Guardian

A text-based adventure

Team 1 - working separately, together - Jude

Day 1 - Creative Writing (Not a completely useless degree after all)

The first day consisted of coming up with a main premise for the game, we agreed on a guardian angel idea watching over “our human”. Jude & Jack took on the roles of co creative directors and discussed characters, plot points and how the game will be structured, making sure with the rest of the team are on board, and taking on any inputs or suggestions. Ershela put together the flowchart with input from the team over Zoom while Jude updated the Trello accordingly. We discussed various puzzle ideas while keeping in mind the limited time, and how we would implement a final boss fight with multiple endings.

Trello / Flowchart - Jude

We were asked to use Trello and a flowchart to plan out our game. On day one we set up both, this helped make sure everyone knew what their tasks were and how they would run into each other.

After day one both of them were only used as a reminder but not updated as we agreed our tasks verbally over chat which was constantly open and available.

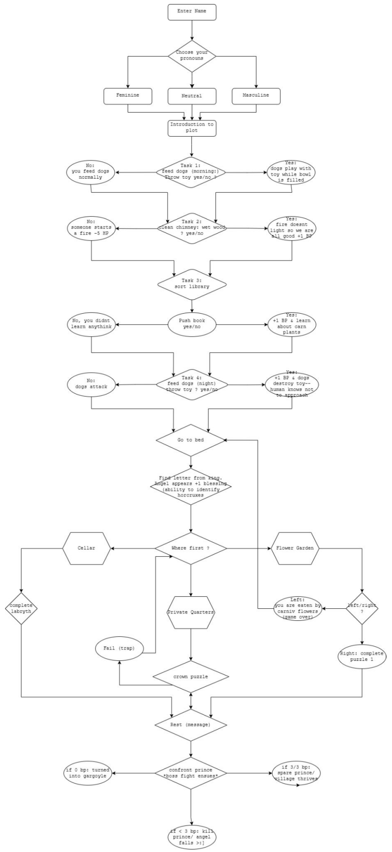
I think one thing that could have made them better specifically towards programming would be a way to share our code so we could all work on it at once as towards the end it was a constant game of resending the file to one person, waiting and resending.

Frontend v backend development

Frontend development is essentially programming everything that a user will interact with - such as visuals, mechanics, loading screens, etc. In short, if you can see it or interact with it, it is frontend.

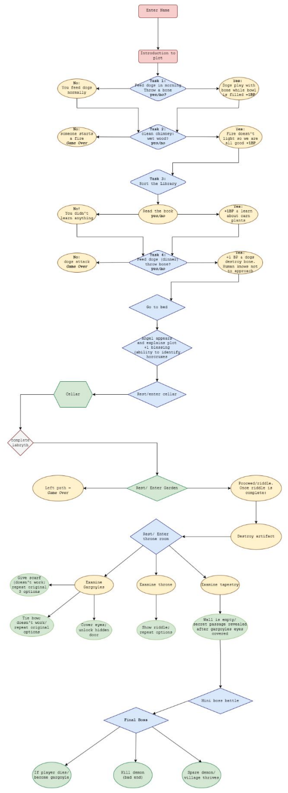
Backend development is everything that goes on behind the scenes that you don’t see - such as the logic and the code that makes everything work. Variables, methods - anything that is not directly shown to the user falls under backend development.

Think of it as a clocktower - the clock face itself is what would be our frontend, whereas the gears and the machines running it inside is what would be our backend development!



Enter Name
Choose your pronouns
Feminine
Neutral
Masculine
Introduction to plot
Task 1: feed dogs (morning:) Throw toy yes/no ?
No: you feed dogs normally
Yes: dogs play with toy while bowl is filled
Task 2: clean chimney: wet wood ? yes/no
No: someone starts a fire -5 HP
Yes: fire doesnt light so we are all good +1 BP
Task 3: sort library
Push book yes/no
No, you didnt learn anythink
Yes: +1 BP & learn about carn plants
Task 4: feed dogs (night) throw toy ? yes/no
No: dogs attack
Yes: +1 BP & dogs destroy toy-- human knows not to approach
Go to bed
Find letter from king. Angel appears +1 blessing (ability to identify horcruxes
Where first ?
Cellar
Flower Garden
complete labryth
Private Quarters
Left: you are eaten by carniv flowers (game over)
left/right ?
Fail (trap)
Right: complete puzzle 1
crown puzzle
Rest (message)
confront prince *boss fight ensues*
if 0 bp: turned into gargoyle
if 3/3 bp: spare prince/ village thrives
if < 3 bp: kill prince/ angel falls >:]

