

Chat gpt api를 이용한 파일 점령 게임

목차

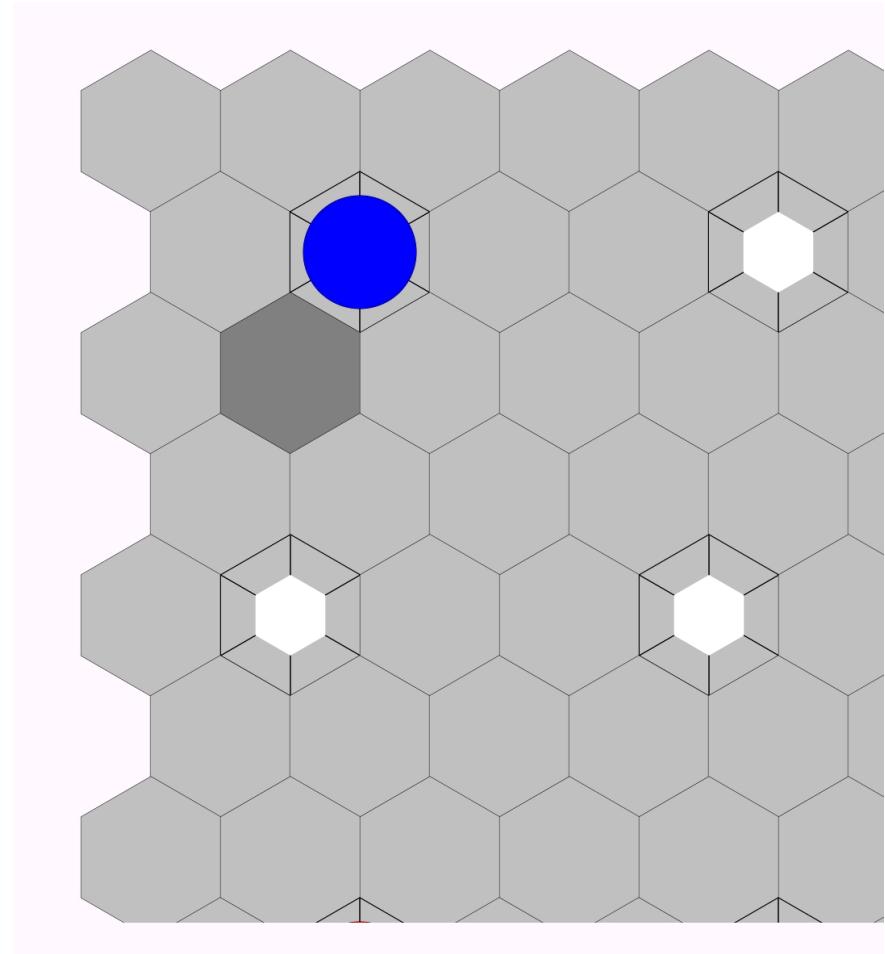
- 게임 개요
- 맵
- 유닛
- 생산
- 상대 ai와 Chat gpt로 상대 ai의 대사 생성
- 시연

