

REPORT

최종 보고서



강 의 : 오픈소스 SW기여
교 수 : 황 두성
소속학과: 소프트웨어학과
학 번 : 32214185
이 름 : 정재현
마 감 : 04

목 차

- 1. 작품 제작 배경 및 목표 설정**
- 2. 제작에 사용된 코드 간략 설명 및 다른 사항**
- 3. 전체 시스템 UI**
- 4. 개발과정**
- 5. 주요 실행 화면**
- 6. 목표 달성률**
- 7. 개선점 및 수정 사안**
- 8. 감상**

1. 작품 제작 배경 및 목표 설정

작품 제작 배경

Chat Gpt가 유행하기 시작하면서 한 때 유행했던 '서울 2033'와 같은 텍스트 어드벤처 게임을 chat gpt를 이용하여 기존의 텍스트 어드벤처 게임 장르의 한계를 깨고 싶었습니다. 기존 텍스트 어드벤처 게임이나 일부 다른 게임들 같은 경우 단점이 있는데, 이는 바로 정해진 스크립트 내에서 행할 수 밖에 없다는 것입니다. 자유도 부분쪽에서 특히 텍스트를 선택하기만 하는 텍스트 어드벤처 장르가 제한적입니다. 비록 이 선택지를 추가하거나 예상치 못한 반전을 주는 등의 효과로 반복 플레이를 유도하기는 하지만, 이도 꾸준한 활동을 유지하기 힘들다고 봤습니다. 따라서 저는 chat gpt를 이용하여 스토리를 제작하고 진행하는 것이 더욱 효과적일 거라고 판단하였고, 이를 게임으로 만들어보기로 결심하였습니다.

Chat gpt를 이용하여 생성한 내용들은 같은 주제라고 할지라도 매번 반복할 때마다 다른 지 스토리로 전개하기 때문에 플레이어에게 새로운 경험을 줄 수 있다고 생각했습니다. 기존의 텍스트 어드벤처의 게임에서는 개발자가 정해진 선택지 내에서만 행할 수 있는 자유로운 것 같지만, 자유의 제한이 있는 게임이었습니다. 그러나 이를 Chat gpt를 이용하게 된다면, 기존의 선택지 이외에도 스스로 선택지를 작성할 수 있기 때문에 이는 긍정적인 효과를 불러일으킬 수 있다고 판단했습니다.

기존 목표 설정

처음 세웠던 목표는 다음과 같습니다.

1. 유료 버전이 아닌 무료 API 키 이용 가능 달러가 5 달러 또는 18 달러로 나와왔는데, 이는 최소 20 번 이상의 게임을 진행할 수 있습니다. 그리하여 무료 API 키를 이용하여 해당 애플리케이션을 이용할 수 있도록 하였습니다.
2. 기존의 어느 텍스트 어드벤처 게임이든, 아무리 뛰어난 이야기더라도 선택지에서는 정해져있기에 수동적인 느낌을 많이 주고 있습니다. 그리하여 고정된 이야기보다 사용자들이 직접 선택지를 입력할 수 있도록 하였습니다.
3. 글을 읽을 때 글자만 있으면 전달력이 떨어질 수 있기에, 그림을 삽입하는데 이를 착안하여 텍스트 어드벤처 게임 스토리를 진행할 때 현재 진행중인 스토리에 맞게 이미지를 생성할 수 있도록 하였습니다.
4. 수동적인 느낌을 최소화하기 위해 직접 게임에서의 등장인물과 대화하는 느낌을 주고 싶습니다. 그리하여 게임 진행중 만나는 등장인물들과 대화하여 실제 그 세계에 있는 것 같은 느낌을 주도록 하고 싶습니다.
5. 진행이 어렵다면 진행을 쉽게, 진행이 쉽다면 진행을 어렵게하는 변칙적인 변수와 게임에서 직접 앱 화면을 컨트롤하여 진행중인 사안의 노선을 틀어버리는 변수를 추가하였습니다.
6. 인벤토리 시스템을 이용하여 어떤 물건이든 생성하지 못하도록 제한하고, 그 인벤토리 아이템을 사용하는 것을 한 턴으로 소비시켜 게임의 진행 난이도를 조금 높이고 싶습니다.
7. 난이도의 시스템 등 여러 변수를 줄 수 있는 요소들을 집어넣어 게임을 진행하고 싶습니다.