GUARDIAN



Enter Name
Introduction to plot
Task 1: Feed dogs in morning Throw a bone yes/no?
No: You feed dogs normally
Yes: Dogs play with bone while bowl is filled +1BP
Task 2: clean chimney: wet wood? yes/no
No: someone starts a fire Game Over
Yes: Fire doesn't light so we are all good +1BP
Task 3: Sort the Library
Read the book yes/no
No! You didn't learn anything
Yes: +1BP & learn about carn plants
Task 4: Feed dogs (dinner) throw bone? yes/no
No: dogs attack Game Over
Yes: +1 BP & dogs destroy bone. Human knows not to approach
Go to bed
Angel appears and explains plot +1 blessing (ability to identify horcruxes
Rest/enter cellar
Cellar
complete labryth
Rest/ Enter Garden
Left path - Game Over
Proceed/riddle. Once riddle is complete:
Destroy artifact
Rest/ Enter throne room
Examine Gargoyles
Examine throne
Examine tapestry
Give scarf (doesn't work; repeat original 3 options
Tie bow; doesn't work/ repeat original options
Cover eyes; unlock hidden door
Show riddle; repeat options
Wall is empty/ secret passage revealed after gargoyles eyes covered
Mini boss battle
Final Boss
If player dies/ become gargoyle
Kill demon (bad end)
Spare demon/ village thrives

Day 1

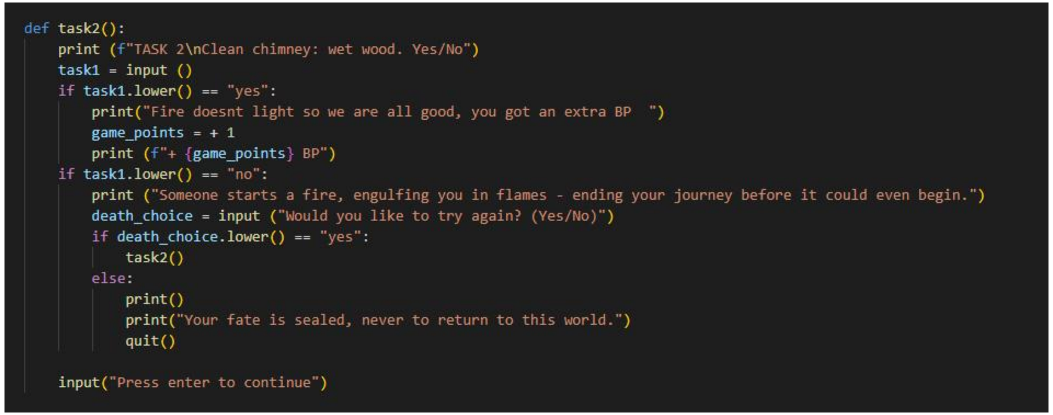
My section of the game consists of 4 different tasks:

- Feed the dogs in the morning
- Clean the chimney
- Sort out the library
- Feed the dogs in the evening

My focus was mainly on writing the code and trying to use all the knowledge gained in the first week of the course.

For my code, I used mainly:

- If statements
- Else statements
- Functions



```
def task2():
    print (f"TASK 2\nClean chimney: wet wood. Yes/No")
    task1 = input ()
    if task1.lower() == "yes":
        print("Fire doesnt light so we are all good, you got an extra BP  ")
        game_points = + 1
        print (f"+ {game_points} BP")
    if task1.lower() == "no":
        print ("Someone starts a fire, engulfing you in flames - ending your journey before it could even begin.")
        death_choice = input ("Would you like to try again? (Yes/No)")
        if death_choice.lower() == "yes":
            task2()
        else:
            print()
            print("Your fate is sealed, never to return to this world.")
            quit()

    input("Press enter to continue")
```

The Labyrinth (Zahir)

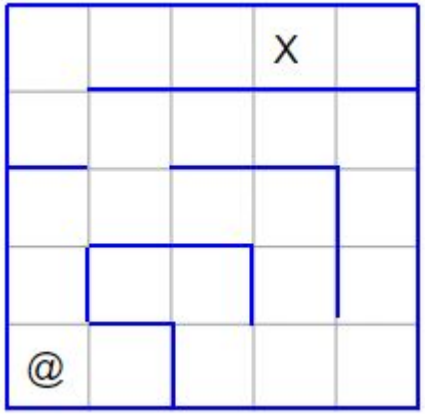
Labyrinth:

a complicated irregular network of passages or paths in which it is difficult to find one's way; a maze.

The Labyrinth (Zahir)

Our labyrinth is a simple 5x5 maze.
Movement is carried out using the compass directions
[N] - to move North
[W] - to move West
[E] - to move East
[S] - to move South

Additionally, we allow the user to quit any time without reason.
[X] - to Exit



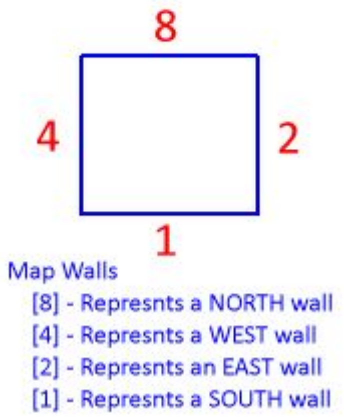
The Labyrinth (Zahir)

Our labyrinth walls.
The labyrinth is made up of many rooms.
Each room is designed by using the format on the right.

Simply add the numbers given, for each wall in your room,
And that's all there is to it!