게임 개요

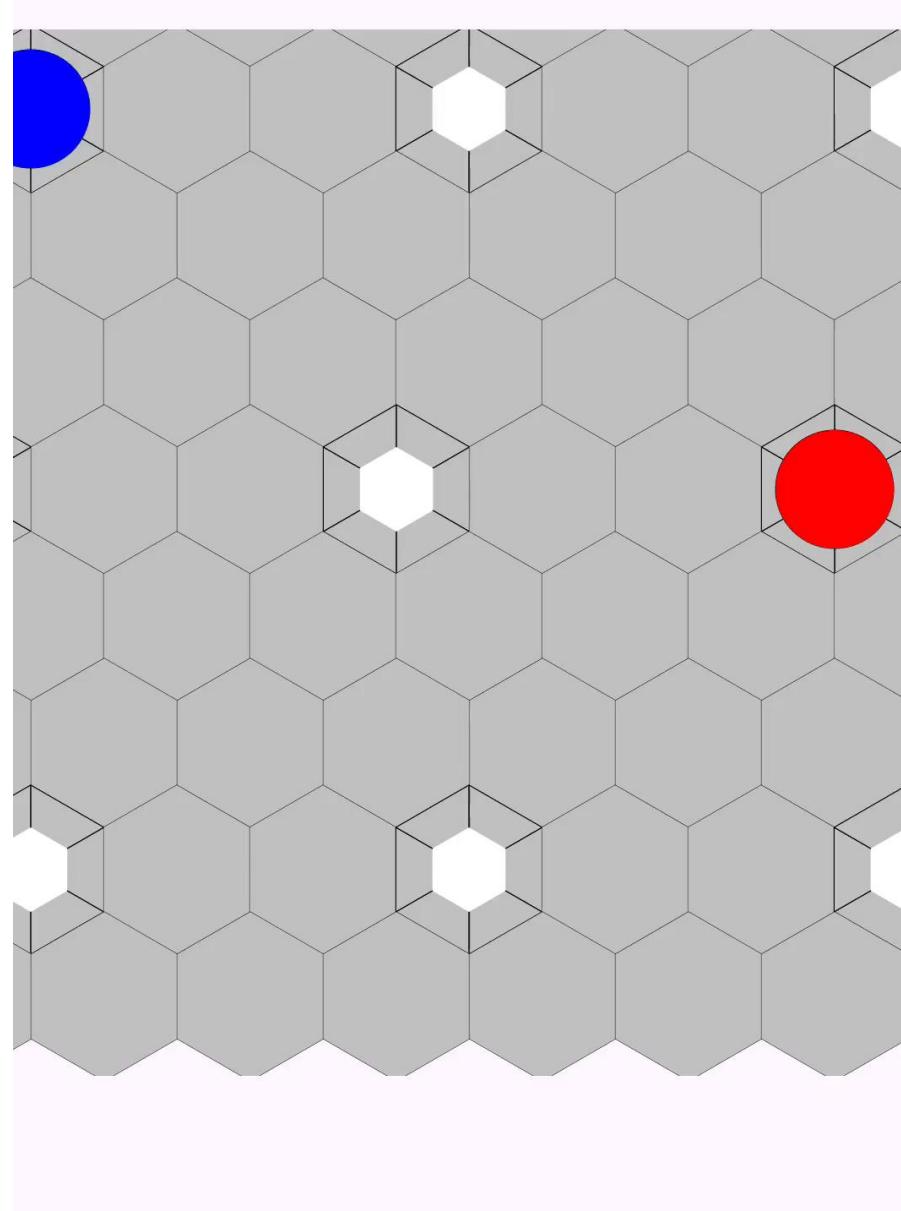
- 9x9 육각형 타일 맵에서 타일을 점령하거나
- 상대 유닛을 공격하여
- 맵의 절반 이상의 타일을 점령하거나
- 상대 유닛을 모두 없애면 승리하는 게임

맵

- 점령 되지 않은 타일은 회색
- 점령된 타일은 해당 유닛의 색
- 중앙에 하얀 육각형이 있는 타일은 유닛을 생산할 수 있는 타일
- canvas로 구현

