CYAN: Choose Your Adventure Neatly

Spike Atkinson
1902160

Supervised by:
David Richerby & Juntao Yu

BSc Computer Science
CE301

I would like to take this opportunity to acknowledge the valuable contributions of several individuals who greatly enriched my time working on this project.

First and foremost, I want to express my gratitude to my supervisor, David Richerby, for his unwavering support and guidance throughout this project. His expertise and wisdom have been invaluable, and I have learned a great deal from his insights and feedback.

I would also like to thank my friends who provided early feedback on the software and helped shape it into its final form. Their suggestions and critiques were instrumental in improving the functionality and usability of the software.

Lastly, I am deeply indebted to my partner, Holly, who has been a shining beacon of support and encouragement throughout this project. She endured countless hours of abandonment as I worked tirelessly to bring this software to fruition. Her love and patience have been my rock.

Abstract

This report documents the creation of CYAN, a Java-based interactive fiction engine that enables the user to create and play text-based adventure games. CYAN provides an intuitive user interface, consisting of a writing component (CYAN Writer) for creating and editing game content and a reader component (CYAN Reader) for playing games. CYAN Writer allows the user to create and link different situations, choices, and events to form a branching narrative. CYAN Reader displays the game text and choices to the user, allowing them to progress through the story.

Matters of social impact, sustainability, legality, and ethics have been deeply considered and discussed within the context of CYAN, ensuring that my actions and decisions align with my values and objectives.

Throughout the project, the Jira project-management tool was used to plan and manage tasks, and the agile development methodology was adopted to ensure that the project was delivered on time and to a high standard.

Overall, the CYAN project has been a great success, resulting in the creation of a user-friendly interactive fiction engine that has the potential to inspire a new generation of game developers and storytellers.

Table of Contents

Literature Review

"Choose Your Own Adventure" (CYOA) is a style of interactive fiction in which the reader/player takes on the role of protagonist and makes choices that determine the course and outcome of the story. Each narrative point typically presents two or more options to the player, each leading to a new narrative point and further choices. The player must navigate the story, making choices until they reach the ending, or one of several possible endings.
CYAN is a software project that aims to provide a user-friendly tool for creating CYOA-style stories with a graphical user interface. The goal of CYAN is to make it easy for anyone and everyone, regardless of programming experience, to create their own interactive stories and share them with others. To achieve this, CYAN requires a combination of effective GUI design, and a deep understanding of CYOA-style storytelling.

Design, Prototyping, Fun, and CYOA

CYAN was rapidly prototyped early in development, based on a strong GUI concept, and tested early. These are principles I learned from "Beyond Game Design: Nine Steps Toward Creating Better Videogames"[1]. This book focuses on providing game designers with a practical framework for creating better video games, and provides a step-by-step guide that covers various aspects of game development, from the concept stage to post-release support.

In the context of CYAN, Beyond Game Design was particularly helpful in guiding through the design process. The book emphasizes the importance of creating a strong concept and prototyping early on in the development process, which was beneficial when working on CYAN, because I was aiming to provide users with an easy-to-use tool for creating Choose Your Own Adventure stories, early prototyping allowed for selection of traits that were more intuitive over others.

The book also highlights the importance of play testing and user feedback, which helped to improve CYAN's usability and ensuring that it meets the needs of its target audience. The book also covers the topic of production, which was useful in ensuring that CYAN is developed efficiently and within the allocated time.

Beyond Game Design describes theory surrounding what makes games fun. By understanding the kinds of fun that the CYOA format has to offer, I was able to make sure that the user is able to implement these kinds of fun, for example, the uncertainty of not knowing exactly what a choice will lead to, would not work if you could see where the choice leads before selecting it, so the next situation is obscured behind the choice.

The concept of CYAN, and indeed three quarters of the letters in the acronym, come from the long established game format based on “Choose Your Own Adventure”, the genre-defining series of books published in the US from 1979 to 1998[2]. "The Warlock of Firetop Mountain"[3] is a game-book written by Steve Jackson and Ian Livingstone, first published in 1982. The book is notable for

being the first in the “Fighting Fantasy” series, and for popularising the concept of Choose Your Own Adventure-style game-books in the UK. In the book, players embark on a quest that takes place mostly inside a maze-like dungeon, making decisions that determine the outcome of their adventure.

In the context of my project, CYAN, "The Warlock of Firetop Mountain" was a useful reference for creating Choose Your Own Adventure stories. The book's structure and narrative provided inspiration for my own test stories, while also demonstrating how branching paths can be used to create engaging and immersive experiences.

The book's success also highlights the potential appeal of CYAN's concept. The Fighting Fantasy series has sold millions of copies worldwide, demonstrating the enduring popularity of Choose Your Own Adventure-style storytelling. By providing users with an easy-to-use tool for creating their own stories, CYAN has the potential to tap into this market and provide a new platform for interactive storytelling.

“Beyond Game Design: Nine Steps Toward Creating Better Videogames"” and “The Warlock of Firetop Mountain” are valuable references for game designers and developers, especially those working on projects like CYAN that aim to provide users with a user-friendly tool for creating their own content. The emphasis on practical advice and step-by-step guidance in “Beyond Game Design” were useful in ensuring that CYAN was developed effectively and meets the needs of its users, while the structure and narrative of “Firetop Mountain” provides inspiration for new stories, while also highlighting the potential appeal of CYAN's concept.

Java Swing

The entire user interface of CYAN is constructed using Swing, a GUI widget toolkit for Java. The Swing Tutorial on www.javatpoint.com [4] is a comprehensive guide that provides step-by-step instructions on how to create various components using Swing; including buttons, text fields, labels, and menus. It also covers topics such as layouts, event handling, and customising components.

Having decided on my concept and development goals/methods, this Swing tutorial was a helpful reference for building my GUI. Since CYAN is designed for users with no programming knowledge, it was important to ensure that the GUI was intuitive and easy to use. The Swing tutorial covers a wide range of components that can be used to create a visually appealing and user-friendly interface.

The tutorial's coverage of layouts and event handling was also useful when developing CYAN's GUI. Choosing the right layout was essential to ensure that the GUI is organized and easy to navigate, while event handling is crucial for implementing user interactions such as button clicks and menu selections.

The Swing Tutorial on www.javatpoint.com is a valuable resource for developers working on projects like CYAN that require the creation of a user-friendly GUI. The tutorial's step-by-step

instructions, examples, and explanations can help developers learn the basics of Swing and create an effective GUI for their application.

