CS 496
Winter 2018

Grus: CMD1 ADVENTURE: Final Report

Grus Team Members: Lauren Spiegel, Charlotte Murphy , Yongshi Ye

I. Introduction

The Grus Team has created a text-based Adventure game with a parser. This game is created to follow the theme of popular escape rooms with various themes and challenges. The actions of this adventure game are taking place inside of a medieval castle. The player will begin in one room, and will be able to travel in a chosen direction to other areas. There are 15 different rooms; each room has at least two of its own features. As with many escape games, the goal of the game will be to obtain a key to unlock the final door. A creative storyline has been developed to keep the player interested. There will be hints along the way, along with some secrets. There are ways to take items and items that are needed to get in and out of rooms. The name of the castle is “Darkwood Castle,” and there are some who never made it out.

A user will be able to create a new game, or to continue on with a saved version. The user will find new experiences traveling to each room, and perhaps learn a little bit about the architecture of a medieval castle in the process. The user will be able to use their creativity in traveling from room to room. Instructions to the game encourage the user to take notes, and develop their own map. There will be various choices for the user in some rooms. These choices will lead to a necessary object to have, or have a penalty. The user will find that some rooms they may enter at leisure, and others are locked until they have completed required tasks. An interesting element is the messages along the way, some in the form of scrolls and some from characters. Challenges will be presented as logic puzzles. When all objects have been obtained and used properly, the user will have access to a final exchange with a dragon to obtain a Golden Key, necessary to exit the castle. If the user is clever, there is a specific item to gather that will allow them to bypass many obstacles.

II. Setup

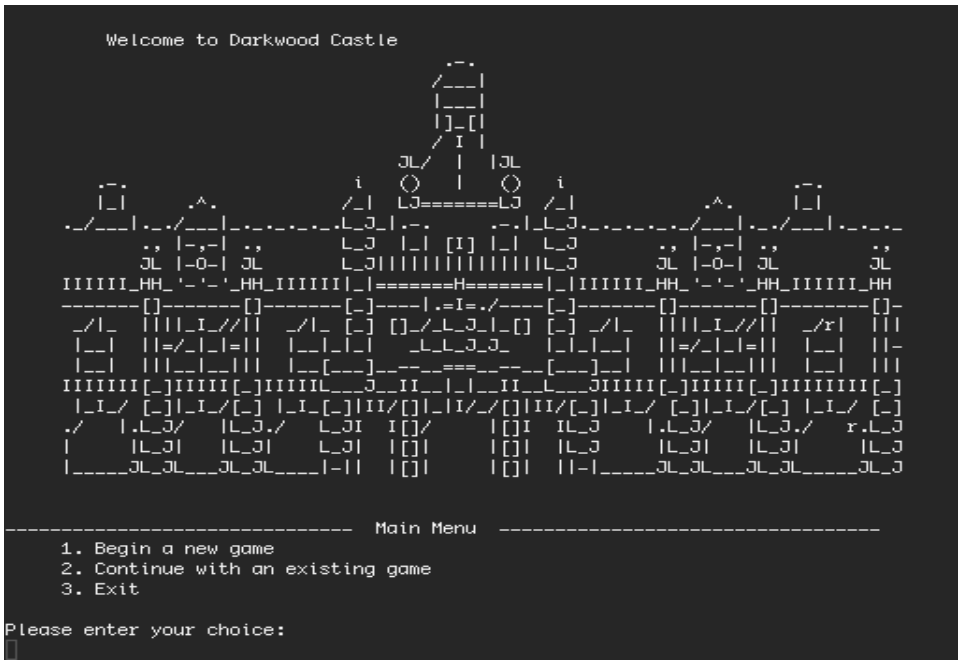
To set-up for the game, compile by typing “make all”. From there, typing “./main” will allow access to the main screen.

★ Commands that can be used for the game:

- **Look**
- **Look at (feature, item)**
- **Map**
- **Go (directions)**
- **Take (items)**
- **Drop (items)**
- **Inventory**
- **Help**
- **Cut**
- **Unlock**
- **Turn**
- **Open**
- **Save**
- **Exit**

★ Instructor Cheat Sheet

To enter the game, type ./main and press enter. Please expand the screen to optimal aesthetics when reading through text. From the home screen, enter 1 to begin a new game. Enter your name and now you are in the first room (Grand Foyer).