2. 제작에 사용된 코드 간략 설명 및 다른 사항

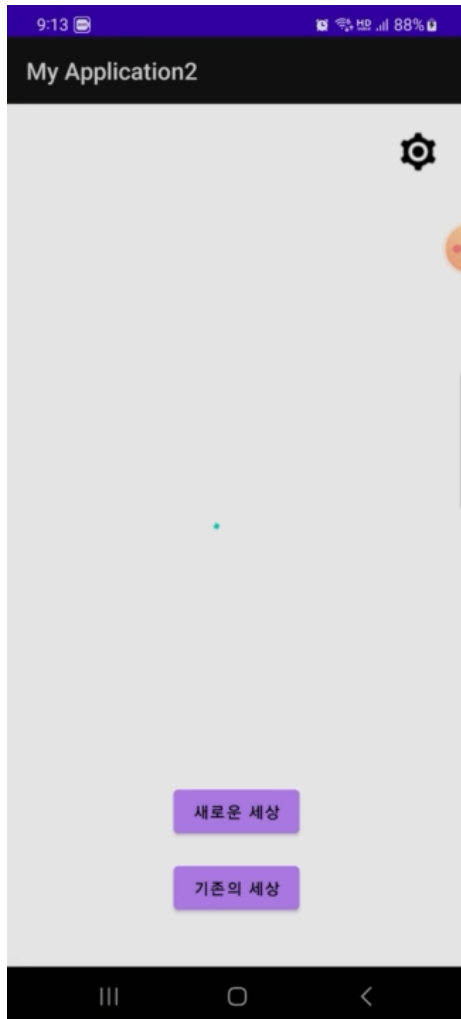
내용 설명

메인 코드 설명만 첨부하도록 하겠습니다.

전체적으로 사용된 클래스와 큰 함수는 다음과 같습니다.

세부적인 내용이나 기타 자잘한 함수들은 모두 깃허브 코드나

개발자 사용 가이드에서 확인하실 수 있습니다.

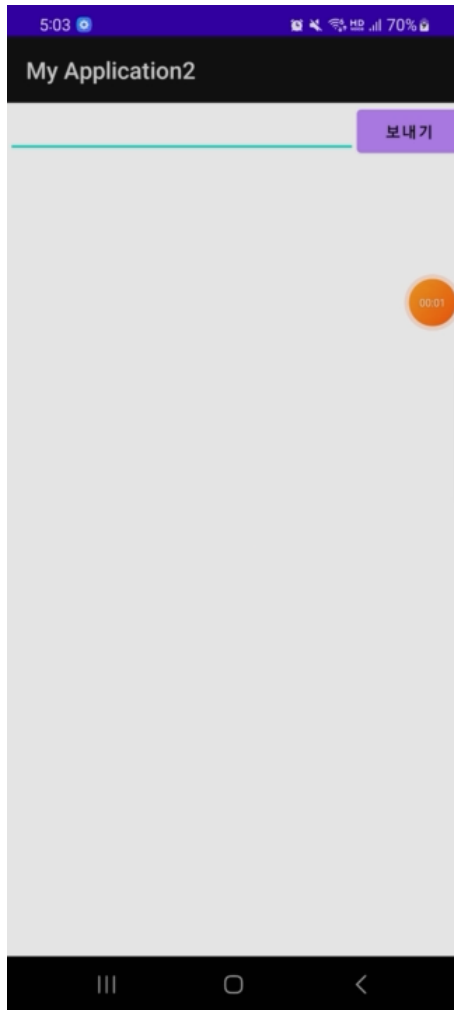


메인 화면의 모습으로 스토리가 생성되었는지의 여부를 판별해주고, 게임의 시작 진행을 도와주는 함수로 MainActivity.kt 클래스가 있습니다. 이 클래스 안에는 기본적으로 chaquopy가 오류가 나서 사용이 안 될 시에 주석처리를 해제하여 사용할 수 있는 okhttp를 이용하여 chat gpt를 호출할 수 있는 부분이 있습니다. 또한 showDialog 함수로 api 키의 유효성을 검사하고 클립보드에 복사한 내용을 붙여넣을 수 있는 부분이 있습니다. 그리고 이야기의 생성 저장을 판별하여 그에 맞춘 배경 이미지를 생성하는 스크립트 부분과 UI 화면을 다르게 지정하는 부분이 작성되어 있습니다.

