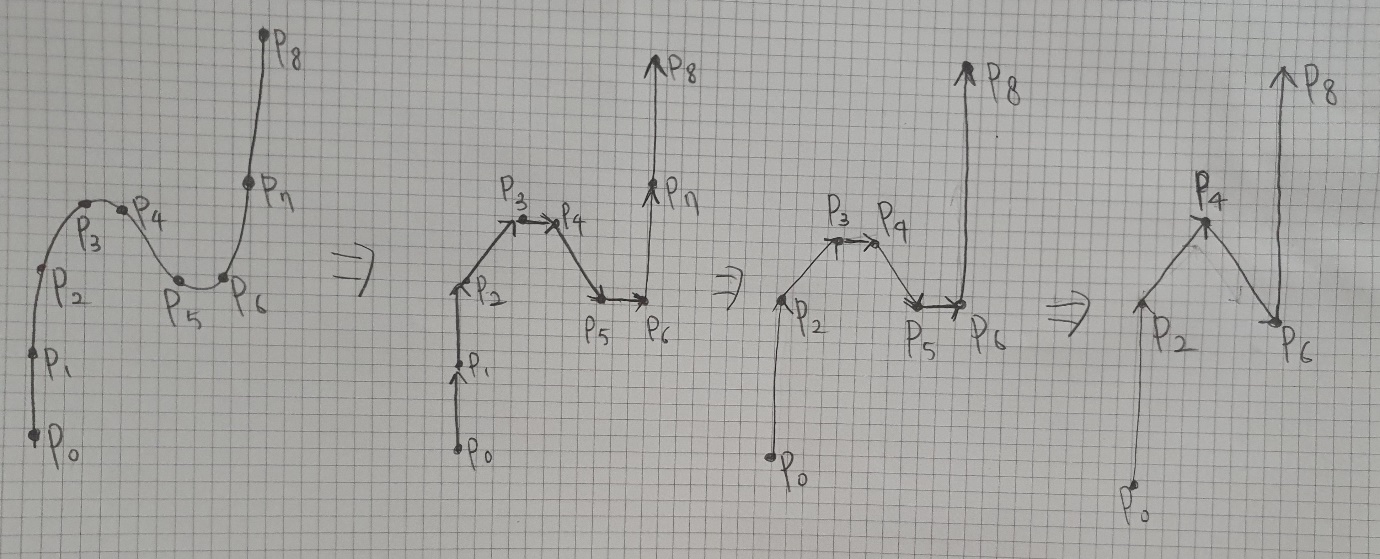
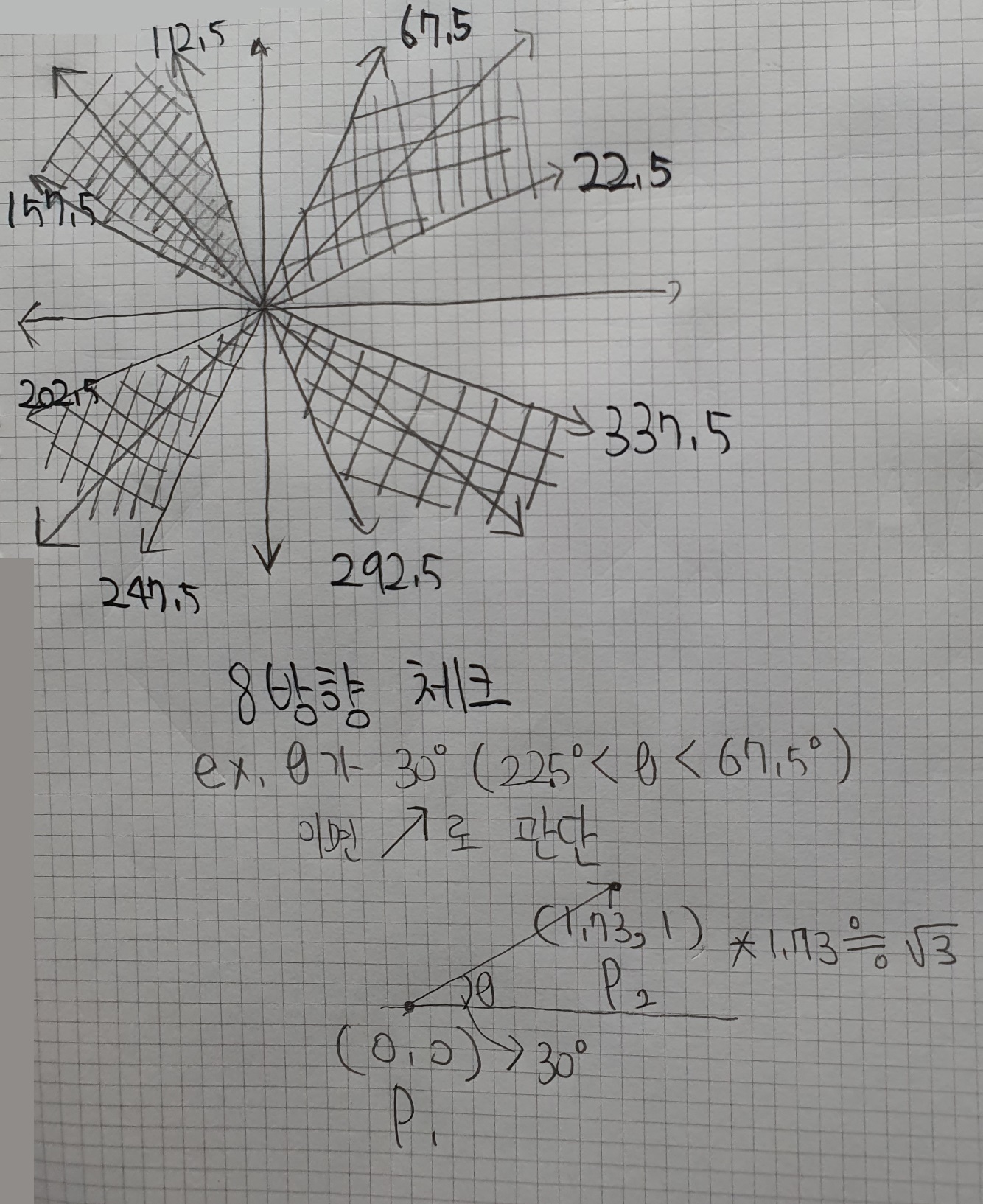
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 5 주차 | **기간** | 2.03 ~ 2.09 | **지도교수** | (서명) |
| 이번주 한일 요약 | 라인 보간 알고리즘 구현 | | | | |