Existing Engines

Research was conducted on CYAN's competing products. Three main Competitors were found. Twine[5], Inform7[6], and Quest[7]. Of these three, Quest is the most direct competitor as it is the only one that does not require specific Syntax to be used for interactivity, although I have concluded that Quest is much more complex than CYAN is intended to be. High complexity, even comprised of simple sub-systems could be just as intimidating to the new user as having to learn a new programming language. Thus, implementing systems with high complexity has been avoided during CYAN's development.

Literature Summary

To ensure that CYAN was developed effectively and would meet the needs of its users, the above resources were consulted before and during the development process. "Beyond Game Design" gave an insight into what makes games such as the CYOA's created using CYAN fun, and emphasized the importance of prototyping, play-testing, and user feedback, which helped to improve CYAN's usability. The Java Swing tutorial on www.javatpoint.com provided a solid foundation for building a user-friendly GUI, while "The Warlock of Firetop Mountain" demonstrated how branching paths can be used to create engaging and immersive stories, inspiring the mechanics to be used for the development of new stories in CYAN. By integrating the concepts and ideas presented in each review, an effective tool for creating Choose Your Own Adventure-style stories was created.

Problem Addressed

CYAN addresses the problem of creating CYOA stories without programming knowledge. With CYAN, users can easily create interactive stories without the need for complex programming languages or software. Unlike other interactive story-making tools such as Twine and Inform7, CYAN does not require users to have any programming knowledge (as with Inform7), or to learn any particular text formatting that is essentially programming language syntax (as with Twine). These tools rely on users to learn a programming syntax in order to create stories, especially stories with interactive elements and logic, which can be a barrier for users who lack programming knowledge or simply do not want to learn a new syntax.

In my research, I have found Quest to be the closest software to what CYAN is striving to be. However, I have observed that Quest offers players more freedom of choice as they must type out their decisions instead of clicking buttons. This approach adds a layer of complexity for both the writer and the player. In addition, Quest is purely text-based and lacks the functionality to include

other media such as images.

CYAN's user-friendly interface simplifies the process of creating interactive stories and makes it more accessible to a wider range of users, including those who may not have a background in programming or technical skills. By addressing this problem, CYAN aims to enable users to create their own interactive stories without feeling intimidated by the technical aspects of creating CYOA stories, which can also lead to a more diverse range of content being available. This is important because it allows for a greater range of voices and perspectives to be heard, leading to a more inclusive and diverse media landscape.

Additionally, CYAN could be used in education as a tool for teaching students about storytelling and game design. There is evidence to suggest that interactive-story-making software like CYAN could be a good educational tool[8], as Twine has been used to teach Humanities in high school.

Sustainability

As a digital product, CYAN has the potential to impact the environment in various ways. One of the main sustainability concerns related to digital products is their energy consumption, as the operation and maintenance of servers and data centres require significant amounts of energy, often derived from non-renewable sources (contributing to carbon emissions and exacerbating climate change).

Another issue that could be relevant to CYAN is electronic waste. 'E-waste' is a growing problem globally as the number of electronic devices with limited life spans continues to increase. When these devices reach the end of their life, they may contain hazardous materials that can harm the environment and human health if not disposed of properly. As CYAN is a digital platform, it is likely that it will be used on a variety of electronic devices, and it is important to consider how to mitigate the environmental impact of these devices when they are disposed of.

To address these issues, CYAN could take steps to reduce its energy consumption, such as using renewable energy sources for its servers and data centres, optimizing its code to reduce energy usage, and encouraging users to use energy-efficient devices when accessing the platform. CYAN could also consider strategies to minimize e-waste, such as encouraging users to recycle their electronic devices or providing guidance on how to properly dispose of them.

Legal

There are several legal issues that could have relevance to CYAN, particularly regarding data protection, intellectual property rights and user-generated content.

Firstly, CYAN development may need to consider data protection and privacy laws if it starts to collect and process user data. Updates will need to ensure that CYAN is in compliance with The Data Protection Act 2018 and has adequate measures in place to protect user data from unauthorized access or breaches.

Additionally, CYAN involves the creation of user-generated content, which raises questions about ownership and copyright. Users may create stories that are based on existing works or include elements of copyrighted material. Users may also create stories that include references to, or use trademarks (i.e. the names of existing products or brands). One solution could be to provide resources and guidance to users on how to properly obtain permissions or licenses for any third-party content used in their stories; However, copyright may only be an issue if CYAN monetised user-created stories [9].

Finally, CYAN development may need to consider laws and regulations related to user-generated content platforms. These may include regulations related to online speech, content moderation, and collection of personal information.

Careful consideration of the legal issues discussed here, and taking relevant action will help to avoid any legal liability or reputational harm.

Ethical

While CYAN itself may not present any significant ethical issues, the stories created using the platform could potentially raise ethical concerns.

One ethical issue that could arise is the content of the stories. As with any user-generated content platform, there is a risk that some stories created on CYAN may contain inappropriate or offensive material (such as profanity, graphic violence, sexual content, or hate speech). CYAN would need to have appropriate moderation and content policies in place to ensure that such content is not created or published on any platform that is associated with CYAN.

Another ethical issue is the potential for CYAN stories to influence the beliefs and attitudes of readers, particularly if they are targeted towards children or other vulnerable groups. Authors may use CYAN stories to promote particular ideologies or world views, carrying a risk that such content could be harmful or manipulative. CYAN would need to consider the potential impact of user-generated content and have appropriate policies in place to ensure that published content does not carry harmful influence.

Discussed as a legal issue in the above section, privacy is also an important ethical issue to consider. As with any platform that collects personal information, CYAN would need to have appropriate policies and measures in place to protect the privacy of its users; such as secure data storage and user consent when collecting and handling personal information.

Finally, CYAN would need to ensure that its platform is accessible to all users, regardless of their abilities. This could include making the platform compatible with screen readers and other assistive technologies, as well as providing clear and accessible instructions for use. Ensuring that the platform is accessible to all users is an ethical imperative that would help to promote inclusivity and

diversity within the CYAN community and prevent discriminatory exclusion.

Intellectual Property

CYAN, as a tool for creating interactive stories, can potentially raise issues related to intellectual property. Specifically, there are two primary areas of concern: ownership of the created content and potential infringement of existing intellectual property rights.