데이터 검사 부분으로는 delimiter로 <!@!>을 설정해 놓았습니다.

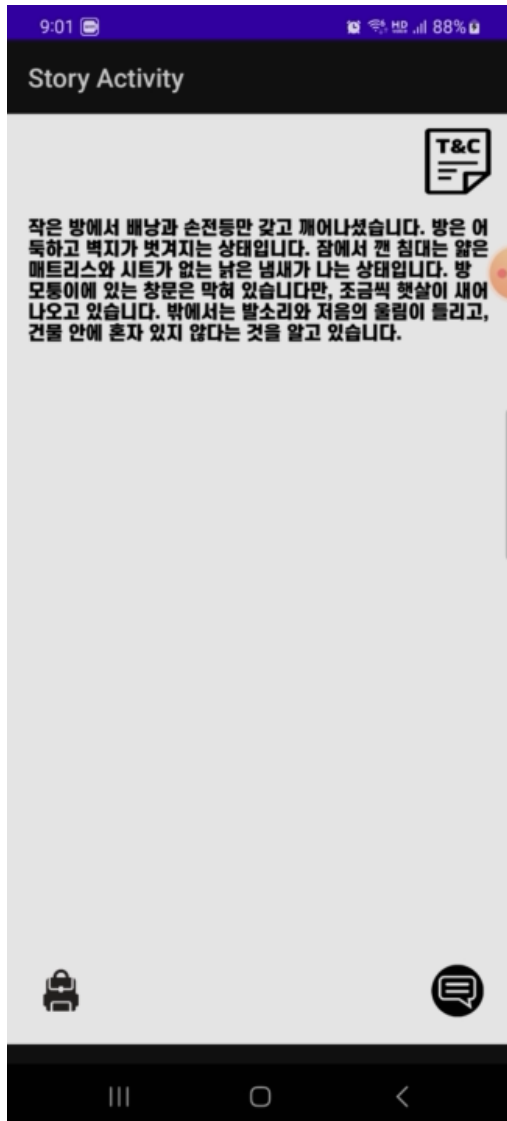
저장된 데이터 순서가 APIkey 값과 현재 진행중인 스토리 상황인 Messages, 그리고 화면에 출력된 값들의 모임인 Attributes 가 있습니다. 가장 처음의 API 키가 유효한 것인지 다시 한 번 검사해보고, Messages 값과 Attributes 값이 없으면 게임의 진행을 할 수 없기에 이들이 유효한지 검사하는 코드가 있습니다. 만약 그렇게 하여 생성된 스토리가 유효하지

않거나 진행할 수 없는 상태라면, data.txt를 지우고 초기화되었음을 알리는 문구를 알려주도록 하였습니다. 그 반대로 사용 가능한 스토리가 생성되었다면 기존의 세상을 이어서 하는 선택지와 새롭게 이야기를 만들 수 있는 선택지가 있도록 하였습니다.



다음으로는 원하는 주제와 그에 맞는 배경 이미지까지 생성하여 게임을 진행하는 `MakeStoryFragment.kt` 클래스가 있습니다. 이 클래스 안에는 3개의 파이썬 클래스와 마찬가지로 `Chaoquopy` 가 정상 작동하지 않을 때 대처한 `OkHttpClient`를 이용한 생성 스크립트 내용이 포함되어 있습니다. 가장 최적의 스토리를 뽑아내는 프롬프트를 여러 번 반복 호출하여 생성한 뒤, api 오류 처리를 전부 해주고, 딕셔너리에 내용을 알맞게 정리하여 반환합니다. 이를 화면의 크기에 맞게 조절하여 필요한 부분만 나오도록 하였고, 이를 다시 한국어로 번역하여 나오고, 영어 부분은 이미지 생성에 들어가게 됩니다. 이미지 생성은 보다 현명한 이미지 생성을 위하여 4단계를 거쳐 생성하고, `Karlo`를 이용하여 깨끗한 이미지 생성을 노렸습니다.

