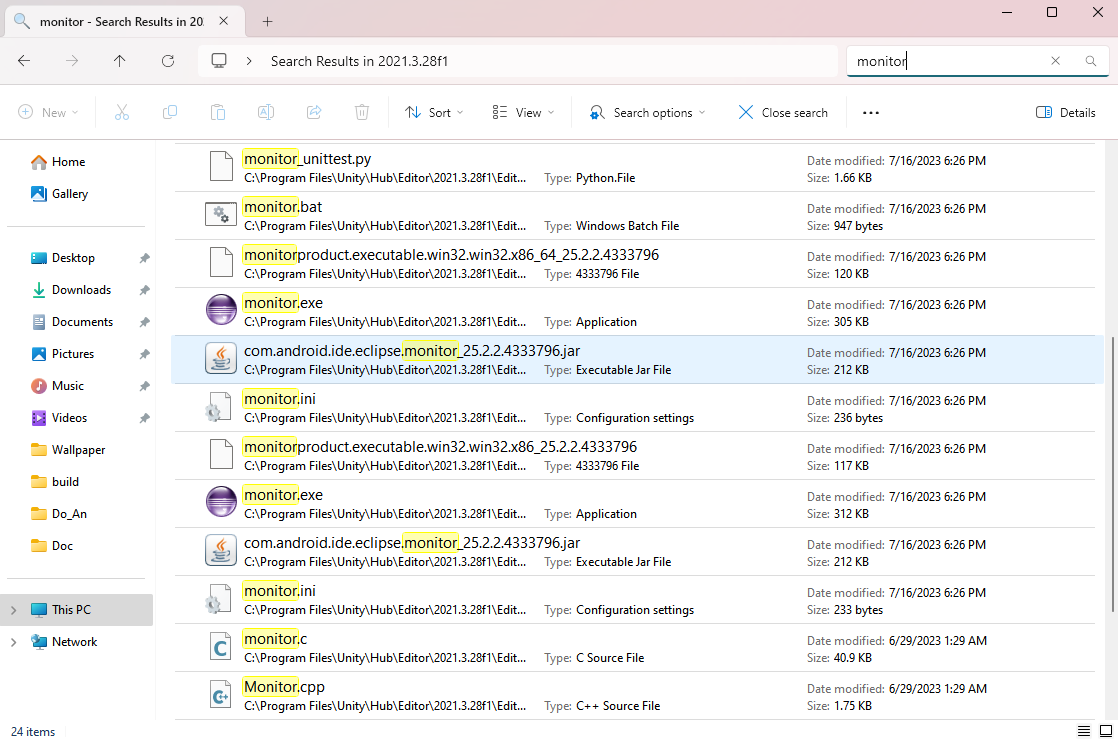
Bước 2: Mở ứng dụng và sử dụng



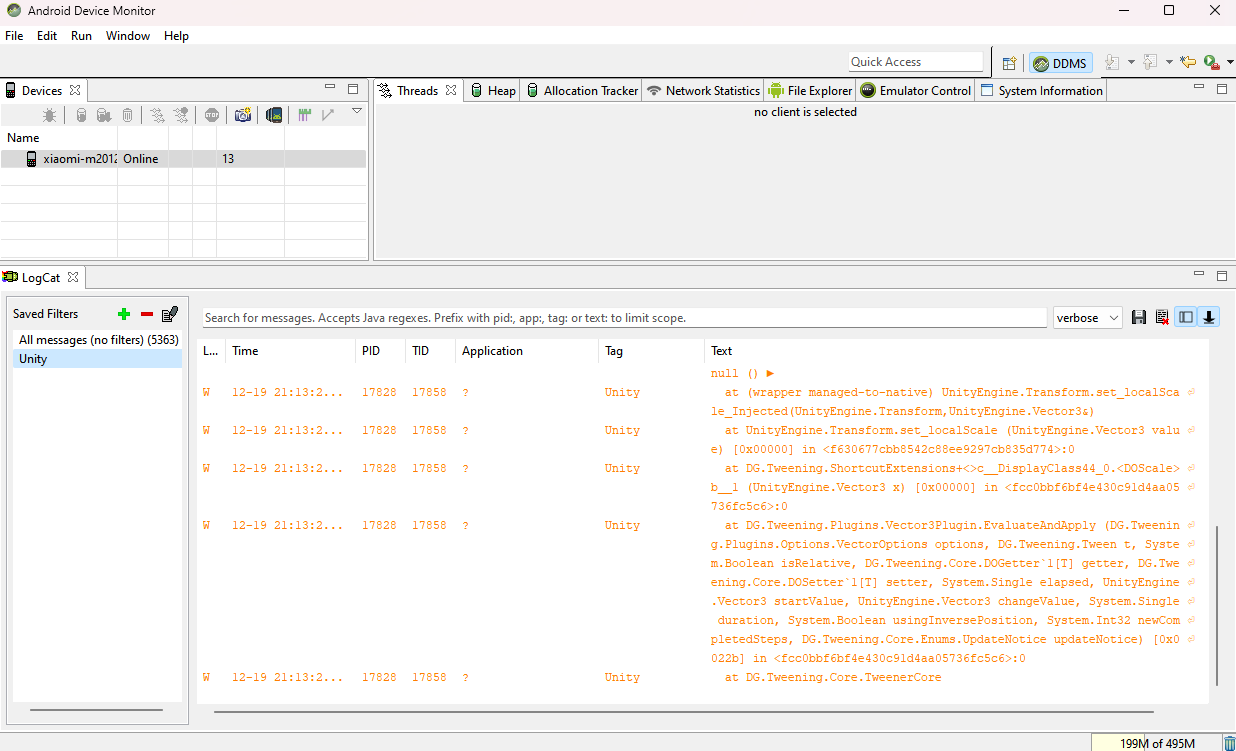
### 3.3.5 Sử dụng debuging trên Android Studio hoặc Monitor để gỡ lỗi