Any side of the room which does NOT have a wall, would have a doorway.

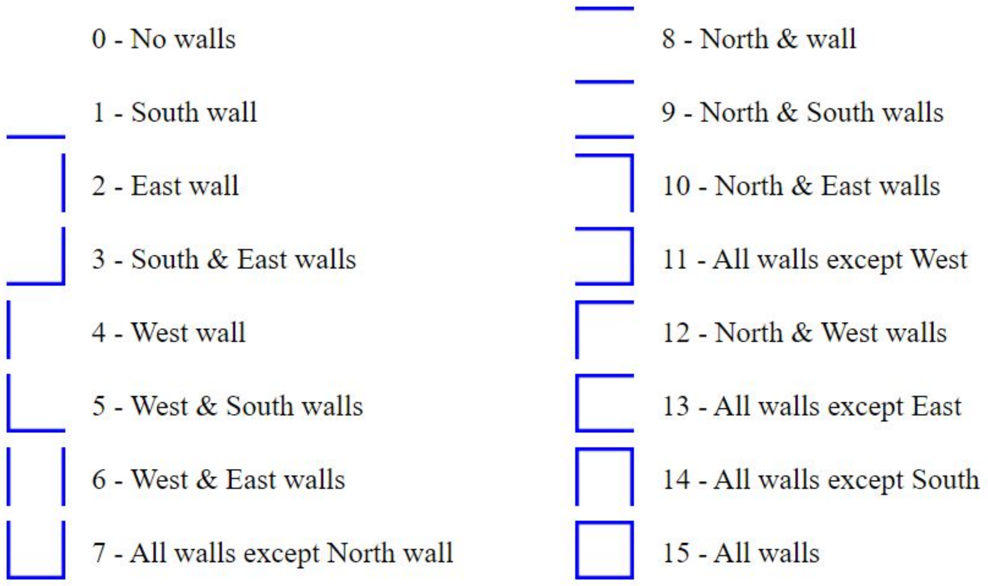
It's easier to understand when you view the next slide.


8
4
2
1
Map Walls
[8] - Represnts a NORTH wall
[4] - Represnts a WEST wall
[2] - Represnts an EAST wall
[1] - Represnts a SOUTH wall

The Labyrinth (Zahir)

Room Variants.
Taking into account everything in the last slide.

it is possible to create 16 different types of rooms



0 - No walls
1 - South wall
2 - East wall
3 - South & East walls
4 - West wall
5 - West & South walls
6 - West & East walls
7 - All walls except North wall
8 - North & wall
9 - North & South walls
10 - North & East walls
11 - All walls except West
12 - North & West walls
13 - All walls except East
14 - All walls except South
15 - All walls

The Labyrinth (Zahir)

The labyrinth has been written in such a way that the maze can be edited in the python ide so as to give a different maze layout. Maybe at a later time a maze editor could be created?.

This would not affect the navigation system,, and not break the labyrinth section of the game.

Maybe the additional feature of maze scaling could also be included.

Off course the idea's were many and time was little, and it really helps to design all the separate sections so there is less need for bug fixing or major changes part-way through.

Communicating with team members can also help, fix issues and give a new insight from a different angle, even simply passing idea's amongst the team can help in ways that you could not foresee / expect..

The Labyrinth (Zahir)

Coding Used

Clearing the screen - I imported the 'OS' module to clear the screen
Time Delays - I imported the 'Time' module for where I wanted to insert delays
The Maze - I used a list to store the maze design
Navigation - I used functions to navigate through the maze taking user-input for directional choices
Room Layout - I used functions for laying out the room & assessing which type of room it is
UserInput- I used functions to validate the user input
Maze Layout - I used loops of room layout functions to do this
Menu - I used a dictionary for use as the part of menu system

Additional Information

Although I did the maze section on my own, it would not have been possible without the help and advice of each of the members in the team - Thank You Team 1!!

Jack - The Poison Garden

I had decided to code two main levels of the game, the poison garden and the throne room/castle tower. Since we had decided originally to only have one combat-based level, I decided to implement puzzles that would differ per level.
For this level, I included a choice of which path to take, one leading to a game over and one leading to a riddle.
I knew that Ershela was coding Day 1 of the game, and suggested an option to have a hint be revealed in the library- Ershela coded this but I went back later to add the hints for my level which comes later.
I mostly coded this level using if statements, inputs for choices and of course defining my own functions for readability and also to allow progression through each stage.

This is arguably a simple level however as I did most of the story writing in the game, rewrote lots of dialogue in sections that others coded and coded another level, I created it using my time wisely by not making it overly complicated. However, I think the straightforwardness of this level is made up for by the implemented hint in earlier code and riddle that I wrote myself, which is quite hard without the hint ! There's also risk of a game over on this level, so if the player ignored the previous hint, they might not make it out so easily…

Jack - The Castle Tower

I wrote this level immediately after I finished working on my poison garden. I wanted to make this one much more interactive, meaning this time a sequence of events must happen in a certain order in order to unlock a secret passage. I did this by creating menus & inputs based on if statements.