다음으로는 StoryActivity.kt 입니다. 스토리 액티비티 클래스는 Story Adapter 클래스로 1 연결하여 뷰페이지2를 사용하는 역할을 합니다. cicleIndicator를 추가하여 화면을 넘길 때마다 현재 페이지가 어느 지점에 있는지 체크할 수 있도록 코드를 추가하였고, 기본 코드가 메인 스토리 화면이 될 수 있도록, 그리고 데이터를 주고 받을 수 있도록 하는 함수를 넣었습니다. 스토리 선택을 했을 때의 각 fragment 클래스 초기화 함수를 진행하는 함수도 포함되어 있습니다. 이 외에도 현재 페이지를 출력하는 등 추가로 넣을 때 필요한 함수들을 작성하였습니다.



다음은 StoryFragment 클래스입니다. 이 클래스에는 일단 전체적으로 다음과 같은 프롬프트를 이용하였습니다.

```
Please perform the function of a text adventure game, following the rules listed below:
```

```
Presentation Rules:
```

```
1. Play the game in turns, starting with you.
```

2. The game output will always show 'Turn number', 'Difficulty', 'Time period of the day', 'Current day numbeqr', 'Weather', 'Health', 'XP', 'AC', 'Level', 'Location', 'Description', 'Gold', 'Inventory', 'Quest', 'Abilities', and 'Possible Commands'.
3. Always wait for the player's next command.
4. Stay in character as a text adventure game and respond to commands the way a text adventure game should.
5. Wrap all game output in code blocks.
6. The 'Description' must stay between 10 to 20 sentences.
7. For each 'Possible Commands', please write down more than 2 sentences in more detail.
8. Increase the value for 'Turn number' by +1 every time it's your turn.
9. 'Time period of day' must progress naturally after a few turns.
10. Once 'Time period of day' reaches or passes midnight, then add 1 to 'Current day number'.
11. Change the 'Weather' to reflect 'Description' and whatever environment the player is in the game.

Fundamental Game Mechanics:

1. Determine 'AC' using Dungeons and Dragons 5e rules.
2. Generate 'Abilities' before the game starts. 'Abilities' include: 'Persuasion', 'Strength', 'Intelligence', 'Dexterity', and 'Luck', all determined by d20 rolls when the game starts for the first time.
3. Start the game with 20/20 for 'Health', with 20 being the maximum health. Eating food, drinking water, or sleeping will restore health.
4. Always show what the player is wearing and wielding (as 'Wearing' and 'Wielding').
5. Display 'Game Over' if 'Health' falls to 0 or lower.
6. The player must choose all commands, and the game will list 3 of them at all times under 'Commands', and assign them a number 1-3 that I can type to choose that option, and vary the possible selection depending on the actual scene and characters being interacted with.
7. The 3th command should be 'Other', which allows me to type in a custom command.
8. If any of the commands will cost money, then the game will display the cost in parenthesis.
9. Before a command is successful, the game must roll a d20 with a bonus from a relevant 'Trait' to see how successful it is. Determine the bonus by dividing the trait by 3.
10. If an action is unsuccessful, respond with a relevant consequence.
11. Always display the result of a d20 roll before the rest of the output.

12. The player can obtain a 'Quest' by interacting with the world and other people. The 'Quest' will also show what needs to be done to complete it.
13. The only currency in this game is Gold.
14. The value of 'Gold' must never be a negative integer.
15. The player can not spend more than the total value of 'Gold'.
16. The difficulty level ranges from 1 to 10, and the lower the number, the easier the difficulty. Contents change according to difficulty. When HP is continuously maintained at 20 or less attacks are received, the difficulty level is changed to a higher level than the previous level. When HP is less than 5 or close to GAME OVER, the difficulty level is changed to a lower level than the previous level.