Picture 1-1 Main Menu

Room #	Room Name	Feature Name	Feature 1			Feature Name	Feature 2			
			Require	Require	Reveal		Require	Require	Require	Reveal
0	Foyer	painting	-	-	-	chest	-	-	-	blue gem
1	Royal Chambers	chandelier	-	-	candle	bed	-	-	-	-
2	Throne Room	throne	-	-	-	dog	-	-	-	red gem
3	Armory	silver	-	-	sword	knights	text	-	-	armor
4	Solar	chairs	-	-	-	desk	-	-	-	magnifying glass
5	Ballroom	paino	-	-	-	musician	text	candle	-	code 2
6	Courtyard	fog	-	-	-	stone	magnifying glass	-	-	purple gem
7	Garden	hedges	-	-		fountain	-	-	-	green gem
8	Mage's Tower	table	-	-	-	wizard	poison	-	-	
9	Great Hall	walls	-	-	hint	food	-	-	-	white gem
10	Kitchen	pantry	-	-	poison	hearth	-	-	-	-
11	Stables	horses	-	-		loft	-	-	-	yellow gem
12	Treasury	gold	-	-	-	door		-	-	connection to dungeon
13	Dungeon	shackles	-	-	-	dragon	armor	sword	3 gems or rainbow	key
14	Gatehouse	guard	code	or rainbow gem	-		key		OR rainbow gem	Quest complete!

Picture 1-2 Room and Required items

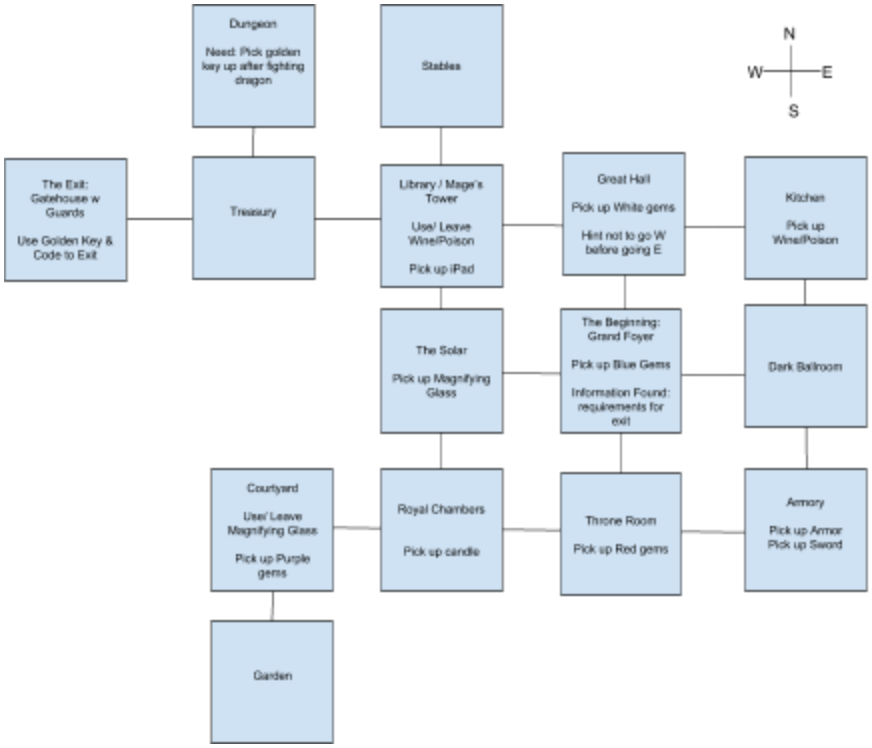
1. **Grand Foyer**
    a. Type “Look at chest”
    b. Type “Take Blue Gem”
    c. Type “Go south” -> To the Throne Room
2. **Throne Room**
    a. Type “Look at dog”
    b. Type “Take red Gem”
    c. Type “Go east” -> To the Armory