<상세 수행내용>



1. 8방향 체크

* 직선이 어떤 방향을 향하는지 판단함
* Ex. P0 -> p1 의 각이 22.5~67.5 사이라면 ‘우상’의 직선이라 판단함
* Ex. P1 -> p2 의 각이 100도 라면 67.5~112.5 도 사이이므로 ‘상’의 직선으로 판단

1. 보간 알고리즘 3단계

* 1단계 : p0->p1, p1->p2 …. 의 직선들이 어떤 직선인지를 판단함 (상, 좌상, 좌하 등등)
* 2단계 : 직선의 성향이 연속적이라면 하나의 직선으로 합침 (p0->p1, p1->p2의 직선은 ‘상’직선 이므로 p0->p2의 직선이 됨)
* 3단계 : ‘p3->p4’ 직선은 전체적인 길이로 따져봤을 때 짧은 직선이니 p3의 좌표를 탈락시키고 ‘p2->p4’의 직선을 만듬 (
* **3단계 보간의 이유**
* 1. 문자를 그리는건 사람이기 때문에 의도치 않은(손떨림 등으로 인한) 선이 그어질 수 있음 이를 방지하기 위함
* 짧은 직선을 배제시켜 정확도를 올림

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | - | | |
| **해결방안** | - | | |
| **다음주차** | 6 주차 | **다음기간** | 2.03 ~ 2.10 |
| **다음주 할일** | 게임에서 사용할 마법문자들을 입력하여 일치판단을 해봄 | | |
| **지도 교수**  **Comment** |  | | |