CYAN users create and share interactive stories, which means that they may be considered the creators and owners of the content they produce. However, CYAN itself is the tool that enables users to create their stories, so there may be questions and concerns around whether CYAN holds any rights to the stories created using its platform. It is important for CYAN to make clear in its terms of service and other legal documents how much ownership users retain of the content they create using CYAN.

Additionally, there is the issue of potential infringement of existing intellectual property rights. This is discussed in the Legal section above with regards to existing intellectual property, but as an intellectual property itself, CYAN may face issues related to copyright and ownership. The software, source code, and any creative content produced by the software, such as CYOA stories, may be subject to copyright laws. Also, if CYAN includes any third-party software or content, such as fonts or images, there may be additional copyright or licensing issues to consider as well as acquiring permission or licenses for the use of such third-party content.

Technical Documentation

CYAN is a Java-based software that uses Java Swing as its primary user interface toolkit. It is designed to be a user-friendly and accessible platform for creating and experiencing interactive CYOA stories. To use CYAN, users simply need to download and run an executable JAR file, which opens the Main class in the CYAN_Mutual Package.

Upon opening the program, the user is presented with a 'Main Menu'. A JWindow that displays options in the form of JButtons for accessing different components of CYAN, as in *Figure 1*. The user can choose to open CYAN Writer (The story writing component of CYAN), which allows them to create their own interactive CYOA stories using the intuitive and user-friendly interface. Alternatively, the user can choose to open CYAN Reader (the story reading component of CYAN), which allows them to read and interact with existing CYOA stories. The user can also close the application via the 'Close' JButton.

The use of Java Swing in CYAN allows for a consistent and familiar user interface across different platforms and devices. Additionally, it provides a wide range of components and widgets that can be used to create a visually appealing and interactive user interface for CYAN.

The Main class in the CYAN_Mutual Package serves as the entry point for CYAN and presents the user with options to open CYAN Writer or Reader, both of which are powered by the Java Swing user interface toolkit.

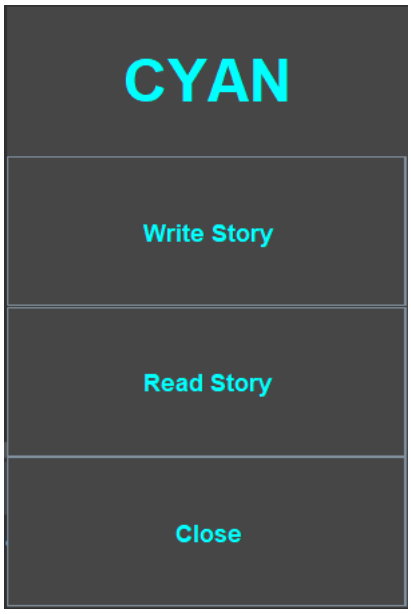

Figure 1: Main Menu

CYAN Writer

CYAN Writer is a crucial part of the interactive story creation process. Clicking on the 'Write Story' option in the Main class JWindow opens the Writer class in the CYAN_Writer package, which serves as the main class for CYAN Writer. The Writer class, in turn, calls the Overview Class which displays as a JFrame (Figure 2).

Initially, the Overview JFrame contains only a JMenuBar with the items "File" and "Edit". The "Edit" menu item is disabled until a Story is being edited, whereas the "File" menu gives the user options such as "New" to create a new story, "Save" to save the story (disabled until a story is being edited), and "Load" to load a previously saved story. When a story is loaded, the "edit" menu item is enabled, giving the user the option to edit "player" (not implemented at the time of writing), "items" (not implemented at the time of writing), or "events". After editing the story, the user can save the story to a Json file. If a new story is created, in order to keep stories organised, a file structure is created within a directory of the users choosing. It is contained within a folder named by the user which should be the title of the story ideally. Within this folder, similarly to Figure 8, a folder for assets is created (for images only currently), and a Json file is created for the story (see Gson below) containing the default story with one blank situation.

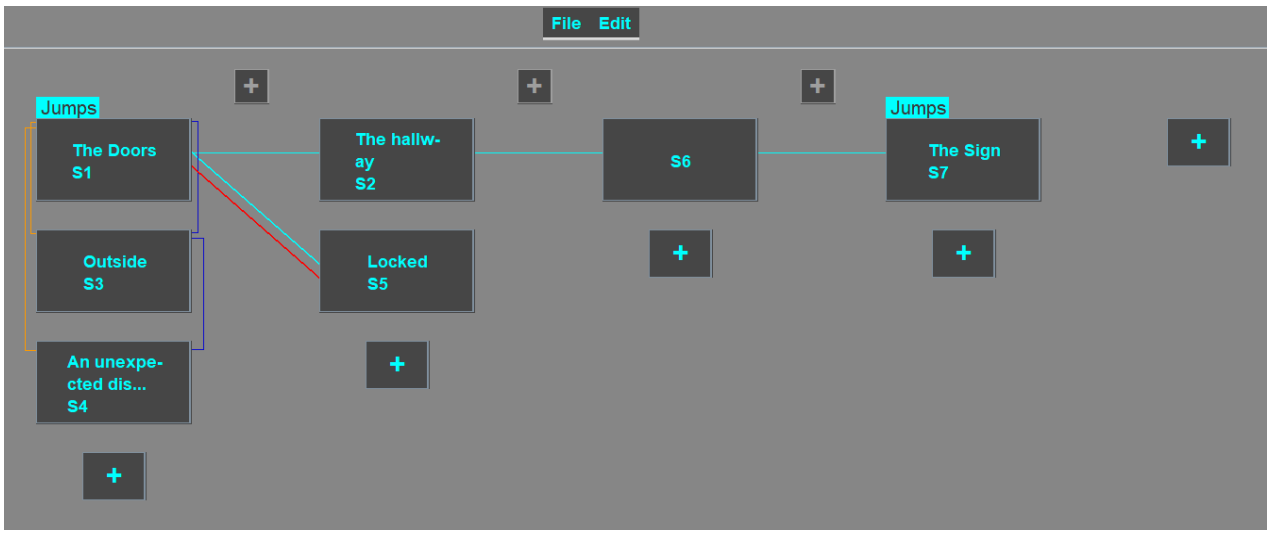

Figure 2: The Writer Overview

Situations

If the user starts a new story, they will initially have one 'Situation', the representation of a state in the story, comparable to a single page in a “Choose Your Own Adventure” book. Multiple Situations are arranged in a sequence of columns, to allow the author to group related situations together. A good use of this layout would be that a column represents all of the possible outcomes of the previous choice, while moving to the next column represents advancement the story’s timeline. another possible usage would be that a column represents all of the locations in a town, you can visit any location in any order, and then leaving the town is represented by a link to the next column. If the user loads a story for editing, there may be any number of situations present in the story.