3. **Armory**
    a. Type “Look at silver”
    b. Type “Take sword”
    c. Type “Look at knight”
    d. Answer to Puzzle is “e”
    e. Type “Take Armor”
    f. Type “Go west” -> To the Throne Room
    g. Type “Go west” -> To the Royal Chambers
4. **Royal Chambers**
    a. Type “Look at chandelier”
    b. Type “Take candle”
    c. Type “Go North” -> To the Solar
5. **The Solar**
    a. Type “Look at desk”
    b. Type “Take magnifying glass”
    c. Type “Go south” -> To The Royal Chambers
    d. Type “Go west” -> To the Courtyard
6. **Court Yard**
    a. Type “Look at stone”
    b. Type “Take purple gem”
    c. Type “Go east” -> To the Royal Chambers
    d. Type “Go east” -> To the Throne Room
    e. Type “Go east” -> To the Armory
    f. Type “Go north” -> To the Dark BallRoom
7. **Dark BallRoom**
    a. Type “Look at musician”
    b. Answer to Puzzle is “86”
    c. Type “Take Dungeon code”
    d. Type “Go north” -> To the Kitchen
    e. Type “Go west” -> To the Great Hall
    f. Type “Go west” -> To the Library / Mage’s Tower
    g. Type “Go west” -> To the Treasury
    h. Type “Go north” -> To the Dungeon
8. **Dungeon**
    a. Type “Look at Dragon”

b. Type “cut chains”
c. Type “Take Golden Key”
d. Type “Go south” ->To the Treasury
e. Type “Go west” -> To the Gatehouse

9. **Gatehouse**
    a. Type “Look at guard”
    b. Type “unlock Shackles”
    c. Type “Look at windlass”
    d. Type “turn windlass”

Congratulations! You escaped alive! Game Ends, Finished Successfully.



Picture 1-3 Game Map

★ Instructor “Quick Cheat” Cheat Sheet

The Rainbow Gem is a great find for anyone playing the game to navigate through the castle. It also sets the stage for being able to build on the game in the future. For example, if we were to link different castles, the rainbow gem would be a quick pass through one of them.

To enable “cheat status,” use the command “TAKE RAINBOW GEM” in any room to add the rainbow gem to the player’s bag. Cheat status unlocks room navigation, item availability, and

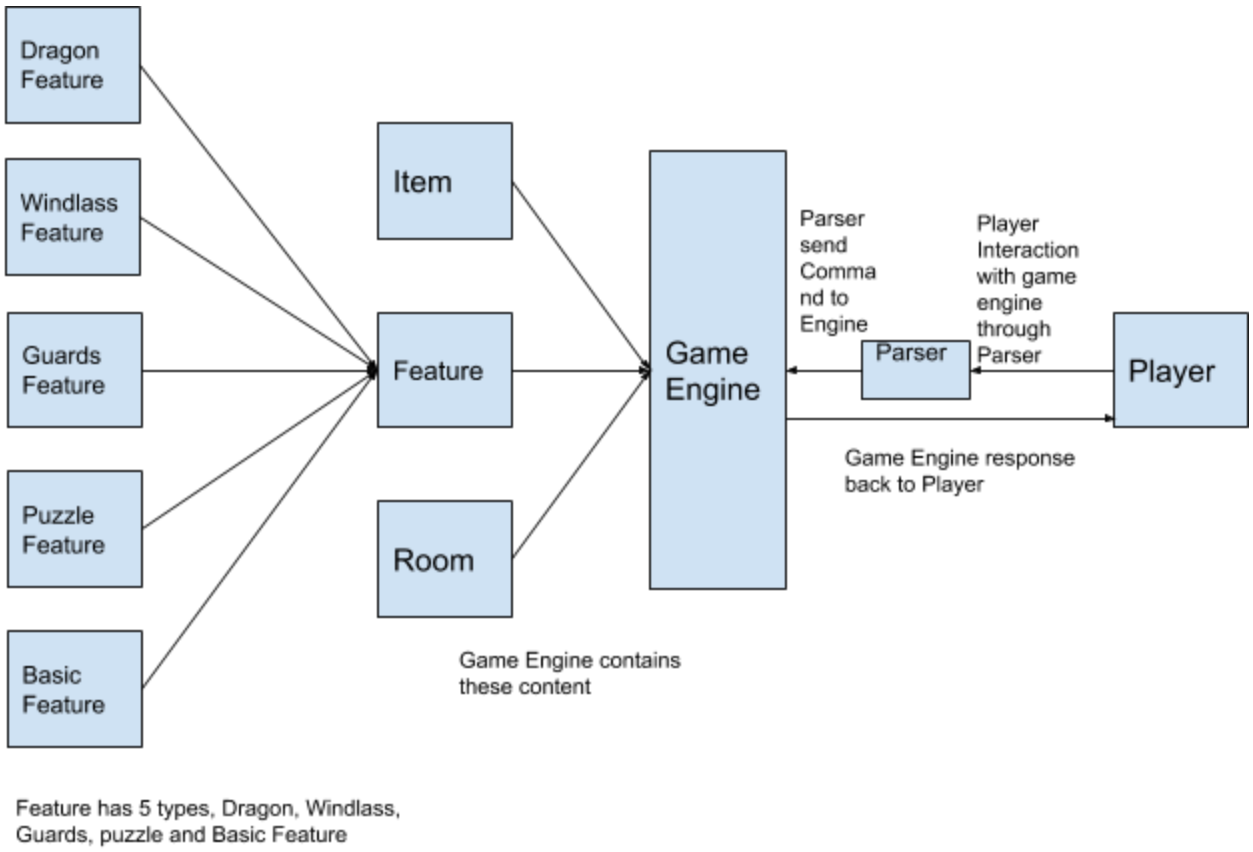
end of game requirements. To disable cheat status at any time, use the command “DROP RAINBOW GEM.”