After my immersive storytelling, the player is given the option to examine three parts of the room: the throne (containing a riddle as subtle instructions for the puzzle), the tapestry (the image on it relating to the puzzle, and it also covers what at first seems to be a regular stone wall…) and the final place the player can examine is the gargoyle statues, which offer the options to tie a bow, give a scarf or cover their eyes… (SPOILERS:) if the player covers the eyes of the gargoyles and then chooses to examine the tapestry again, this time the wall will have disappeared, revealing a secret passage that leads to a miniboss battle that I asked Jude to implement.

I coded the various choices using if statements and functions, and then used “while” along with boolean values to decide whether or not there will be a wall or a secret passage.

Story writing, puzzles, riddles… interactive AND immersive !

Jack - ACII Artist & Creative Director

It was also my sole responsibility to implement and colour all ascii art. I shared my screen and/or asked the team for their input for each ASCII art, to make sure we all like it and want to use it. I created them using functions so that the code is tidier and more readable, and also added colour to all dialogue spoken by our angel character.
I think i'm justified in referring to myself as creative director as I wrote all of the storytelling on the poison garden and tower levels, I wrote all of the storytelling on day 1 (including the narration, game introduction, choices & consequences and the rest at the end of the day)- though it was Ershela who wrote the code– I just added the storytelling after she had created the options.
Though I must give credit where credit is due- my angelic co-director Jude helped me come up with the plot and characters of the game. We snowballed ideas back and forth, whittling them down together while asking our other team members for their inputs and ensuring they are happy with our decisions. I also slightly edited the storytelling on Zahir's maze, but his maze dialogue was already pretty good. I just needed to tweak the wording and add a beginning and end to the level in regards to storytelling (entering after a rest & destroying the artifact) I'd also like to mention that i took on a general leadership role, allocating tasks, checking in on my fellow team members and their tasks(making sure everyone is on track), regularly ensuring my team's happiness with how things were going & helping alisha put each person's code in the correct order.

Jack - What didn’t I write ?

This is an easier question to answer than “What did i write?”, since I pretty much wrote the whole story/dialogue of the game besides the miniboss fight, final boss battle and endings. Everything else besides some dialogue in Zahir’s maze was entirely written by me. I did also edit Zahir’s maze as it was originally made of “#” symbols, so i swapped it for “`█” symbols, which made it more readable and presentable`

However I did make sure to get Zahir and Ershela’s consent to edit their code for the sake of storytelling, as this was the plan from the very beginning.

I’ve taken pride in my riddles and storytelling, so I hope you enjoy it and find it enticingly immersive ! >:] There are plenty of morbid ends, but it’s nicely contrasted by bits of humour here and there (especially in Judes code :P) After writing and adding all of the story and ASCII art, i ran & tested the game several more times to ensure I hadn’t broken it, that all formatting was correct (i had to add a lot of f {name}’s) and that there were no spelling errors to be found.

I also wrote our flow chart (with the whole team on Zoom to talk me through it together), and then handed it over to Ershela to tidy up the arrows as I struggled to get them aligned correctly– \such a finicky program !

Jack - Team Leader

Though me and Jude agreed to be co-creative directors, I think it’s fair to say that I was the main team leader as it was me who allocated tasks, let everyone know what they were doing each day, updated Alisha after their absence and caught them up with where we were at in terms of development and giving them the task of debugging and formatting.
I also took the initiative to talk Ershela through updating the flowchart and editing the story / updating dialogue in other sections.
I believe it was my initiative that led team decisions (while asking for everyone’s opinions/input) and allocated tasks.

Jack, Alisha, Zahir- Many Amendments

The three of us tested and ran the program over and over and over to identify all manner of mistakes and work on them. For jack this meant fixing dialogue, for zahir this meant adding a typewriting effect (credit to jude for writing the code) to spoken dialogue (with jacks help identifying it) and Alisha fixed all major and minor coding bugs / missing code that was needed for every function to work.

Boss Battle & Multiple endings

One Winged Angel plays in the background

Like a Boss - Jude

I immediately snatched up doing the boss battle as I already had an idea of how to do it (or thought I did).
My first attempt simply consisted of the player chipping away at the bosses health. This included a randomised hit strength and a boss that was happy to stand there and be killed.
Then I thought… What if it was turn based? I did a quick google search and decided, that may be above my skill level.
One sleep later, it was turn based.
I basically copied our attacks and turned them around for the boss, not sure why google made it seem so complicated.
After managing turn-based I focused on making it less repetitive, I made the boss have a randomised line for its turn instead of saying, "It grinned unnaturally wide and said, 'My turn'" because why would it be grinning that much anyway? I also managed to figure out how to make the game end once the boss is defeated using some good ol' boolean statements to turn is_playing to false.

Of course I also had to make an input for the name as one didn't exist yet for the overall project.