The user can add new situations to the story by clicking the "+" buttons displayed in the Overview JFrame. There is a '+' button at the bottom of each column of situations, that will add a situation to the bottom of that respective column. Another in between each column that will insert a new column there with a new situation. And finally, one at the far right side of the columns that will add a new column to the far right with a new situation.

Each situation is represented by a JButton in the Overview JFrame, and the user can edit The situation's attributes via a "SituationEditor" JFrame (Figure 3) that opens upon clicking. The SituationEditor allows the user to edit the title, image, description, alternative description (shown on second and subsequent visits to this situation), and any choices that the user has in this situation, as well as the situations that those choices link to. When “add image” is clicked, the user is prompted to select an image from their computer, which is then copied to the “assets” folder in the story directory. The situation then contains a reference to this image and it will be displayed in CYAN Reader. When a link is created between two situations, it is visually represented in the Overview JFrame as a line between the two situation JButtons. Additionally, the user can delete situations via the Situation Editor.

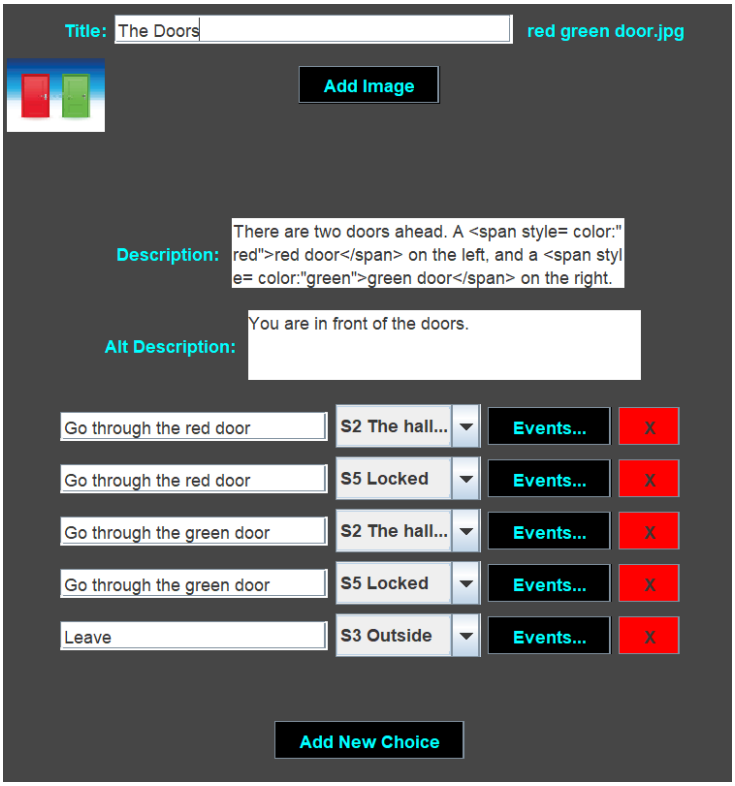

Figure 3: Situation Editor

Descriptions

One of the key features of the Situation Editor is the ability to input descriptions and alternate descriptions for each situation. These descriptions can be formatted using HTML tags and inline styles, allowing the user to change the appearance of the text that will be displayed to the reader in CYAN Reader.

Using HTML tags and inline styles, the user can change the font size, colour, style, and more. For example, the following code would display red text:

This is red text

The alternative description is another important feature of the SituationEditor. This description will be displayed when the reader revisits a situation they have already seen. If no alt description is provided, the initial description will always be displayed. This can be useful for situations where the initial description contains important information that the reader needs to see each time they revisit the situation. The alternate description is useful in situations where the initial description is very long and/or detailed, but the next time the user is there, they only need a brief reminder of where they are. Or for more realistic interactions with characters; If the user returns to a situation describing a gate, and a character who says “I am the gatekeeper, you have earned my favour, here is the key to the gate, now I must rest.”, it does not make sense to have this description display

again, it makes more sense to use an alternative description that says “You are back at the gate, the gatekeeper is resting.”

During the development of CYAN, the functionality of the SituationEditor JText boxes was extensively tested. Numerous stories were written for testing purposes and the JText boxes were used to create engaging descriptions for each situation. The software was also given to friends to test, who were able to find and report bugs that we later fixed. The JText boxes were found by testers to be easy to use, and found to provide a lot of flexibility when it came to formatting descriptions for CYAN Reader.

Figure 4: CYAN Reader

Links

Users have the ability to add new choices to a situation through the Situation Editor. After selecting a situation, the user can add a new choice by clicking on the "Add New Choice" button. The user can then rename the choice by selecting it and typing in a new label in the text field. The JComboBox allows the user to specify which situation the choice leads to. This means that the reader will be taken to the situation that is specified in the JComboBox if the choice is selected. When a situation is selected, a Link object is created between the current situation and the situation specified in the JComboBox. Links are represented by a colour-coded line in the Overview JFrame, which connects the two situations. CYAN uses cyan-colored lines to represent links that run from left to right between situations in different columns, red lines to represent links that run from right to left, and orange or blue lines to represent links that run up or down, respectively, between situations in the same column. It was considered to use arrows to indicate the direction of the links instead of colour coding, but ultimately, the arrowheads take up too much space around situations that have more than a handful of links and end up obscuring lines and other arrowheads.

Events

Events in CYAN, are objects containing a boolean variable that can either be toggled between true and false by choices made by the reader, or that are evaluated to be true or false depending on certain conditions. This feature allows for dynamic storytelling, where the reader's experience is influenced by the choices they make and the events that are triggered. The user can create and define new StoryEvent objects by selecting MenuBar > Edit > Events. Once a new StoryEvent object is created, the user can change the name and specify what kind of event it is. There are three types of events that the user can select: "Simple", "Sequence", or "Player" (which is not yet implemented). The "Simple" event type is just a boolean that the user can specify as true or false by default. The "Sequence" event type is true if a specified selection of other events are true. If the user selects the "Sequence" event type, a button appears next to the event that opens a JWindow. This JWindow allows the user to specify which events to check to determine whether the "Sequence" event is true or false, and whether all of the selected events need to be true or if any of the events are true, it will cause the sequence event to be true, i.e. whether the selected events are evaluated using logical 'and', or logical 'or'.