The rainbow gem gives the player unrestricted access to castle rooms, regardless of valid room connections. For example, if the player is in cheat mode and currently in the Garden, the command “GO TO STABLES” will navigate the player directly to the Stables, by passing intermediary room connections. The rainbow gem also give the player unrestricted access to game items, regardless of what items are available in the current room. For example, if the player is in a room without any available items, the command “TAKE CANDLE” will add the candle to the player’s inventory. And finally, The final game requirements in the Dungeon and Gatehouse are automatically satisfied when the player possesses the rainbow gem.

III. Software and System

Our game is compiled and should run on the OSU server flip.engr.oregonstate.edu.

Software Hierarchy:



Game Engine:

The game engine is responsible for loading and saving games. It also integrates with the parser to interact with players through command. It controls the location of player, items in the player's bag or in the room. For feature, it checks whether users have required items, correct answers to the questions and correct actions to examine the feature.

Parser:

The parser takes in user input and translates it to understandable commands for our game engine. It can map different actions to one command. For example, take, pick up and grab are all translated to the [TAKE] command. It also ensures that the object the player has entered is something related to the game. The parser was flexible enough that all group members could contribute and add in new commands as ideas came about. The parser was tested for all needs

of the text-based adventure game. It is designed such that it can be built upon if we were ever to expand the game.

Feature:

Feature is where player can find items. Different features have different requirements. Features may require items, like you need to have at least 3 gems for dragon. Also it might require text. For example, you need to solve a puzzle for knights to get an Armor. Last but not least, some features requires actions to finish the feature like cut chains for dragon. Player needs to figure out what is needed and to be able to examine feature.

Room:

The game has 15 rooms, all rooms have different features, items and descriptions. Each room features a unique long description and short description that the player has access to. Each room give a glimpse into a medieval castle and generates a flow to connect all 15 rooms. Some rooms require items from other areas in the game in order to proceed.

Item:

Item has its name and description. All of the items is hidden in different rooms, player needs to examine features to able to take item in their inventory. Rainbow gem is a hidden item which is available to pick in all rooms. But player can't see it. If they are able to pick it up, the game changed to cheat status. It will change back to normal status if player dropped it.

Player:

Player has its location and inventory. They can take items available in current locations, and drop items from their bag. Dropped item will be available in current room, they can come back and take it again. They can also move to different direction or rooms if that direction or room is accessible. For features require items, player needs to have items in their bag to examine the feature.

IV. Development Tools

*Language: C++
*Library: Boost Library http://www.boost.org/
*Tools: GitHub Repository
We are using GitHub as our version control tools. Each of us submits commits through Github so that we can easily track our code. Over 300 commits were made on Github throughout the design of the program. Several branches were created exclusively for designer experimentation and commits, while others chose to primarily commit directly. GitHub proved to work extremely well for all of the program requirements. As a group we were able to even contribute to different text files as instructions amongst ourselves through GitHub during the design process. We did run into a few complications along the way during times in which we were all editing at the same time and stepping closely into what another was working on.

Team members used IDEs locally to write the program, such as Visual Studio, and Sublime Text Editor.

The program was extensively tested on the flip engineering servers. Testing on the flip engineering servers allowed for consistency among all group members.

V. Individual Team Member Responsibilities

All group members were able to participate in the majority of conference calls throughout the duration of the project, and were able to contribute meaningful ideas and report unfavorable findings to all members. As a whole, the group stayed on track and in alignment to the initial anticipated project deadlines along the way.

