PROG50016 Project 1
Alessandro Profenna
November 1, 2019

Text-Based Adventure Game
’Bearly’ Made It Out

Planning

I’m pretty happy with the end result. I believe I scoped out the size of the game relatively well, however some of the implementation details caused challenges and took longer than I thought. Balancing between the tool and this did take some time to plan, and caused me to actually start coding later than I should have. I think I may have tried to let the tool more than was actually required for this game. It was an iterative process that often caused me to redo a lot.

Overview

You’re a bear trying to escape a zoo. See the map included in this submission for more details on the layout.

As a bear sneaking around a zoo at night, you traverse different animal enclosures and the pathways between them. You start of in the bear enclosure and then go off to search for 5 pieces of clothing to create a disguise.
Once all items are found, go back to the bear enclosure, type ‘GO TO SLEEP’. The bear will go to sleep, wake up in the morning, put on all of the clothes, and walk out of the zoo unnoticed. You win!

You can search, attack, travel (north south, east, west), pickup and drop items. If there is a guard in your location, they will tranquilize you if you perform any actions before attacking them. This is like “dying’ in the game, and you respawn back in your enclosure but your items are dropped. Once a guard is attacked, they go unconscious, and must be hidden. If you don’t hide the body, the next time you visit the location, you’ll get tranquilized by other guards who found the body.

Item Checks
There are some items and directions that are unlocked only if the player and player’s location have specific items. For example, one of the items are unlocked if a banana is in that location.

Saving and Loading

You can new/save/load at anytime.

There are two files to care about here, both in prog50016.Project1.AP/TextAdventureGame/Build/Assets/SaveData/ .
Backups of the these files are included with the submission in the Backup folder, in case the they are overwritten and you want them back.

NEW -> Loads a new game from newGameData.game.json

SAVE -> Saves data to gameSaveData.json
This file is overwritten every time, so there is only one save file.

LOAD-> Loads the default game data from newGameData.game.json and then loads the save data from gameSaveData.json

So,
newGameData.game.json
gameSaveData.json

are used for loading and saving. A third file, newGameData.editor.json, is for loading into the tool editor.

Database

Every time the player uses an action or gets tranquilized, it is recorded. Then when the user quits, this data is stored as a new record in the TextBasedGameAnalytics database in proj50016.Project1.AP/TextAdventureGame/Build/Assets/Database/

List of commands:

Go [Direction]
Pickup/Take [x]
Attack [x]
Drop [x]
Search [x]
Hide [x]

Look
Inventory
Help
New -> new game
Save -> saves game
Load -> loads game save file
Q -> quits game

Example set of commands to win the game:
(The second column is a continuation of the first; I just wanted to fit this on one page)

```
new
go south
go west
search large pile of feathers
take crowbar
go east
go east
go north
attack guard
hide guard
go north
go north
attack guard
hide guard
go east
search trashcan
take mouse
go west
go south
go south
go east
drop mouse
go east
attack guard
search guard
take key card
hide guard
go north
go west
go south
go south
go west
attack guard
search guard
take boots
hide guard
go west
go south
go south
attack guard
hide guard
search ice cream stand
take banana
go east
search locker
take trench coat
go west
take sunglasses
go east
go north
go north
go east
go east
go east
take bucket of fish
go west
go west
go west
go west
go north
drop banana
take fedora
go south
go east
go south
go east
drop bucket of fish
take rubber gloves
go west
go west
go north
go to sleep
```

Incomplete vs what was asked:

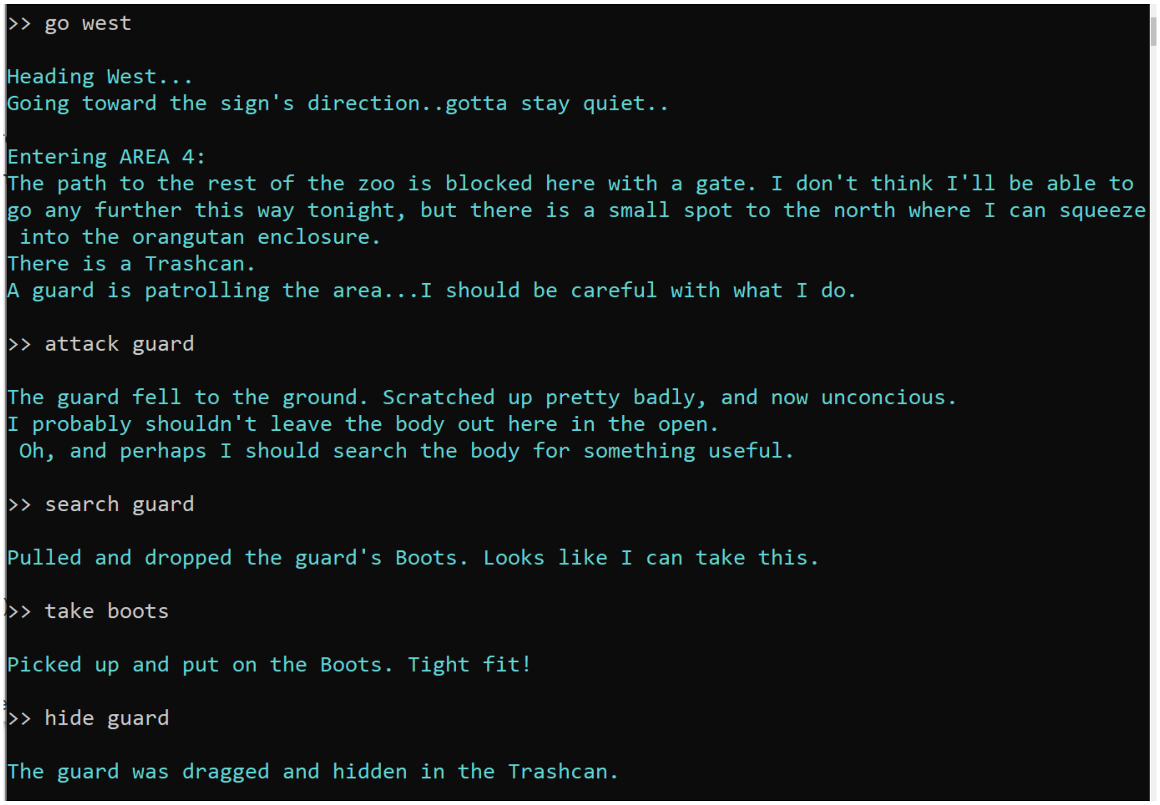
I believe all requirements have been satisfied. I did have some commands like NEW and LOOK set in an if-else chain, but player action commands are in a function pointer map. Not sure if that matters. Loading and saving work, however there could be possible bugs depending on the json file data, and when the loading/saving is done.

Wishlist (vs bugs and what I wanted to do)

- I wanted separate classes for room type and animals for more unique interactions
- I used the ItemCheck class to trigger events in the game. It worked as expected but ended up limiting me in many ways. It is great because it lets the designer using the data tool to implement event triggers, but was limiting in some ways. This also led to some events triggering at weird times.
- Better text parsing and contextualizing, multiple word parsing like “Hide guard in Trashcan”
- I actually used my data editor tool to create the newGameData json file that is loaded into the game. However, the tool encodes some special characters like quotation marks that are printed weirdly in the console. I would have liked to fix this. Because of this, I had to manually edit the special characters in the newGameData file that I submitted.

There may be more bugs that further testing will find.

Screenshot



```
>> go west

Heading West...
Going toward the sign's direction..gotta stay quiet..

Entering AREA 4:
The path to the rest of the zoo is blocked here with a gate. I don't think I'll be able to
go any further this way tonight, but there is a small spot to the north where I can squeeze
 into the orangutan enclosure.
There is a Trashcan.
A guard is patrolling the area...I should be careful with what I do.

>> attack guard

The guard fell to the ground. Scratched up pretty badly, and now unconcious.
I probably shouldn't leave the body out here in the open.
 Oh, and perhaps I should search the body for something useful.

>> search guard

Pulled and dropped the guard's Boots. Looks like I can take this.

>> take boots

Picked up and put on the Boots. Tight fit!

>> hide guard

The guard was dragged and hidden in the Trashcan.
```