Figure 5: Event Editor

After defining an event, the user can then go into the Situation Editor for any situation and next to each choice, there will be an "Events" button. This button opens the “Link Editor” (Figure 5) which allows the user to only display the selected choice in CYAN Reader if a particular event is true. Additionally, the user can specify whether selecting this choice triggers any events to become true, false, or to swap between true/false. The ability for a choice to set an event as True or False has proven to be useful. The option to have a choice swap an event between true/false depending on what it is currently, is a feature that seemed vaguely useful, and was reasonably quick and easy to add, but I have not found a practical use for it yet, so it may be removed at a later date.

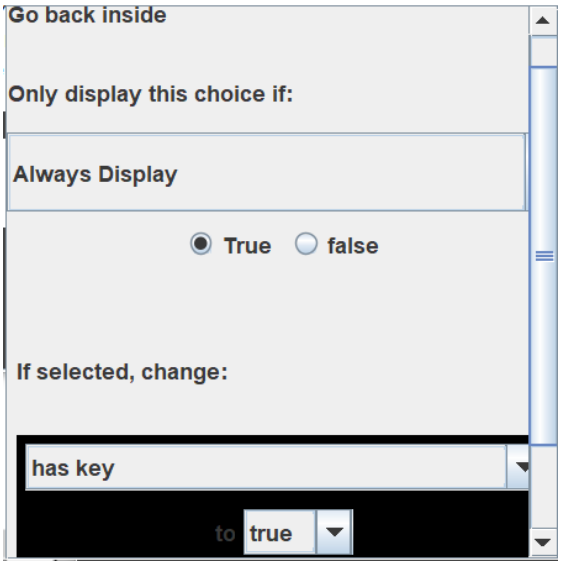

Figure 6: Link Editor

CYAN Writer Summary

CYAN Writer allows a user to create or edit CYOA stories by using situations to present narrative and connecting them with links that represent the choices the reader will have. Stories can be dynamic through the use of events that determine which options are presented to the reader, dependant on previous choices that the reader has made. These constructed and saved CYOA stories are presented to the reader via CYAN Reader.

CYAN Reader

CYAN Reader is responsible for displaying the story to the user. When the user launches CYAN Reader from the Main Menu, it opens a new JFrame that displays two options as Jbuttons; To start a reading a story, or to continue reading a story. The latter will prompt the user to choose a Json file containing Story Progress. If the user chooses to start a new story, they will be prompted to choose a Json file containing the story. They will then be prompted to enter a username that their reading progress will be associated with (Figure 7). The first situation of the story will then be displayed to the reader (Figure 4). CYAN Reader displays the primary or alternative text of the current situation (dependant on whether alt text has been entered for the current situation, and if the reader has visited this situation before), any images associated with it, and the available choices in the form of JButtons. Each choice is linked to a situation, and selecting a choice will cause CYAN Reader to display the situation that is linked to that choice.

CYAN Reader also allows users to exit the story by clicking the 'X' in the top right corner of the JFrame. When the user exits, their story progress is automatically saved under their associated username. This allows users to continue their story where they left off if they come back to CYAN at a later time. The saved progress includes the user's current situation, the player object, which includes all previously visited situations, and the state of any StoryEvent objects that have been triggered.

In addition to displaying the story, CYAN Reader also provides a way for users to interact with the story. For example, if a Situation has any StoryEvent objects associated with it, CYAN Reader will check the state of these events before displaying the choices to the user. If a particular StoryEvent object is not true, CYAN Reader will not display any choices dependant on that event being true.

Figure 7: Name prompt in Reader

Example Story

The following is a brief example of the contents of a typical CYOA game-book like Warlock of Fire Mountain:
Title: The Haunted Mansion

Page 1: You find yourself standing in front of a large, creepy mansion. The gate creaks open, inviting you inside. Do you:
A) Walk through the gate and enter the mansion (turn to page 2)
B) Turn around and leave (turn to page 5)

Page 2: You step inside the mansion and immediately feel a chill run down your spine. There are two doors in front of you. One leads to the left, and the other to the right. Do you:
A) Choose the door on the left (turn to page 3)
B) Choose the door on the right (turn to page 4)

Page 3: The door leads you to a room filled with cobwebs and dust. Suddenly, a ghostly figure appears in front of you! Do you:
A) Confront the ghost (turn to page 6)
B) Run away (turn to page 7)

Page 4: The door leads you to a room with an old, musty book lying on a table. Do you:
A) Open the book and read it (turn to page 8)

B) Ignore the book and continue exploring (turn to page 9)

Page 5: As you leave the mansion, you hear strange whispers and giggling from behind the gate. Do you: A) Turn around and investigate (turn to page 10)
B) Keep walking and leave the area (turn to page 11)

Etc.

One advantage that users of CYAN Writer have is that a new situation's does not necessarily have to be written for every choice, which exponentially increases the number of narrative points you have to include. For example, here is a brief textual representation of a CYOA that could be written in CYAN (The situation each choice leads to is not visible to the reader):

Title: The Illusion of Choice

Situation 1: You have two doors ahead of you, do you go left or right?
Go left (Situation 2)

Go right (Situation 2)

Situation 2: It is dark behind the door, you try to proceed slowly into the dark so your eyes can adjust, but before they do. You find yourself sliding down a steep and slippery ramp! You slide all the way to the bottom and land with a thud. You see there is a dimly lit hallway ahead of you.
Alt Description: You are at the bottom of the ramp, looking down a dimly.
Try to climb back up the ramp (situation 3, only displays if event “Has tried to climb ramp” is false)
Proceed along the hallway (situation 4)

Situation 3: It is no use, the ramp is too steep and too slippery.
Turn Around (Situation 2)

Situation 4: After walking some time you reach the end of the hallway, there is a large lever on an ornate pedestal. The lever has three positions, marked “Win”, “null”, and “Lose”. The lever is currently in the middle “null” position.
Push the lever towards “Win” (Situation 5)
Push the lever towards “Lose” (Situation 6)

Situation 5: You feel euphoric and are instantly teleported to safety. You Win!