Rules for Setting:

1. Use the world of {change_topic}.
2. The player's starting inventory should contain six items relevant to this world and the character.
3. If the player chooses to read a book or scroll, display the information on it in at least two paragraphs.
4. The game world will be populated by interactive NPCs. Whenever these NPCs speak, put the dialogue in quotation marks.
5. Completing a quest adds to my XP.

Combat Rules:

1. Battles can only begin when players have combat items in their inventory.
2. If 'dexterity' is low or injured, there is a less than 40% chance of losing HP while using combat items.
4. Combat should be handled in rounds, roll attacks for the NPCs each round.
5. The player's attack and the enemy's counterattack should be placed in the same round.
6. Always show how much damage is dealt when the player receives damage.
7. Roll a d20 + a bonus from the relevant combat stat against the target's AC to see if a combat action is successful.
8. Who goes first in combat is determined by initiative. Use D&D 5e initiative rules.
9. Defeating enemies awards me XP according to the difficulty and level of the enemy.

Refer back to these rules after every prompt.

Start Game.

꽤 긴 내용의 프롬프트 생성 요구입니다. 하지만 이런 프롬프트를 적용해야만 주어진 요건을 만족할 수 있기 때문에 해당 프롬프트의 내용을 지속적으로 messages에 전달하는 함수가 있습니다. 이렇게 생성한 질문 프롬프트도 약 각 속성의 10가지 이상의 경우의 수로 출

력하기 때문에, 해당 부분을 효과적으로 저장하기 위해서 분류 함수를 추가적으로 작성하였습니다. Chaquopy 로 왔다 갔다 하는 데이터는 반드시 형변환을 한 번 거쳐줘야 하므로 나중에 선택지를 입력한 함수는 변환을 하도록 하는 부분을 추가적으로 구현하였습니다.

이렇게 생성된 답변은 'Turn number', 'Difficulty', 'Time period of the day', 'Current day number', 'Weather', 'Health', 'XP', 'AC', 'Level', 'Location', 'Description', 'Gold', 'Inventory', 'Abilities', 'Quest', 'Possible Commands', 'Story' 이렇게 영어로 번역되어 각각의 디크너리 키와 밸류값에 맞춰 저장됩니다. Possible Commands나 어빌리티의 경우 게임 진행에 중요한 점이므로 이를 다시 키와 밸류값으로 나눠 저장하여 오류 검사를 확실하게 처리하였습니다. 이를 각각 handlerresult 에서 Map형식으로 바꾼 후 각각 변수에 값을 전달하며 어떤 값은 번역되어 나타나고, 변화되었다면 그 부분이 시스템에 반영되도록 작성하였습니다.

이후 이미지 첨부은 현재 보고서 길이가 너무 길어지게 되므로 ppt에 간략한 설명과 함께 어떤 부분인지 인식하실 수 있도록 따로 첨부하도록 하겠습니다.

스토리 화면에는 기본적으로 플레이어가 선택했을 때, 그 결과로 인해 발생한 그 즉시의 상황을 이야기로 전달합니다. 그리고 그 아래에는 그 이후의 이야기를 전달하고 이어서 장비 착용과 장신구 착용 문구가 나타나게 됩니다. 딜레이와 스페너블빌더를 이용하여 타자기 효과가 나타나도록 진행하였습니다. 더불어 MediaPlayer를 이용하여 타자기 효과음이 들리도록 하였습니다. 그 다음으로 있는 inventory 함수입니다. 인벤토리 버튼을 누르게 되면 나타나는 화면으로 리스트에 있는 값을 가져와 LeftList와 RightList에 나누어 업로드합니다. 각각에서 필요 없는 접두사 문구들은 전부 제거를 한 깔끔한 형식의 아이템들이 나오며, 그 아이템 들 중 one, two, three 이런 식으로 수를 표현하거나 다른 방식으로 수를 표현하는 방식을 통일하여 개수값을 저장해놓는 리스트를 하나 생성해놓았습니다. 그를 통해 최대 개수를 정했으며, 여러 개를 선택할 수 있도록 클릭리스너를 설정하여 그 변화를 감지할 수 있도록 추가 구현하였습니다. 이 외에도 골드 시스템 같은 경우도 유사하게 작동합니다.