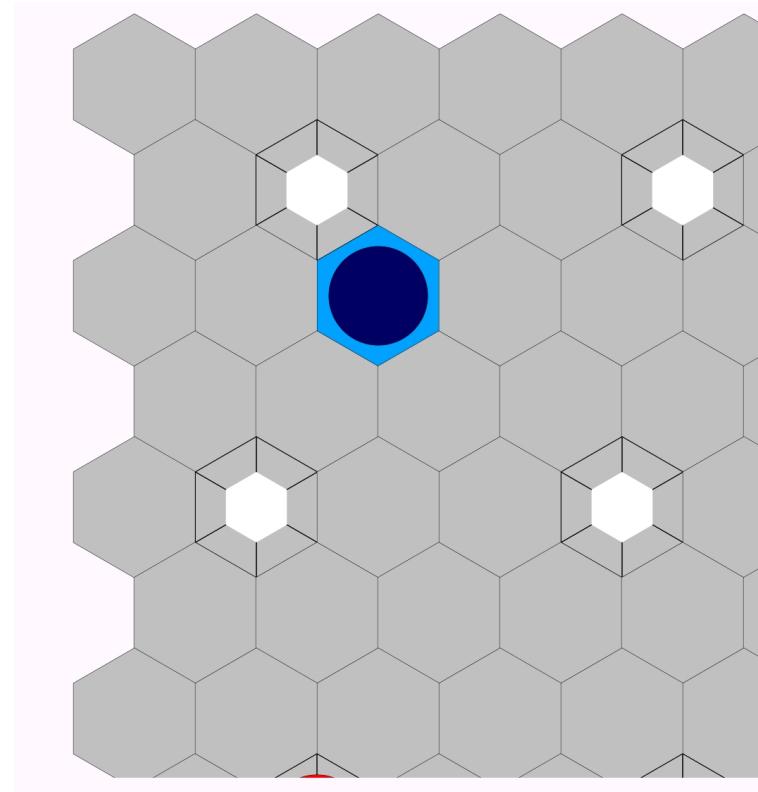
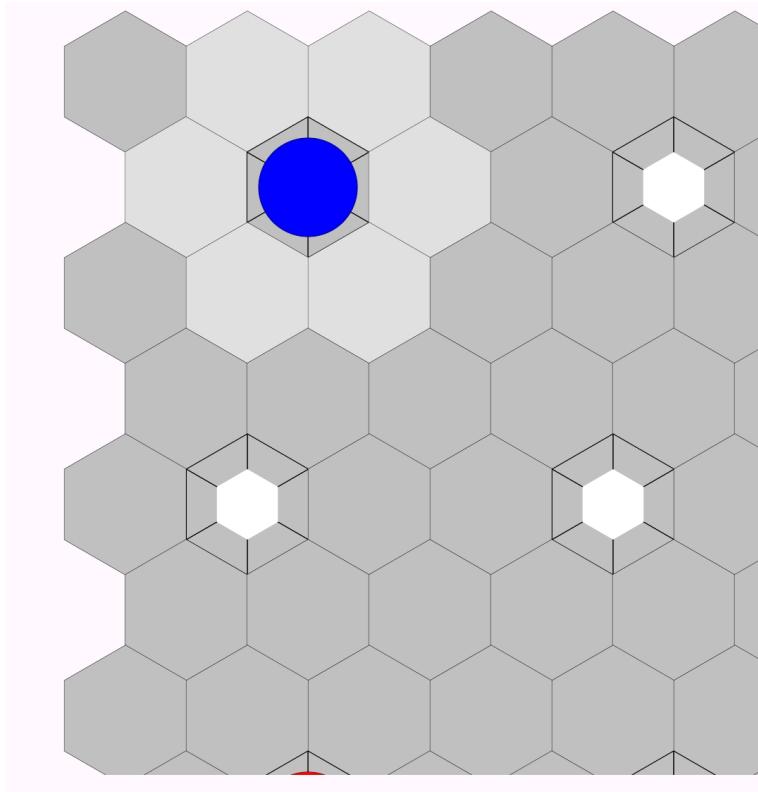


