Computer Science
COC251
B927949

Creating a knowledge base for The Binding of Isaac

by

Tyler J. Bowcock

Supervisor: Dr. D D Freydenberger

Department of Computer Science
Loughborough University

May 2023

Abstract

Contents

List of Acronyms

HTTP	Hyper-Text Transfer Protocol

Chapter 1

Introduction

1.1 Problem Definiton

Item interactions are an important mechanic of most modern roguelike/roguelite games, including The Binding of Isaac. However, with hundreds of items, each with a handful of good or bad interactions, it is nearly impossible to effectively remember them all. Graph databases are purpose-built to store and navigate relationships.[1] The ouput of this project will be a web application that leverages this feature of graph databases to allow users to query item interactions in The Binding of Isaac.

1.2 Aims and Objectives

The goal of this project is to make querying item interactions in The Binding of Isaac quicker and easier by using graph databases. Users will also be able to update the data in the database to ensure it matches any changes in the game.

The aims of the project are to:

1. Create a graph database containing relevant data about The Binding of Isaac.
2. Develop a web application that utilises a graph database to helps users to find item interactions in the game.
3. Explore testing methodologies to aid in producing a stable application with high quality code.
4. Search for possible ways to extend the project with future updates.

1.3 Risks and Constraints

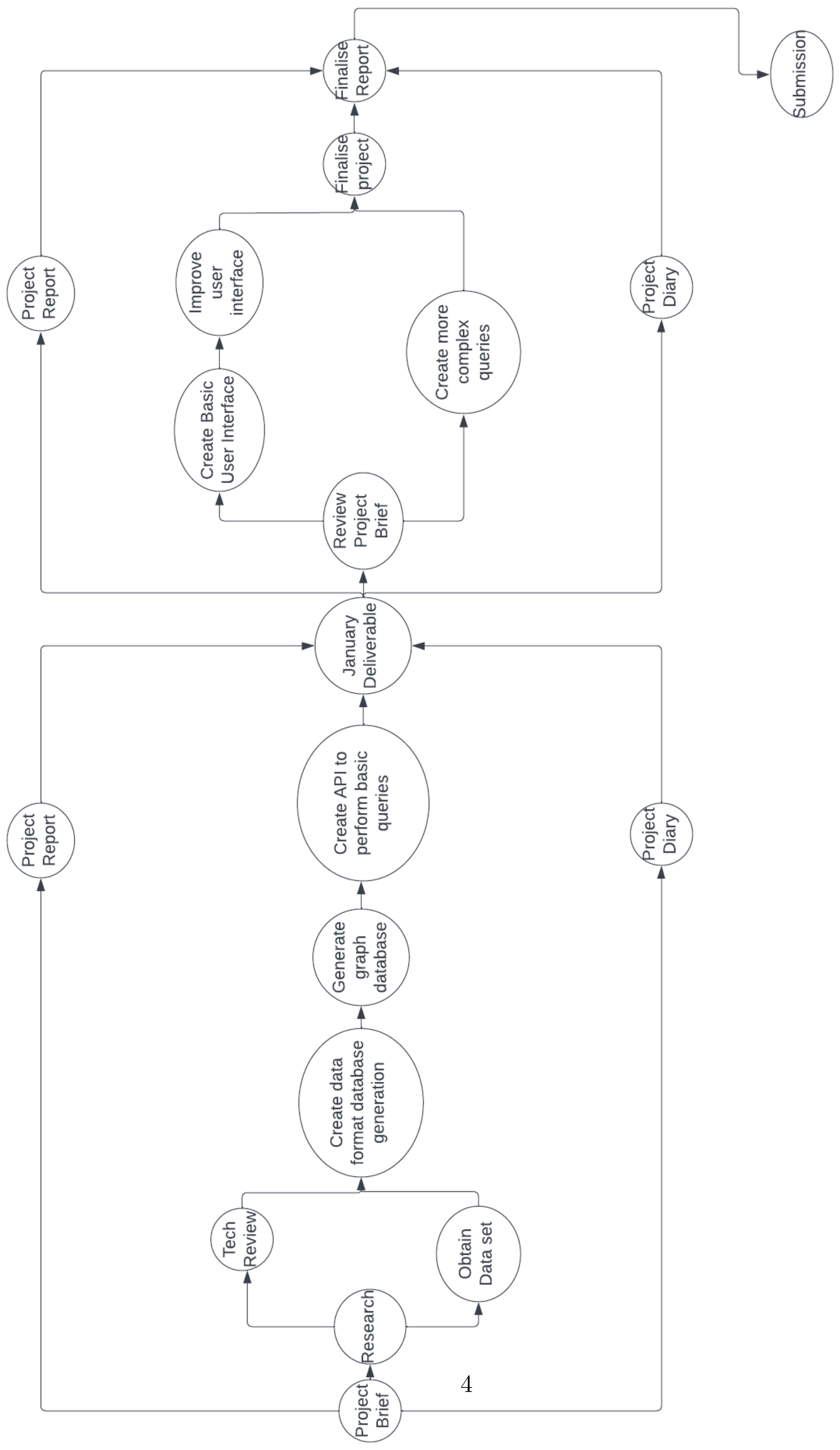
Cost

This project has no budget and so any services used in the development of the application will need to be free.

Dataset Availability

The data needed to create the database may become unavailable or unuseable.