Situation 6: The mighty force of the lever's divine wrath causes you to explode instantly. You Lose.

Note that the user can be given the illusion of choice by having two choices that lead to the same situation, by returning to the same situation via another situation but this time with the choice

removed, but can also be given branching, meaningful choices that can become as complex as the writer wishes.

Gson

Serialisation is the process of converting an object or data structure into a format that can be stored or transmitted, such as a file or a network stream. Deserialisation is the opposite process, converting the serialised data back into the original object or data structure.

Gson is a Java library developed by Google that can be used to serialise and deserialise Java objects to and from JSON (JavaScript Object Notation) format[10]. In the context of CYAN, Gson is used to write/read the user-created CYOA story to/from a Json file that can be used by CYAN Writer/Reader.

When the user creates a CYOA story using CYAN, the story is stored in an object-oriented data structure in memory. Gson can be used to convert this data structure into a Json string representation of the CYOA story. This Json string can then be written to a file using standard Java file I/O operations. After the serialisation process, the user-created CYOA story is saved as a Json file. When the user decides to continue working on their CYOA story, the file is loaded using CYAN Writer. This is done through deserialisation, which converts the Json file back into a Java object. The Gson library provides a convenient way to perform deserialisation in Java. When the Json file is loaded, Gson reads the file and recreates the Java objects that were previously serialised. This allows the user to continue editing their CYOA story as if they had never closed CYAN.

Similarly, CYAN Reader uses Gson to write and read the reader's progress through the CYOA story. When CYAN Reader is closed Gson is used to serialise the relevant updated Java objects to a Json string, which is then stored in a separate Json file to the Story Json file it is based on. When the reader resumes the story later, Gson is used to read the Json string from the save file and deserialise it back into a Java object, which can then be used to update the reader's progress in the story. This progress stored in the separate Json file is combined with the data of the original story during the loading process.

Gson is an important tool used in CYAN to write and read CYOA stories in a Json format. By using Gson, CYAN is able to store and retrieve CYOA stories and reader progress in a standard format that can be easily manipulated using standard Java file I/O operations.

Figure 8: Reader Save File

Testing

During the development of CYAN, I tested the software using various methods. One of the primary methods of testing was writing numerous stories of varying complexity to test the functionality of the software. These stories ranged from simple one-situation stories to more complex branching narratives with multiple endings.

As the development progressed, changes to the codebase often resulted in the stories becoming obsolete due to changes in the JSON file format. However, the story could sometimes be recovered following minor changes by loading it in the new version of CYAN Writer and saving it, making it compatible with the new version of CYAN Reader. In future, a class may need to be implemented to convert old Json files to include new variables in serialised classes to prevent obsoleting and therefore loss of user-created stories.

Additionally, I gave the software to some friends to test for me. Their feedback was helpful in identifying bugs that I was able to address in subsequent updates. Their testing also gave me insights that the software was user-friendly and accessible.

Valuable lessons about testing in general were learned. During development, I was trying to add functionality to the CYAN Reader. However, making changes to the code was not yielding any of the expected upon testing CYAN Reader. After many hours, I found that I was editing a similar but different block of code, one that related to loading a story, where I was trying to add functionality to a block of code that relates to starting a new story, and testing that block, not realising that I hadn't made any changes to that part of the code. In future I aim to keep sections of code more distinct, and to more thoroughly check what I am editing before beginning work.

The testing process was an essential part of the development of CYAN. It allowed me to ensure that the software was reliable, bug-free, and user-friendly. The combination of my own testing and the feedback from my friends allowed me to create a robust software application that meets the needs of its users.

Project Planning

Project planning played a critical role in the success of the CYAN project. Throughout the project, Jira was used as a tool to help maintain momentum and track progress. Jira was consulted, reviewed, and updated approximately every week during term time and every two weeks during breaks. This regular monitoring helped to ensure that the project stayed on track and that issues were addressed in a timely manner.

The project was successful in maintaining momentum and adapting to change. Although there were times when progress was slow, the project made steady, meaningful progress throughout the year.

The methodology used proved suitable for the project, and the regular use of Jira helped to keep the project on track.
Time spent on issues in Jira was reported, but an estimated time was not given at the start. As a result, the time tracking report shows 0% accuracy. This lack of accuracy makes it difficult to evaluate the progress of the project and to identify any potential delays or inefficiencies. One of the benefits of giving an estimated time at the start of an issue is that it helps to establish a clear timeline for the task. This timeline can then be used to track progress, identify delays, and adjust the project plan if necessary. By giving an estimated time, workload can also be better managed and tasks can be prioritized accordingly.

There are still issues in CYAN's backlog, but it is evident from the cumulative flow diagram in figure 9 below that the vast majority of reported issues are either completed or in progress at the time of writing.

The methodology used proved to be suitable for the project, and regular monitoring through the use of Jira helped to ensure that progress was maintained. Although there were areas where the project could have been improved, such as the examples above, overall, the project was successful, and provided valuable learning experiences.