Trong trường hợp cài đặt trên thiết bị di động phát sinh lỗi, có thể log lỗi thông qua các tool như Android Studio hoặc Monitor. Trong trường hợp dưới đây là cách gỡ lỗi cũng như log các thông tin lỗi thông qua Monitor.

Bước 1: Mở folder chứa phiên bản Unity hiện sử dụng cho dự án sau đó tìm Monitor được cài sẵn cùng với phiên bản(có thể yêu cầu cài đặt môi trường Java trước khi chạy).



Bước 2: Kết nối thiết bị di động với máy tính đồng thời mở Monitor để xem các log.



### Một số hình ảnh từ ứng dụng

## Kết luận

Chương 3 đã trình bày hướng dẫn về cách cài đặt hệ thống bao gồm cài đặt mã nguồn cùng các công cụ liên quan.Dựa vào kết quả của chương 3, chương tiếp theo sẽ đưa ra kết luận, đánh giá về kết quả cài đặt hệ thống và phương hướng phát triển của hệ thống sau này.

# PHẦN TỔNG KẾT

Nội dung chính của chương là đưa ra đánh giá về kết quả thu được khi thực hiện đồ án và đề ra phương hướng phát triển cho đồ án

## Đánh giá kết quả của đồ án

Sau thời gian tìm hiểu, nghiên cứu các công nghệ để thực hiện đồ án với đề tài xây dựng website chấm bài trực tuyến cho việc học và luyện tập lập trình, bản thân em đã thu được những kết quả cụ thể như sau:

* Xây dựng hoàn chỉnh hệ thống chấm code online phục vụ cho việc học và luyện tập lập trình.
* Hiểu được cách giao tiếp giữa máy khách và máy chủ trong mô hình REST.
* Vận dụng được các kiến thức đã học để xây dựng các API theo chuẩn REST.
* Tìm hiểu thêm được những công nghệ mới: mongoDB, reactJs, express, socket.io.
* Hiểu hơn về ngôn ngữ lập trình javascript đặc biệt là vấn đề về bất đồng bộ trong javascript.
* Vận dụng được các công cụ hỗ trợ cho javascript để phát triển ứng dụng nhanh hơn.

Bên cạnh đó, đồ án còn có nhiều hạn chế, cụ thể là:

* Số lượng bài tập trên hệ thống còn hạn chế
* Ngôn ngữ lập trình hỗ trợ trên hệ thống còn ít
* Hiệu năng hệ thống chưa đảm bảo
* Chưa tính toán được bộ nhớ sử dụng cho chương trình của người dùng
* Cài đặt trên các môi trường khác nhau còn xuất hiện nhiều lỗi

## Phương hướng phát triển

Với kết quả thu được, em đã đề ra phương hướng phát triển cho đồ án trong lai cần triển khai, bao gồm các nội dung sau:

* Thêm các tính năng mới để hệ thống có sức hút và mạnh mẽ hơn như chia sẻ màn hình code, chống gian lận trong khi làm contest, hỗ trợ thêm actor: giảng viên, thêm chức năng tạo lớp học, tạo task cho thành viên trong lớp học, thêm blog chia sẻ kiến thức.
* Cải thiện hiệu năng của hệ thống.
* Hỗ trợ thêm nhiều ngôn ngữ lập trình và phiên bản của các ngôn ngữ lập trình.
* Sử dụng docker cho việc cài đặt hệ thống để có thể dễ dàng thay đổi môi trường cài đặt.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "Stack Overflow Developer Survey 2022," [Online]. Available: https://survey.stackoverflow.co/2022/#technology. [Accessed 12 26 2022]. |
| [2] | "Node.js Architecture and 12 Best Practices for Node.js Development," [Online]. Available: https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development. [Accessed 12 26 2022]. |
| [3] | "What is REST? | Codecademy," [Online]. Available: https://www.codecademy.com/article/what-is-rest. [Accessed 12 26 2022]. |
| [4] | Trần Đình Quế, Nguyễn Mạnh Sơn, "Phân tích thiết kế Hệ Thống Thông Tin," in *Phân tích thiết kế Hệ Thống Thông Tin*, Học Viện Công Nghệ Bưu Chính Viễn Thông. |
| [5] | Stephen R.Schach, "Object-Oriented and Classical Software Engineering," in *Object-Oriented and Classical Software Engineering*, McGraw Hill. |
| [6] | Nguyễn Mạnh Hùng, Đỗ Thị Bích Ngọc, "Phân tích thiết kế và đảm bảo chất lượng phần mềm," in *Phân tích thiết kế và đảm bảo chất lượng phần mềm*, Học Viện Công Nghệ Bưu Chính Viễn Thông. |

# PHỤ LỤC