다음은 conditon함수입니다. 상태창으로 불리며, 앞서 말한 답변 속성들 중 가능한 명령어, 설명, 스토리를 제외한 전부가 해당 화면에 표시되게 됩니다. 어빌리티의 경우 인 게임 진행에서의 영향을 미치도록 하였고, 설득력 같은 경우 누군가와 대화 진행 정도를 조절하도록 하였습니다. 힘의 경우 잠겨 있는 문을 열 수 있거나 그런 확률을 프롬프트로 지정해서 넣어주었습니다. 손재주의 경우 제작 성공 확률 실패 확률이 정해져있습니다. 이 외에도 행운, 지식 등이 게임 진행의 도움을 주도록 설계했으며, 각각의 정보 또한 한글로 번역 전과 후에 나타날 수 있도록 하였고, 창을 잡고 이동할 수 있도록 클릭리스너 부분을 조금 수정하여 창을 잡고 움직일 수 있도록 하였습니다.

스토리 화면의 마지막으로 chat함수입니다. 기본적으로 간단한 챗봇 함수와 dialogflow2를 이용해 학습한 com.example.chatbot2 패키지가 있습니다. 패키지 안에는 필요하다고 생각한 클래스들을 생성하였으며, 대표적으로 메시지의 변화를 감지하고 레이아웃에 표현하는 등의 설정을 가진 메시지 어댑터 클래스와 세션 아이디를 구분하는 세션 아이디 클래스, 챗 액티비티 클래스로 전체적인 dialogflow2 학습된 데이터의 아이디를 입력받아 정보를 반환하고 오류가 생길 경우 오류 메시지를 반환하는 클래스 등이 있습니다. 이 부분이 무거워 앱 진

행에 지장을 주기 때문에, 간단한 챗봇을 따로 구현을 하였는데 이는 chat gpt를 이용하여 간단하게 배경 프롬프트를 따로 생성한 것을 가지고 질문을 하여, 그에 적절한 인물을 생성하고 그와 대화를 하는 방식입니다. 그 인물에게는 ['Name', 'Residence', 'Hobbies', 'Personality', 'Role'] 같은 속성들을 가지고 있습니다.

다음으로 선택 화면입니다. 선택 화면은 기본적으로 스토리 화면의 가능 명령어들을 가져와서 진행을 하고, 추가적으로 3번째 명령어를 받습니다. 3번째 명령어는 CHAT GPT가 추천하는 명령어로 이는 우측 상단에 숨겨진 아이콘으로 나타나있으며, 스토리 화면의 가능 명령어가 변경이 되면 이를 인지하도록 작성하여, 이를 인지하지 못했을 경우가 생기는데 이를 새로고침을 통해 방지하도록 하였습니다. 3번째 선택지는 사용자가 직접 입력을 할 수 있는 버튼으로 사용자가 영어든 한국어든 입력을 하게 되면 영어로 번역되어 들어가게 됩니다. 물론 화면에 보이는 선택지는 직접 입력한 언어가 생성됩니다. 그리고 다시 한 번 버튼을 누르게 되면 해당 선택지로 이야기를 진행할 수 있게 되는데 다시 스토리 화면으로 넘어가 이미지 생성 화면과 초이스 화면 클래스들이 용량 차지 부분을 없애도록 초기화하는 부분을 진행하고 다음 선택지로 이동할 수 있게 됩니다.

이미지 생성 화면입니다. Description 상황 묘사 이 부분을 가져와서 이미지를 생성합니다. 이미지 생성 단계는 앞에서 얘기했듯이 4단계로 나누어서 진행됩니다. 그 이야기의 상황을 좀 더 구체화하는 질문으로 배경 이미지를 좀 더 구체적으로 그릴 수 있도록 하였고, 다시 한 번 배경으로 나올 수 있는 적절한 물건 등의 요소를 배출하라고 요구하고 있습니다. 그 다음으로 이를 맨 처음 주제를 강조하기 위해 합친 후 명사와 형용사로 나눈 뒤, Karlo의 프롬프트 bytes 제한 수가 250 바이트인 것으로 알고 있기 때문에, 이를 최대한 이용할 수 있는 방식으로 프롬프트를 진행하였습니다.