Testing, testing, testing - Jude

Ah yes, the bane of existence, the error I got when ending the boss fight.
Back in the olden days I had a simple line for surviving the boss fight and losing it. My code didn't return an error but it had one anyway. When the boss would die, it would print another attack despite being dead, it would also give both ending outcomes. Later it turned out I was using >= where I should be using just > and == where I should be using just =.

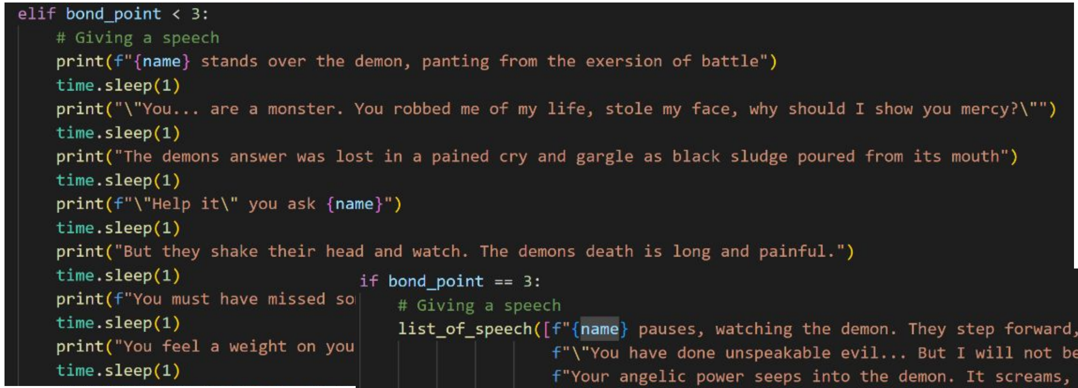
But to find that I inserted two cheat codes into the fight. So if you haven't played the game, stop reading here! One cheat code which enabled me to test the bad ending was allowing the input "die" to have the player character die instantly and check that the rest of the code runs correctly. Then to test the good ending (and later medium when I included bond points) I made the cheat code "kill" which does 30 damage to the boss.

Think this was particularly interesting as most code is problem solving and it can be especially difficult to debug something when there are multiple minute errors. I'm glad I managed to at least make it faster with the cheat codes.

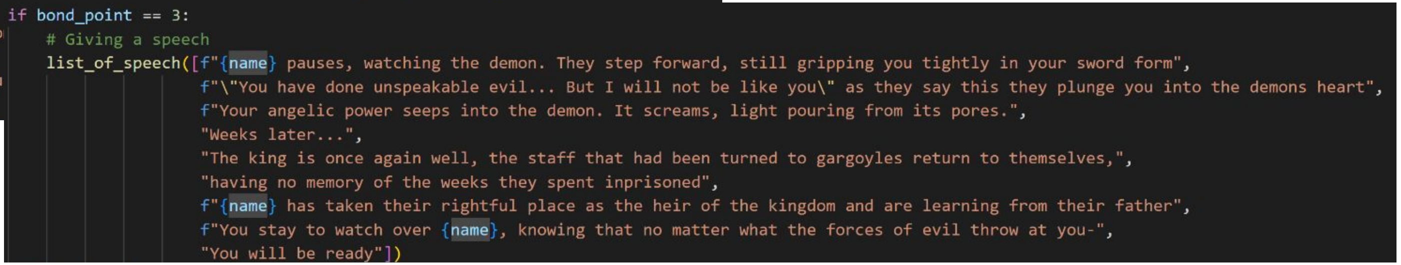
Yes. They are still in the game. Yes you can technically skip the whole fight. Please don't, I worked hard on that.

Multiple Endings - Jude

The multiple endings were fairly easy in terms of coding though when I was working alone I had to set up my own bond points since I didn't have the previous times they would have been awarded throughout the game. The way I initially had the endings written out was long and clunky. Refactoring helped a lot!



```python
elif bond_point < 3:
    # Giving a speech
    print(f"{name} stands over the demon, panting from the exersion of battle")
    time.sleep(1)
    print("\"You... are a monster. You robbed me of my life, stole my face, why should I show you mercy?\"")
    time.sleep(1)
    print("The demons answer was lost in a pained cry and gargle as black sludge poured from its mouth")
    time.sleep(1)
    print(f"\"Help it\" you ask {name}")
    time.sleep(1)
    print("But they shake their head and watch. The demons death is long and painful.")
    time.sleep(1)
    print(f"You must have missed so
    time.sleep(1)
    print("You feel a weight on you
    time.sleep(1)
```

