

- 일단 모든 modal이 무조건 닫혔다가 열리는 것 같음.
- user의 진도는 모두 MongoDB users collection에 있음
  - \* user.info.progress -> [city][step][unit]
  - \* level의 값
    - 1: 성공한 것들
    - 2: 실패한 것들
    - 3: 현재위치 + success
    - 4: 현재위치 + fail
- start, end는 따로 처리
- start~level~end의 모든 정보는 각 game style 별 service에 있음
- step>unit의 정보들은 MongoDB appPages의 {"name": "pcd"} document에 있음.
  - > game의 형태가 어떤것인지(Art, ...) -> 이것통해서 적당한 화면을 만들어 내는 듯
  - > title정보, mode정보, maxLevel정보
- 1. 문제풀이 화면 slider -> 번호로 페이징
  - > slider가 모든 modal 형태의 template에 추가되어 있는 듯

## \* 수정포인트

art.pug, bee.pug, board.pug, cross.pug,  
dig.pug, lab.pug, maze.pug, multi.pug, puzzle.pug,  
editor.pug, unplugged.pug

추가적인 부분(style 등) 확인요망

-> game.service.js에서 whereStartGame을 통해 처음 열리는 level 판단

-> 각 game type 별 모든 controller에서 listInit를 통해 선택된 level의

list item style을 변경해준다.

\* left navigation으로 바꿀 때 해당 class의 css 수정요망

2. 문제풀이 화면 페이지 추가/삭제

-> art.service.js, bee.service.js, ... 에서 템플릿들 (start~/~x~end) 수정해줘야한다.

-> MongoDB users collection migration 필요한지 고민

\* game.service.js에 unit, level 개수 맞추는 작업이 있음(compareProgress)

이 부분만으로 migration없이 가능할까?에 대한

고민

-> 첫 페이지, 마지막 페이지라면 단순하게 template 수정(game type 별 service에 있음)

3. 힌트보기사 문구 나오는 것 영역 확인

-> art.pug, bee.pug, cross.pug, dig.pug, lab.pug, maze.pug, multi.pug

-> 필요문구 보내줘야함. <요청하세요>

4. 정답 코드 클릭 시 그림 나오는 것 영역확인

-> 3번과 동일 영역에 나오도록

-> 1번 방안 : 정답코드 캡처 후 화면에는 그림파일 띄워주기(추천)

: zoom 되도록

-> 2번 방안 : xml로 block 만들어서 보여주기

5. 나가기 버튼 영역확인

-> art.pug, bee.pug, board.pug, cross.pug, dig.pug, lab.pug, maze.pug, multi.pug, puzzle.pug

-> button 옆에 글자 추가하고 listener 영역 수정

6. 네비게이션 메뉴 버튼 넣을 영역 확인

-> gameHeader 수정된 부분에 추가

7. 이전/다음 영역확인

-> gameHeader 수정된 부분에 추가