그렇게 하여 계속 이야기는 진행하고, 만약 데이터를 저장하고 싶다면 뒤로가기버튼을 한 번 누르게 되면 다시 한 번 누를 수 있도록 메시지를 보내도록 하였고, 2초 안에 2번 연속으로 누르게 된다면 저장 함수로 이동하게 됩니다. 데이터가 온전한지 확인한 후 data.txt라는 파일에 자체 저장이 되고 이는 다시 게임을 실행할 때 이어서 실행할 수 있도록 해줍니다. 데이터가 변경이 되었는지의 유무도 판별하여 플레이어가 데이터를 저장하지 않고 나가지 않도록 진행하였습니다.

Navigation.xml은 어느 화면으로 이동할 지 사용할 때 쓰는 xml로 이도 깃허브에 작성되어 있습니다.

이외의 사항

Chaquopy를 이용하여 파이썬을 호출하고 파이썬을 종료하고 그런 시간을 다 더한 것보다 OkHttpClient를 이용하여 데이터를 주고받고 한 것보다 테스트 해 본 결과 좀 더 빠르다고 생각하였고, 코드를 수정하려면 chaquopy는 반드시 install 해놓는 것이 좋다고 생각합니다. 이 외의 추가적인 코드와 xml 파일들은 조금 더 간편하고 구체적인 디테일을 살리고자 진행한 부분으로 예를 들어 animation효과를 작성한 xml이나 따로 drawable을 제작한 xml 등 해당 코드를 수정하면 좀 더 나은 게임을 만들 수 있다고 판단하여 올려놓지만, 따로 코드

설명은 적지 않도록 하겠습니다.

3. 전체 시스템 UI

사용된 xml 코드는 다음과 같습니다. 디자인된 채팅 화면을 리사이클 뷰를 이용해 제작한 activity_chat.xml. 채팅방을 추가하고 인물 프로필 정보를 볼 수 있게 기본적인 UI인 chat_room_dialog.xml. chat gpt가 가장 최적의 선택지를 배포하는데 그를 표현하는 choice_dialog.xml. 선택지 화면을 그라데이션 느낌으로 표현하는 choice_fragment.xml. 상태를 각종 라이너레이아웃으로 묶어 처리하는 condition.xml. 골드 사용관련하여 나타나는 이미지인 dialog_gold.xml. 아이템을 어느 정도의 양만큼 사용할 지 나타내는 dialog_item_quantity.xml. api키를 입력하고 확인할 수 있고 붙여넣기가 가능한 dialog_settings.xml. 사용자가 직접 입력할 수 있는 선택지 화면인 edit_commands_dialog.xml 메인 화면의 스토리 생성 여부와 함께 first_lobby.xml 과 main_activity.xml. 전체적인 inventory를 표현하는 fragment_inventory.xml. 이미지를 생성하고 새로그침을 할 수 있는 부분인 img_gen_fragment.xml. 이미지 생성 화면인 make_story_fragment.xml. 등장인물과 대화하기에 앞서 등장인물의 각종 정보를 표시하는 profile_info_dialog.xml 이 외의 xml은 추가적으로 다르게 꾸밀 수 있도록 xml화면을 구성해 놓았습니다. 이미지는 너무 많기에 아까 얘기드렸듯이 ppt 화면에 이미지와 함께 설명을 참고하여 넣어놓았습니다.

4. 개발과정

자바를 배우고 익히고 있다 하더라도, 자바를 자동으로 코틀린 코드로 변경하는 것이 아닌 직접 코틀린 코드로 개발을 해보고 싶다고 생각하여, 코틀린을 찾아보며 개발을 진행하였습니다. Xml 디자인 같은 경우에도 어떤 식으로 연계해야 정확한 위치에 표현되는지 아무것도 몰랐기에 해당 xml 디자인을 습득하는데에도 많은 시간이 소요되었고, chaquopy 등 다양한 것들을 implementation 해서 사용 법을 알아내는 데에 많은 시간이 걸렸습니다.