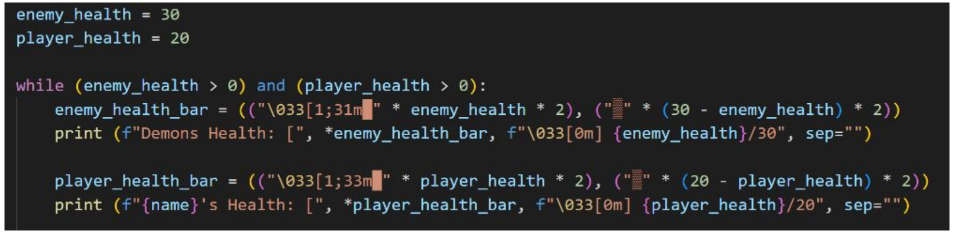


```python
if bond_point == 3:
    # Giving a speech
    list_of_speech([f"{name} pauses, watching the demon. They step forward, still gripping you tightly in your sword form",
                    f"\"You have done unspeakable evil... But I will not be like you\" as they say this they plunge you into the demons heart",
                    f"Your angelic power seeps into the demon. It screams, light pouring from its pores.",
                    "Weeks later...",
                    "The king is once again well, the staff that had been turned to gargoyles return to themselves,",
                    "having no memory of the weeks they spent inprisoned",
                    f"{name} has taken their rightful place as the heir of the kingdom and are learning from their father",
                    f"You stay to watch over {name}, knowing that no matter what the forces of evil throw at you-",
                    "You will be ready"])
```

As you can see, making it a function made it much more readable :)
It also made it reusable for everyone else!

Refactoring - Jude

I was showing off my code to all my friends and one of them is a professional programmer, he saw it and decided it was time to teach me about refactoring. This is basically making your code 1. More readable. And 2. Reusable. So let’s take the health

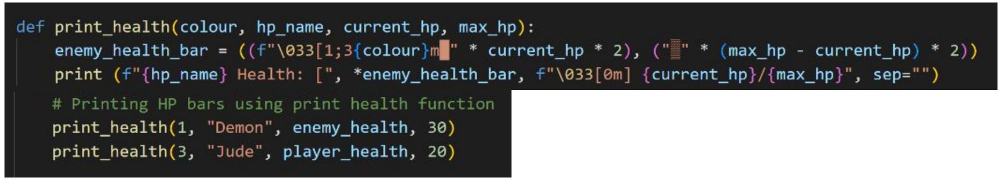


```python
enemy_health = 30
player_health = 20

while (enemy_health > 0) and (player_health > 0):
    enemy_health_bar = (("\033[1;31m█" * enemy_health * 2), ("▒" * (30 - enemy_health) * 2))
    print (f"Demons Health: [", *enemy_health_bar, f"\033[0m] {enemy_health}/30", sep="")

    player_health_bar = (("\033[1;33m█" * player_health * 2), ("▒" * (20 - player_health) * 2))
    print (f"{name}'s Health: [", *player_health_bar, f"\033[0m] {player_health}/20", sep="")
```

This is the original health bars for both the player and the demon.

As you can see, this has unnecessary repetition.

The refactored version is a function which can be called. This means the text INSIDE the while loop is more concise. It also means to make another health bar, I’d just have to call the function again.

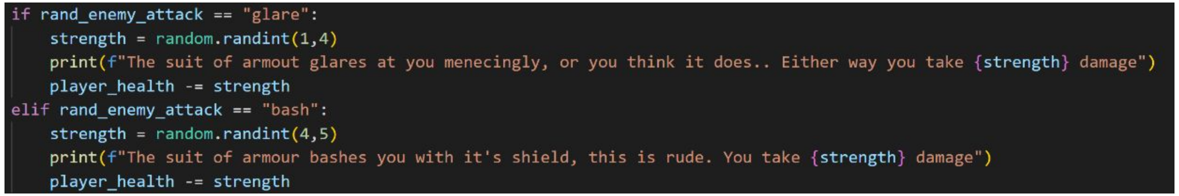


```python
def print_health(colour, hp_name, current_hp, max_hp):
    enemy_health_bar = ((f"\033[1;3{colour}m█" * current_hp * 2), ("▒" * (max_hp - current_hp) * 2))
    print (f"{hp_name} Health: [", *enemy_health_bar, f"\033[0m] {current_hp}/{max_hp}", sep="")
    # Printing HP bars using print health function
    print_health(1, "Demon", enemy_health, 30)
    print_health(3, "Jude", player_health, 20)
```

