MAIN GAME

1. Main game lifecycle
2. Global store cache and management
3. Random map generation
4. Character generator
5. Grid and Tile object, positioning
6. Optimized drawing update
7. Character information drawing
8. Character action value update (turn-based core logic)
9. Character stats create, update and management
10. Action and character action, probability based action
11. Character class, faction and hostile
12. Character level management and class level
13. Event management
14. Combat event initialization and management (life-cycle)
15. Character vision and surround area inspection
16. Character memory, remember entity and management
17. Character power and power examination
18. Character movement based on memory and power estimation
19. Character behavior
20. Character escape from battle at low health and depend on behaviors
21. Character status and status definition, affection on character stats and expired
22. Character status initialization via actions
23. Special tiles affect character action and combat, refactor logic when character move into a new tile => In-progress
24. More tile types, map generation and regions
25. Random generated objects and items/loots
26. Character interaction with items/loots
27. Longtime decision that manage character action (goal system)
28. Complex character leveling (motivate for complex decision instead of repeating basic actions)
29. Complex memory management
30. More character action types
31. Character strategies

OTHER COMPONENTS

1. Nextjs server for character and combat information
2. Kafka message broker for handle update to mongodb
3. Python consumer that update data to mongodb and send notification to subcribed notify
4. Nodejs server for handle notification subcription, communicate with Python through gRPC, send notification to Nextjs through websocket
5. Containerization & processes management

DOCUMENTATIONS

1. Class diagram
2. Interactions & mechanisms