1.4 Project Plan



Chapter 2

Background

2.1 Introduction

2.2 Existing Solutions

Fandom Wiki

The Binding of Isaac has two wiki sites hosted on the Fandom Wiki platform; one for the original flash game[2], and one for the modern version, commonly referred to as 'Rebirth'[3]. For the purposes of this project we will only be considering the modern version as it is widely considered the 'goto' version within the game's community.

The website is contains comprehensive information on all aspects of the game, and it is continualy updated by the community. Users can navigate the site using either predefined categories or a powerful search tool.

Advantages

- Contains information on all aspects of the game
- Actively maintained by the community
- Usful search functionality

Disadvantages

- So much information can make it hard to find what is relevant
- Unable to search for interactions, have to go via each item

Platinmum God

Platinmum God is a self-described 'Isaac Cheat Sheet'[4] and it contains item and key mechanic information for all versions of the game. The site is maintained by one person, and it claims to be more accurate than the community wiki as it update is 'tested thoroughly in the game using Cheat Engine'[5]. The information is split into pages based on the version of the game; users can navigate this using the item icons which are arrayed on the page, or by using the search functionality. The search tool has some supported keywords, but will still usually require entering an exact match to an entry in the data. For certain versions of the game there is also a synergy finder tool which lets the user enter two items to see how they interact. However, this is limited to older versions of the game and only a small set of the items are actually included in the tool.

Advantages

- Information is more reliable than the community wiki
- Easier to reference quickly due to there being less information

Disadvantages

- Only one maintainer can mean long update times
- Only contains basic information about each item
- Limited or no synergy information for most items
- Harder to find items without knowing the name or what the item looks like

2.3 Technology Review

2.3.1 Client Side Framework

Angular

‘Angular is an application-design framework and development platform for creating efficient and sophisticated single-page apps.’[6]

React

‘React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.’[7]

Vue

‘An approachable, performant and versatile framework for building web user interfaces.’[8]

Conclusion

2.3.2 Server Side Framework

Django

‘Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.’[9]

Flask

‘Flask is a lightweight WSGI web application framework.’[10]

Conclusion

2.3.3 Database

Neo4j

‘Neo4j is an open-source, NoSQL, native graph database that provides an ACID-compliant transactional backend for your applications’[11]

Amazon Neptune

‘Amazon Neptune is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency.’[1]

Conclusion

The decision was made to use Neo4j. This is primarily because the Aura platfrom provides a permanent free database instance which has ample resources for this project. There also exists a Python libraries, neomodel and django-neomodel, for easily integrating database access in Django.

2.4 Conclusion

Chapter 3

Requirements

3.1 Introduction

3.2 User Requirements

3.3 System Requirements

3.4 Wireframes

3.5 Conclusion

Chapter 4

Design

4.1 Introduction

4.2 System Design

4.3 User Interface Design

4.4 Conclusion

Chapter 5

Implementation

5.1 Introduction

5.2 Tools

5.2.1 IDE

5.2.2 Version Control

5.2.3 Project Management

5.2.4 Database Visualisation

5.2.5 Hosting

5.2.6 CI/CD

5.2.7 Testing

Postman

Look at what testing frameworks can be used

5.2.8 Libraries

5.2.9 Client Side

5.2.10 Server Side

5.3 Data Processing

5.4 Database Interacion

5.5 Client Side

5.6 Conlcusion

Chapter 6

Testing

6.1 Introduction

6.1.1 Functionality Testing

6.1.2 Non-Functionality Testing

6.2 Conclusion

Chapter 7

Evaluation

7.1 Introduction

7.2 Project Evaluation

7.3 Future Work

7.4 Lessons Learned

7.5 Conclusion

Appendix A

Appendix

References

[1] *What Is a Graph Database?* Amazon Web Services, Inc. URL: `https://aws.amazon.com/nosql/graph/` (visited on 01/06/2023).

[2] *The Binding of Isaac Wiki.* URL: `https://bindingofisaac.fandom.com/wiki/The_Binding_of_Isaac_Wiki` (visited on 01/09/2023).

[3] *Binding of Isaac: Rebirth Wiki.* URL: `https://bindingofisaacrebirth.fandom.com/wiki/Binding_of_Isaac:_Rebirth_Wiki` (visited on 01/09/2023).

[4] *Isaac Cheat Sheet - Platinum God.* URL: `https://platinumgod.co.uk/` (visited on 01/09/2023).

[5] *Frequently Asked Questions - Isaac Cheat Sheet - Platinum God.* URL: `https://platinumgod.co.uk/faq` (visited on 01/09/2023).

[6] *Angular - Introduction to the Angular Docs.* URL: `https://angular.io/docs` (visited on 01/09/2023).

[7] *Tutorial: Intro to React – React.* URL: `https://reactjs.org/tutorial/tutorial.html` (visited on 01/09/2023).

[8] *Vue.Js - The Progressive JavaScript Framework — Vue.Js.* URL: `https://vuejs.org/` (visited on 01/09/2023).

[9] *Django.* Django Project. URL: `https://www.djangoproject.com/` (visited on 01/09/2023).

[10] Armin Ronacher. *Flask: A Simple Framework for Building Complex Web Applications.* Version 2.2.2. URL: `https://palletsprojects.com/p/flask` (visited on 01/09/2023).

[11] *What Is a Graph Database? - Developer Guides.* Neo4j Graph Data Platform. URL: `https://neo4j.com/developer/graph-database/` (visited on 01/09/2023).