Mini boss - Jude

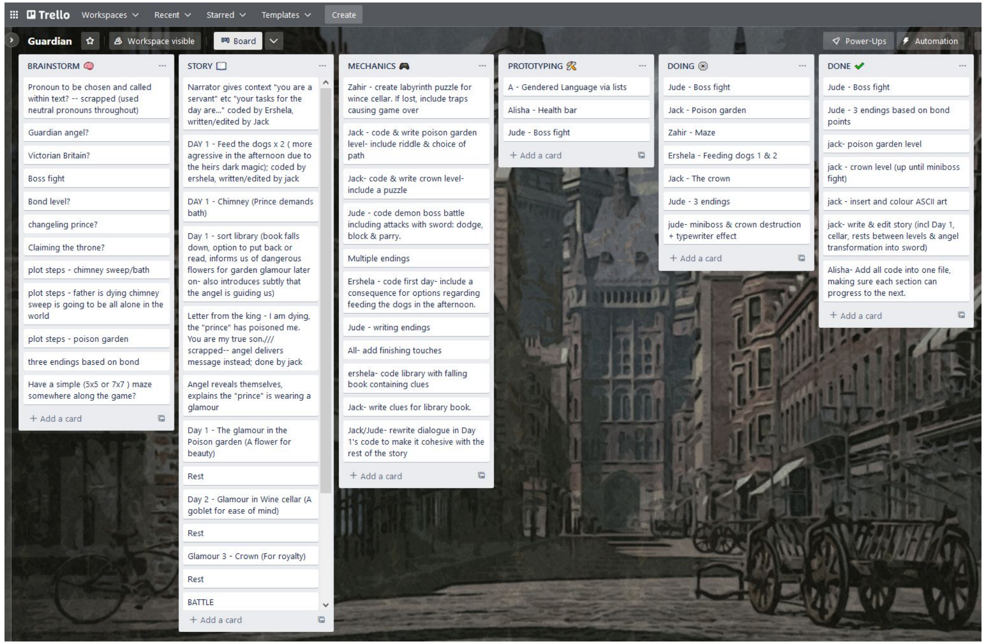
Jack wanted to put a mini boss inside the tower so I offered to code it as I was familiar with my code and could quickly simplify it and shorten it to create the mini-boss. When he was describing it he said he wanted it to be quite humorous so I included jokes inside the script including attacks such as bonk and rust (which implies the suit of armour is rusty which hurts its feelings).

I had quite a bit of fun with this just coming up with some jokes but what was really pleasing to see was how easy it was to make a new enemy and adapt it.



```
if rand_enemy_attack == "glare":
    strength = random.randint(1,4)
    print(f"The suit of armout glares at you menecingly, or you think it does.. Either way you take {strength} damage")
    player_health -= strength
elif rand_enemy_attack == "bash":
    strength = random.randint(4,5)
    print(f"The suit of armour bashes you with it's shield, this is rude. You take {strength} damage")
    player_health -= strength
```

Alisha - Bug Fixing, Formatting & Compiling



BRAINSTORM

- Pronoun to be chosen and called within text? -- scrapped (used neutral pronouns throughout)
- Guardian angel?
- Victorian Britain?
- Boss fight
- Bond level?
- changeling prince?
- Claiming the throne?
- plot steps - chimney sweep/bath
- plot steps - father is dying chimney sweep is going to be all alone in the world
- plot steps - poison garden
- three endings based on bond
- Have a simple (5x5 or 7x7) maze somewhere along the game?
- + Add a card

STORY

- Narrator gives context "you are a servant" etc "your tasks for the day are..." coded by Ershela, written/edited by Jack
- DAY 1 - Feed the dogs x 2 (more agressive in the afternoon due to the heirs dark magic); coded by ershela, written/edited by jack
- DAY 1 - Chimney (Prince demands bath)
- Day 1 - sort library (book falls down, option to put back or read, informs us of dangerous flowers for garden glamour later on- also introduces subtly that the angel is guiding us)
- Letter from the king - I am dying, the "prince" has poisoned me. You are my true son./// scrapped-- angel delivers message instead; done by jack
- Angel reveals themselves, explains the "prince" is wearing a glamour
- Day 1 - The glamour in the Poison garden (A flower for beauty)
- Rest
- Day 2 - Glamour in Wine cellar (A goblet for ease of mind)
- Rest
- Glamour 3 - Crown (For royalty)
- Rest
- BATTLE
- + Add a card

MECHANICS

- Zahir - create labyrinth puzzle for wince cellar. If lost, include traps causing game over
- Jack - code & write poison garden level- include riddle & choice of path
- Jack- code & write crown level- include a puzzle
- Jude - code demon boss battle including attacks with sword: dodge, block & parry.
- Multiple endings
- Ershela - code first day- include a consequence for options regarding feeding the dogs in the afternoon.
- Jude - writing endings
- All- add finishing touches
- ershela- code library with falling book containing clues
- Jack- write clues for library book.
- Jack/Jude- rewrite dialogue in Day 1's code to make it cohesive with the rest of the story
- + Add a card

PROTOTYPING

- A - Gendered Language via lists
- Alisha - Health bar
- Jude - Boss fight
- + Add a card

DOING

- Jude - Boss fight
- Jack - Poison garden
- Zahir - Maze
- Ershela - Feeding dogs 1 & 2
- Jack - The crown
- Jude - 3 endings
- jude- miniboss & crown destruction + typewriter effect
- + Add a card