Lauren Spiegel

Lauren put extensive work into the parser and creating an editable and testable files that all group members would be able to contribute to as inspiration to the game took place. Countless hours of research and planning were put into the text development for the game and it helped to bridge and simplify ideas for the game throughout. Many steps were taken along the way to enhance visual aesthetics of the game, and to simplify processes for usability purposes. As with all members of the group, testing was performed in all considered ways to ensure a unique and seamless play. Lauren did all initial testing of the parser and original implementation into the main game files. Lists and charts were created to better organize the content of commands and their relationship to items. Original Plan: Data Format Developer: creates the data files, writes the text loading and parsing software

Charlotte Murphy

Charlotte did a great job setting up the initial platform for GitHub and worked extensively as a planner and communicator for meetings and group interactions. Extensive work was put into the structure of the castle rooms to ensure that everything flowed smoothly. She designed a class outline and diagram and used it as a reference to develop the data structures, including many of the “get” and “set” functions.

As the data format developer, Charlotte designed and implemented a system for storing, loading and saving the game data. This includes the game data file structure, the format and content of the text files, methods for reading the files and translating the data to program variables, and methods for saving the current game state of all game objects. She also contributed the instructions functionality and text.

Charlotte assisted with game engine development, testing, and implemented features, without requirements, revealing items and adding item to rooms. She also assisted with display formatting by adjusting print statements, writing a function to display long text with line appropriate line breaks for the game window size. Charlotte contributed to the parser’s functionality by building its command and object “vocabulary” and assisting with command implementation.

Yongshi Ye

Yongshi acted as a very strong contributor to the game engine and inspired many ideas throughout the design process. Yongshi worked on the Game Engine, like how the game started, how to quit the game and check if the player has won the game. She worked on

methods processing player's commands. She implemented most of command functions like take, drop, inventory, go, etc,. She implemented examine feature function which makes sure player can interact with feature properly. She implemented Free Dragon, Move Windlass and Unlock Shackles features as well as Puzzles which can interact with user to get the answer as a challenge. Yongshi ran and tested the game to make sure it works properly and fixed bugs when walking through the game. Yongshi contributed a great head start to writing the final report.

VI. Deviations from Initial Plan

There were several deviations from the initial plan. Many of the initial command words were used, but we found that several needed to be added to make the game more interesting and to create more layers of activity for the player. Many of the initial items were used, but one or two were added and one or two were removed. These changes came about when discussing in further detail the best approaches to usability design and game function.

We wanted the game to have a good flow, and came up with some creative ways to tie things together. One of the biggest changes that came about were the character interactions. For example, in the initial plan it was considered that the player would fight a main character, the dragon, towards the end of the game. It was later decided that it would be fun and unique to send the player on a hunt for different colored gems and to use those gems to turn an otherwise dangerous and unfriendly character into a useful friend to be able to exit.

A challenging aspect for any new player of the game will be to tie different pieces of the game together, and to pick up hints when they are given. The player will need to connect the dots on their own in several cases. This is a deviation from our original plan in that we had considered creating a more simple layout to some of the rooms but then creating more complicated games and mazes throughout. We decided to “level” the complexity of the game throughout, and to add in helpful surprises for the user instead.

Many new ideas were implemented during the creation stages with deeper thought such as a rainbow gem that is secretly available throughout the game, should the player find the hint to grab it. A map was also a deviation from the initial plan. At first, there was no intention to have a map at all, but towards the final planning stages we found it to be useful for younger players should they choose to use it.

VII. Conclusion

In conclusion, all group members were able to input to the style of the game in ways that were not foreseen at the beginning of the project. One of the biggest challenges that came up consistently was re-working ideas and forming plans to getting the ideas to fit into the game as a whole. There were a few areas that we thought would be easy to implement initially, but more creative and flexible solutions were found along the way.

Through testing, it was found that this game is playable for users of many ages, and has the potential to challenge the average player. Certain features and hints were implemented to make the game more usable for younger crowds but still optional for those that wanted more of a challenge.

Group members were able to have a fluid hands on approach where anyone could work on different aspects that were agreed upon as positive and meaningful changes to the game. The game was designed such that new ideas can always be added, new characters introduced, and the territory expanded. An interesting idea if we were to pursue this game further on our own time, would be to create an easy, medium or hard level for user choice and perhaps even have different castles to visit. The project gave us a great foundation to see the endless possibilities that arise out of planning an interactive game.