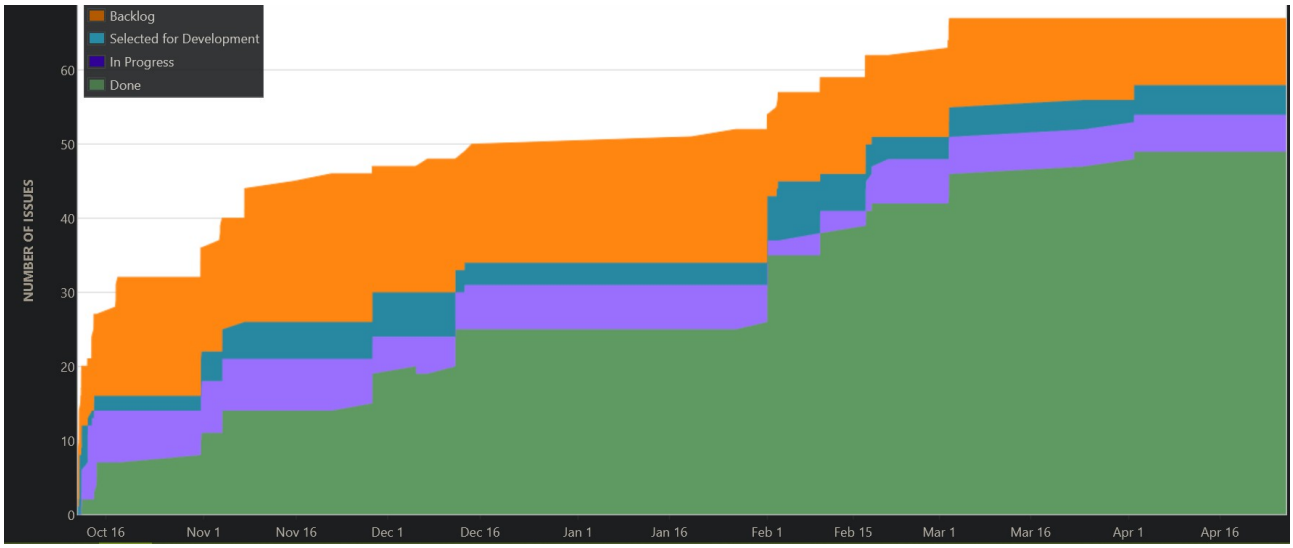

Figure 9: Cumulative Flow Diagram

Improvements

While the CYAN project has achieved a lot in terms of creating a functional interactive fiction engine, there are several potential improvements that could be made in the future.

One potential improvement is the addition of dynamic description text. Currently, each situation has a static description that is displayed to the user when they encounter the situation. However, by using the event system, it would be possible to have the description text change depending on the current state of the story. For example, if the player has obtained a key, the description text for a locked door could change to reflect the fact that the door can now be unlocked. This would add a layer of immersion to the game and make the player feel like their actions are having a real impact on the world.

Another potential improvement is the addition of an inventory system. While users can improvise item collection via the use of events (Having an event called “Player has the key” for example), currently, there is no convenient way to enable the player to accumulate more than one of an item, keep track of items they have picked up, and to use those items to solve puzzles. By implementing an inventory system, players could not only keep track of the items they have acquired, but have a quantity of each item greater than 1, and use them to solve puzzles and progress through the story. This would be particularly useful in the case of currency (Gold pieces or Dollars for example), and consumables such as food or potions/medicine.

A simple combat system could also be implemented. While CYAN is primarily focused on storytelling and exploration, a combat system could add an extra layer of gameplay to the experience. This could involve turn-based combat or simple real-time combat, with the player using a combination of attacks, blocks, and dodges to defeat enemies. This would make player attributes much more meaningful, as the player could be given health that is lowered upon taking damage, strength that determines how much damage the player does, experience that increases the player level when a certain amount is gathered by defeating enemies, etc.

Another potential usage for CYAN is the possibility of adapting CYAN Writer to be used for writing branching dialogue trees for games. Text-based dialogue for NPC's is often handled through a dialogue system, which can involve creating a database or scriptable object that contains the dialogue lines for each character. The dialogue system then handles the display of these lines, often through a text box or speech bubble, and can also include features such as branching dialogue options and quest tracking. Alternatively, some developers may use third-party dialogue tools or plug-ins to simplify the process of creating and managing dialogue in their games. With a few tweaks and possibly writing a separate adaptor for either Unity or Java or similar platforms, CYAN could be adapted to fit the purpose of being a third-party dialogue tool. This would make it possible for game developers to create complex and engaging NPC dialogue for their games with a user-friendly interface. Another similar application could be multiple-choice questionnaires, where the

questions asked depend on the answers to previous questions. The answers could then be saved to a file. The range of potential applications with a small amount of I/O adaptation proves the versatility of the GUI that has been created.

In addition, porting CYAN to Android could also be a valuable improvement. With the increasing popularity of mobile phones and Android's market share, CYAN could reach a wider audience if it were available on mobile devices. This could be achieved through either adapting the current codebase for mobile or developing a separate mobile app that ideally creates/opens Json files in the same way as the desktop app, to allow for cross-compatibility.

Adding the option to export a story as a browser-based story would be another valuable improvement. This would require a JavaScript solution to handle the logical aspect of CYAN. This would allow stories to be shared more easily and accessed from any device with an internet browser. This would also allow for more varied customization of the CYAN Reader interface, witch greater ease by the user.

Another potential improvement for CYAN is the addition of accessibility features for users with impairments. These could include features such as text-to-speech options for visually impaired users, as well as the ability to adjust font size and colour contrast in CYAN Reader and Writer for users with visual impairments. For users with physical impairments, the addition of keyboard shortcuts and voice commands could also be helpful. By implementing accessibility features, CYAN would become a more inclusive platform for all users, regardless of their abilities or impairments. This can help to ensure that everyone has the opportunity to enjoy and engage with the stories created on the platform.

The final improvement I would suggest for the existing software is the introduction of different colour schemes. While CYAN's default dark mode is striking, and high contrast for easy reading, there may be those who prefer a lighter theme or have difficulty viewing the dark background. Therefore, implementing a light theme or customisable themes would greatly enhance CYAN's accessibility and appeal to a wider audience. A light theme option would be easy to implement, and would allow users to switch between the two modes with ease. Customisable themes, on the other hand, would require more development effort, but would provide users with the ability to personalise their CYAN experience to their liking. They could choose from a variety of preset themes or even create their own, selecting colours and other design elements to suit their preferences.

A complementary expansion of CYAN could be the creation of a dedicated online library where users can host and share their CYOA stories with other users. This could be done through a website that allows users to upload their stories and provides a search function for others to browse and download stories that interest them. Not only would this create a community of users who can share and enjoy each other's stories, but it could also serve as a platform for user feedback and ratings, which could help users to improve their own stories and inspire them to create new ones. Additionally, the library could provide a platform for professional writers to showcase their CYOA

stories and potentially monetize them. However, this feature would require a significant amount of development work to implement, as it would need to include features such as user accounts, a search and filtering system, and a secure platform for hosting and downloading stories, as well as constant moderation for offensive material. Nonetheless, such a library could significantly expand the reach and impact of CYAN, and provide an exciting new way for users to engage with the platform.

Overall, the CYAN project has achieved a lot in terms of creating a functional and flexible interactive fiction engine. However, there is always room for improvement, and the addition of dynamic description text, an inventory system, a combat system, and others are just a few of the potential improvements that could be made in the future. These improvements would add extra layers of immersion and gameplay to the experience and would make CYAN an even more engaging and enjoyable platform for interactive storytelling.