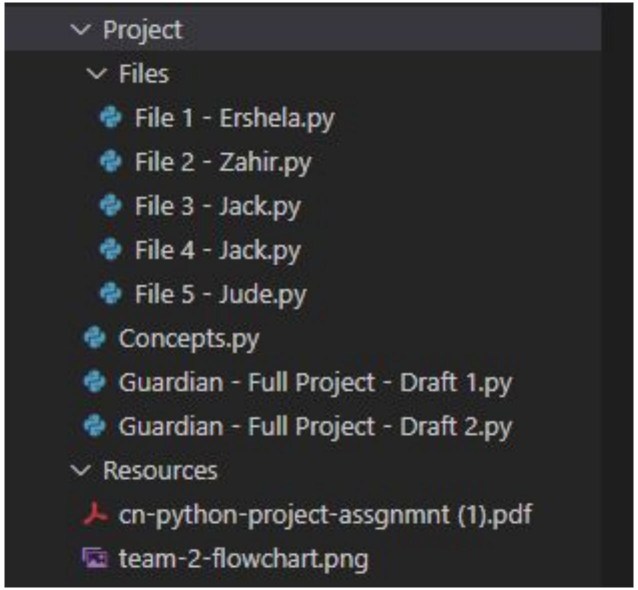
DONE ✓

- Jude - Boss fight
- Jude - 3 endings based on bond points
- jack- poison garden level
- jack - crown level (up until miniboss fight)
- jack - insert and colour ASCII art
- jack- write & edit story (incl Day 1, cellar, rests between levels & angel transformation into sword)
- Alisha- Add all code into one file, making sure each section can progress to the next.
- + Add a card

I was the one to initially create the Trello board that we used throughout the project which enabled us to take our collaborative efforts to the next level and improve the efficiency between us.

As I was unfortunately away for both Tuesday and Wednesday, I wasn't able to sink my teeth into it as much as the other members. Despite that, I was checking in with the others over the discord server when I got the chance, and upon my return I was able to collect each of their individual pieces, and create one single, large script that combines all the others' efforts into one. This raised plenty of bugs and of course - parts of one script now influenced the other upon running the code.

Alisha - Bug Fixing, Formatting & Compiling

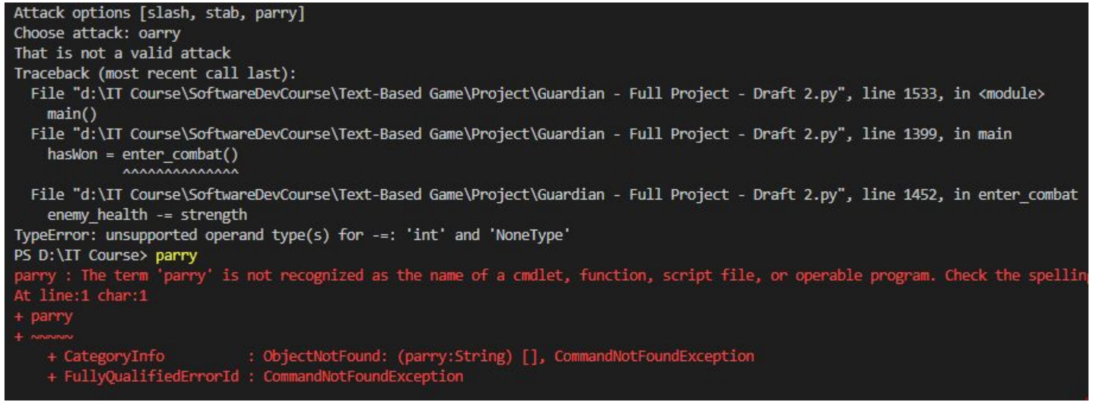


```
⌄ Project
  ⌄ Files
      File 1 - Ershela.py
      File 2 - Zahir.py
      File 3 - Jack.py
      File 4 - Jack.py
      File 5 - Jude.py
    Concepts.py
    Guardian - Full Project - Draft 1.py
    Guardian - Full Project - Draft 2.py
  ⌄ Resources
    cn-python-project-assgnmnt (1).pdf
    team-2-flowchart.png
```

It was my job to resolve all these issues and mold the game into a form that tells one full adventure. I also changed a lot of the formatting to make sure there was more parity between each of our coding and writing styles. So, in short - most of my work isn’t visible, as it’s behind the scenes - but it was crucial to getting the game to work and flow the way it does.

Alisha - Bug Fixing, Formatting & Compiling

Of course, another large part of that is testing it - and that's likely the longest and most difficult part to run through, as it requires running through it repeatedly. One of the hardest parts of that is knocking down a bug and having two more appear. I did the main bulk of it, and it was a fairly repetitive, but honestly I feel like I do quite well with that kind of work, so I was more than happy to do it!



```
Attack options [slash, stab, parry]
Choose attack: oarry
That is not a valid attack
Traceback (most recent call last):
  File "d:\IT Course\SoftwareDevCourse\Text-Based Game\Project\Guardian - Full Project - Draft 2.py", line 1533, in <module>
    main()
  File "d:\IT Course\SoftwareDevCourse\Text-Based Game\Project\Guardian - Full Project - Draft 2.py", line 1399, in main
    hasWon = enter_combat()
             ^^^^^^^^^^^^^^
  File "d:\IT Course\SoftwareDevCourse\Text-Based Game\Project\Guardian - Full Project - Draft 2.py", line 1452, in enter_combat
    enemy_health -= strength
TypeError: unsupported operand type(s) for -=: 'int' and 'NoneType'
PS D:\IT Course> parry
parry : The term 'parry' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spellin
At line:1 char:1
+ parry
+ ~~~~~
    + CategoryInfo          : ObjectNotFound: (parry:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```