□ H
E



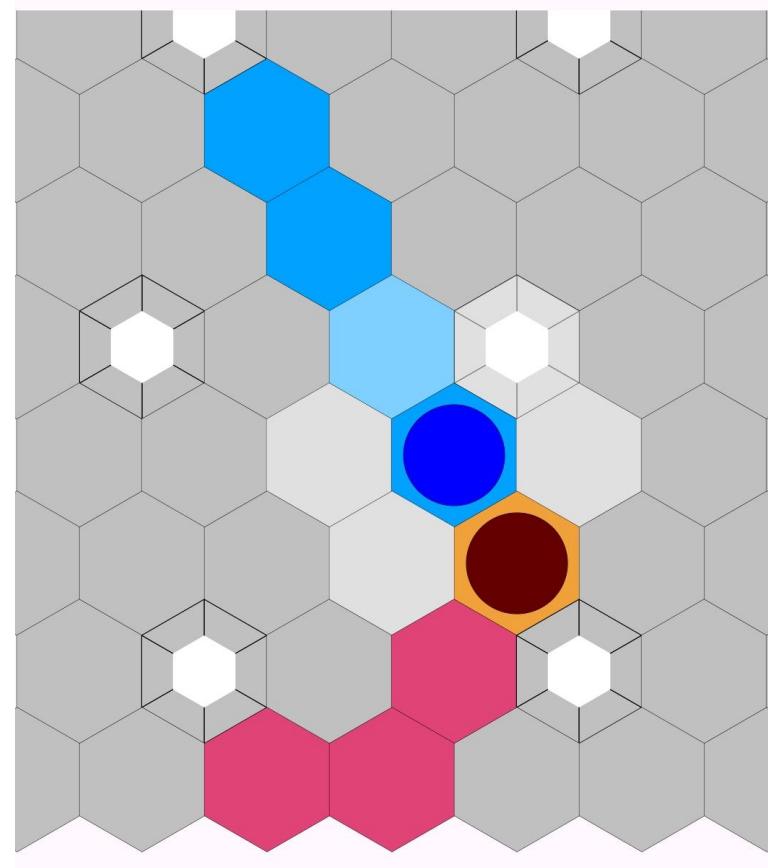
유닛

- 한턴에 한번, 주위의 1타일 안에서 이동하거나 공격 가능



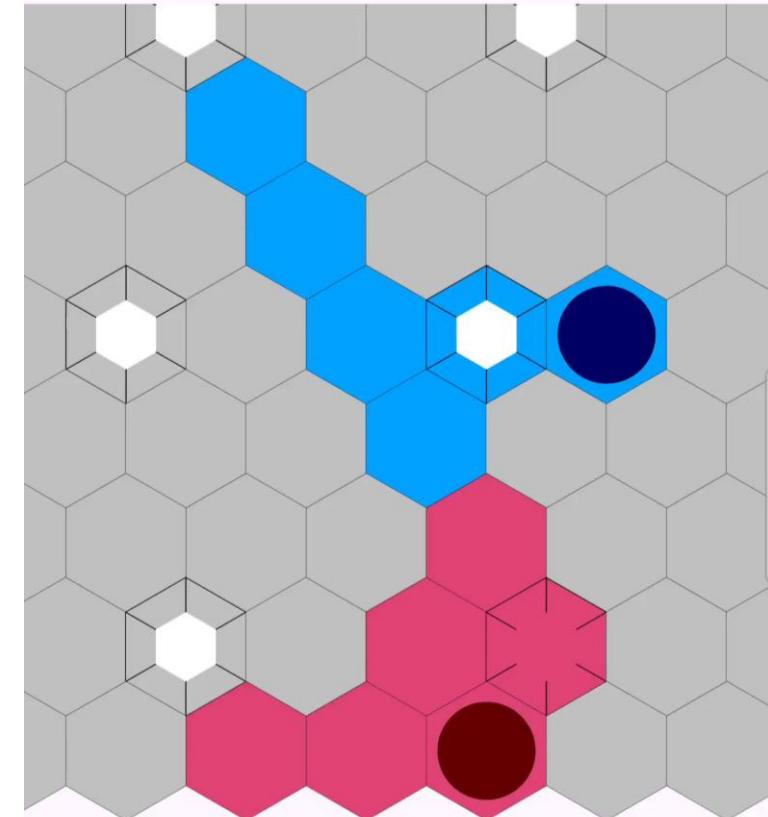
유닛

- 한턴에 한번, 주위의 1타일 안에서 이동하거나 공격 가능



생산

- 생산 타일을 점령하고,
요구하는 타일 개수 이상의
타일을 점령 했을 시 유닛 생산.
- 생산 후 비용은 2배씩 증가



남은 턴

현재 턴 : 7

턴 넘기기

점령한 타일 수 : 6

상대가 점령한 타일 수 : 6

이름 : 생산

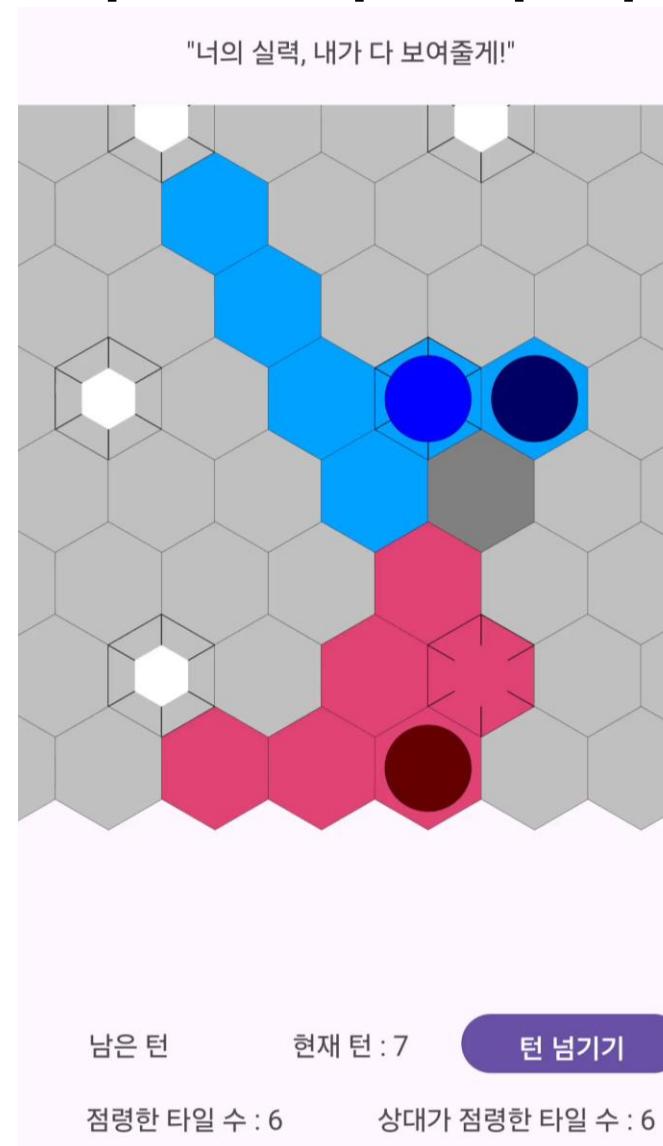
소유자 : user1

비용 : 2

생산하기

상대 ai와 Chat gpt로 상대 ai의 대사 생성

- 턴 넘기기 버튼을 누르면 상대 ai로 턴이 넘어가고 상대 ai는 행동을 한다
- chat gpt 로 상대 ai는 유저를 도발하는 대사를 한다



상대 ai와 Chat gpt로 상대 ai의 대사 생성

```
// GPT API 호출
callGPTAPI( prompt: "턴제 게임에서 상대방을 도발하는 대사를 20글자 이내로 창의적으로 작성해줘.", () -> {
    isAPICalled.set(true);
    checkAndShowTurnButton(isAICompleted, isAPICalled);
});
```

```
// GPT API 호출 메서드
private void callGPTAPI(String prompt, Runnable onComplete) { 1 usage
    String apiKey =
        ;// "asdf"; // 환경 변수나 안전한 저장소에서 가져오는 것을 권장
    Retrofit retrofit = RetrofitClient.getClient(apiKey);
    GPTService gptService = retrofit.create(GPTService.class);

    GPTRequest request = new GPTRequest(prompt);

    Log.d( tag: "GPTRequest" , msg: "Request Data: " + new Gson().toJson(request));

    gptService.getResponse(request).enqueue(new retrofit2.Callback<GPTResponse>() {
        @Override
        public void onResponse(Call<GPTResponse> call, retrofit2.Response<GPTResponse>
            System.out.println(response.message());
            if (response.isSuccessful() && response.body() != null) {
                // 응답 본문이 null이 아닌지 확인
            }
        }

        @Override
        public void onFailure(Call<GPTResponse> call, Throwable t) {
            Log.e("GPTRequest", "onFailure: " + t.getMessage());
        }
    });
}
```

시연

게임 메뉴

게임 시작