Conclusions

At the outset of the project, the overarching goal of CYAN was defined as follows: “A tool that can be used for the creation of "Choose Your Own Adventure"(CYOA)-style games. This will allow a user with no programming experience to create a CYOA game and export it into another format whereby a player can play through the adventure.”. I believe this has been successfully developed and implemented, providing users with an intuitive, flexible, and powerful tool for creating interactive fiction.

A list of desirable features were identified at the beginning of the project [Jira issue: Challenge Week Presentation - C301046-10], and held as goals. Not all were achieved due to time constraints, but many of these features were implemented. The following are goals that I believe were achieved during the project:
“Easy to use, but versatile”: I believe this is achieved due to the lack of previous knowledge required to use CYAN, and the way that CYAN's functionally enables a wide range of expression.
“Save progress”: Serialisation of Java objects into Json files allows progress made to be saved for subsequent CYAN sessions.

“Appropriate Data Structures”: Although the data structures used were revised and refactored many times during the project, through trial and error, I believe that the data structures used at the time of writing are appropriate, and most were chosen with the intention to keep CYAN scaleable and extendable, and to make sure that classes interact with each other efficiently.

“Create a CYOA”: A demo CYOA called “The Hallway” is currently distributed alongside CYAN to anyone who downloads it from my Google Drive for testing purposes. I created many short CYOA stories for testing purposes.

“Visually distinctive and appealing”: I chose a dark theme as the default colour palette due to

personal preference, high contrast is achieved by the titular cyan accents.

There were other goals that were partially or mostly achieved. These are as follows:

“Importing assets”: The importing of images is mostly implemented, but it was intended that assets would also include other media such as background music.

“Player object, Items, and Events”: Player is implemented in the GUI, and this will eventually allow available choices to be dependant on player attributes. Player attributes would also play a part if a combat system were implemented. Items are not implemented. Events are mostly Implemented.
“Tooltips”: There are a few very helpful tooltips, but these are not as prevalent as originally intended.
“Export as Java executable game, and attempt other formats.”: This feature was superseded by CYAN Reader. It makes more sense in terms of storage and simplicity to have an executable CYAN Writer and many, much smaller sized story Json files, than to have many CYAN Writer-sized executables (one for each story). I also did not have time to attempt other formats.
And finally, “Enemies/Combat”, was an intended feature that I did not have time to even attempt, but that I would certainly like to implement during further development.

The use of Jira as a project management tool was particularly effective, allowing for clear tracking of progress and identification of potential issues or risks. The overall methodology proved to be suitable, resulting in steady and meaningful progress over the course of the project. The final result is a system that has achieved most of its goals, and offers a great deal of potential for future development and expansion.

Swing, and open-source library Gson, were both used extensively throughout the development process and are both continual dependencies of CYAN. Both of these technologies brought unique strengths and capabilities to the project, enabling the creation of a powerful and functional system. For example, the use of Swing allowed for the creation of a flexible and intuitive interface, while Gson provided robust storage and retrieval of objects.

One of the key strengths of the CYAN system is its flexibility, which allows users to create interactive fiction in a wide range of different styles and formats. CYAN writer, for example, provides a range of different tools and features for creating complex and engaging narratives, and enables users to manage and customize their stories with ease. Meanwhile, CYAN Reader provides a streamlined and intuitive interface for experiencing these stories, with the ability to save progress and resume later.

Another key strength of the CYAN system is its potential for future development and expansion. The system has been designed with scalability and flexibility in mind, with the ability to add new features and capabilities as needed. This could include the addition of new types of events, Accessibility features , and support for audio content.

In terms of overall achievements and performance, the CYAN project can be considered a success. The system was developed within the time frame, with a high degree of functionality and usability. The use of agile methodologies and project management tools was particularly effective, enabling the project to adapt to newly identified requirements and address potential issues.

Throughout the project, a range of different skills and competencies were required, including programming, design, and project management. These skills were developed and honed over the course of the project, improving my ability to work effectively and efficiently.

Developing CYAN has been a valuable learning experience, demonstrating the importance of clear objectives, effective project management, and careful consideration of user needs and requirements. With its powerful and flexible features, CYAN offers a unique and innovative tool for creating interactive fiction, with potential for continued development and expansion for years to come.

References

[1] Bartle, R., Bateman, C., Falstein, N., Hinn, M., Isbister, K., Lazzaro, N., Ray, S. and Saulter, J., 2009. Beyond Game Design: Nine Steps Toward Creating Better Videogames. Boston, MA: Course, Technology PTR.

[2] S. Lodge, “Chooseco Embarks on Its Own Adventure,” *Publishers Weekly*. [Online]. Available: https://web.archive.org/web/20071009094529/http://www.publishersweekly.com/article/CA640812 6.html. [Accessed: 26-Apr-2023].

[3] Jackson, S. and Livingstone, I., 2012. The Warlock of Firetop Mountain (Fighting Fantasy). Icon Books, Limited.

[4] www.javatpoint.com. 2022. Java Swing Tutorial - javatpoint. [online] Available at: <https://www.javatpoint.com/java-swing> [Accessed 16 September 2022].

[5] “Twine / An open-source tool for telling interactive, nonlinear stories,” *Twine*. [Online]. Available: https://twinery.org/. [Accessed: 26-Apr-2023].

[6] “Inform is a natural-language-based programming language for writers of Interactive Fiction,” *Inform 7*. [Online]. Available: https://ganelson.github.io/inform-website/. [Accessed: 26-Apr-2023].

[7] “Quest - Write Text Adventure Games and Interactive Stories,” *Quest*. [Online]. Available: http://textadventures.co.uk/quest. [Accessed: 26-Apr-2023].

[8] M. Černý *et al.*, “Nonlinear Interactive Stories as an Educational Resource,” *Education Sciences*, vol. 13, no. 1, p. 40, Dec. 2022, doi: 10.3390/educsci13010040. [Online]. Available: http://dx.doi.org/10.3390/educsci13010040 . [Accessed: 26-Apr-2023]

[9]“Inform is a natural-language-based programming language for writers of Interactive Fiction,” *Inform 7*. [Online]. Available: https://ganelson.github.io/inform-website/. [Accessed: 26-Apr-2023].

[10] Google, “Gson: A java serialization/deserialization library,” *GitHub*. [Online]. Available: https://github.com/google/gson. [Accessed: 04-Apr-2023].