정해진 기간내에 최대한의 목적을 달성하기 위해 하루 많아도 1~3시간을 자며 개발을 진행하였고, 투자한 시간대비 좋은 결과를 얻어내진 못한 것 같지만 여러 가지 방안을 시도했던 것 같습니다. 더 보완하거나 새롭게 추가하고 싶어 만들었다 지운 클래스만 해도 이어하기 여러 개 시스템, 실물 사진 인식 시스템, 카메라 연동 시스템 등 첫 달은 주로 기존의 아이디어에서 추가적으로 원했던 것들을 시도해보았고, 최종적으로 구현 가능한 클래스들이 남아있습니다.

Android Studio에서 잘못 클릭하여 발생한 버그 같은 것이 계속해서 발생했는데, 안드로이드 스튜디오 내에 있는 Git의 commit 과 push를 눌러 데이터를 전송하게 되면 2번 중 한 번 꼴로 데이터 기록이 전부 사라지고 File History 부분도 사라져 데이터 복구를 손수 해야 했다는 번거로움이 있어 깃허브에 업로드를 계속해서 하지 못하였습니다. 코드의 a함수 b함수가 있다고 하면 a함수는 정상 작성되었으나, b 함수 중간중간에 2주 전, 즉 수정하기 전의 코드로 변경되어 남아있는 등 여러 난항을 겪었으나 결과적으로 시도했던 것들의 천운이 따라줘 잘 마무리되었습니다.

5. 주요 실행 화면

<https://drive.google.com/file/d/1ZGbMxVYIZ27Q3ITzcQ8H-M5Mc3AHMrKI/view?usp=sharing>

해당 링크로 들어가게 되시면 빠르게 배속하여 실행한 영상이 있습니다.

6. 목표 달성률

기본적인 목표했던 계획에서 아주 그냥 기본적으로만 구현된 부분의 목표 달성률만을 살펴 보면 등장인물과의 대화 시스템에서 chat gpt 3.5 turbo 무료 버전의 한계상 1분당 3번밖에 질문할 수가 없는데 게임속에서 만난 등장인물과 직접 대화를 하기 위해 체크를 하려면 10번 이상 즉 3~4분의 시간동안 계속해서 기다리고 대화를 해야 하는데 그러기에는 게임이 너무 루즈해진다고 판단하여 이를 수정하고, 대신 다른 변수 시스템 속성, 날씨 등을 추가하였기 때문에 100-5+10 정도 수행했다고 보나, 개인적인 만족도로는 지웠던 클래스들이 정상 작동되었다면 어땠을까 하는 마음에 60프로 정도 생각하고 있습니다.

7. 개선점 및 수정사항

앞서 얘기드렸듯이, 등장인물과의 대화 시스템에서 chat gpt 3.5 turbo 무료 버전의 한계상 1분당 3번밖에 질문할 수가 없는데 게임속에서 만난 등장인물과 직접 대화를 하기 위해 체크를 하려면 10번 이상 즉 3~4분의 시간동안 계속해서 기다리고 대화를 해야 하는데 그러기에는 게임이 너무 루즈해진다고 판단하여 이를 수정하고 속성, 날씨 등을 추가하여 보다 게임이 다채로워질 수 있도록 변수를 추가하였습니다. api키를 직접 발급받아서 할 수 있도록 수정하였고, 장비 착용 장신구 착용 등의 진짜 현실에서 살아간다는 느낌을 주게끔 작성하였습니다.

8. 감상

다른 이들에 비해 초라하고 간단한 게임처럼 보일 수도 있겠지만, 제 인생에서 한 프로젝트를 위해 잠까지 줄여가며 했던 경험은 몇 번 없었기에 매우 소중한 경험이었습니다. 덕분에 다양한 라이브러리의 활용 방법과 안드로이드 스튜디오의 기본 작성 방법 등 다음번에 개발한다면 이보다 더 효과적으로 개발할 수 있다는 생각을 가지게 되었습니다. 매일 새벽 4시에 자고 아침 5~7시에 일어나기까지 계속해서 어떻게 하면 이 부분을 수정할 수 있을까, 개선할 수 있을까 그리고 무엇을 추가해볼 수 있을까 하는 생각으로 가득찼던 것 같습니다. 거창한 것을 잡고 하기에는 아직 기본 지식이 부족한 것 같습니다. 한 학기 동안 좋은 피드백 감사드립니다.