

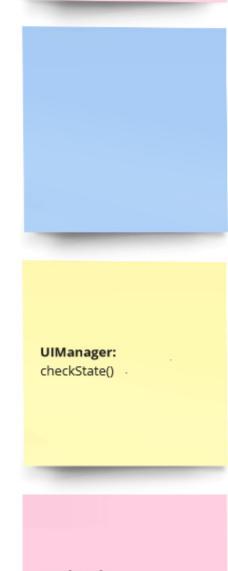
curDialogue)

private:

int curDialogue

vector<string> dialogues

void action(Player* player)



Constants:





Screen:

CharacterClass: Warrior Healer Magician

MiniGame:

GameManager: Team: bool isSafeScreen(Screen currScreen) void addMember() void randomEnemyEncounter() vector<Entity*> bool isTeamAlive()

setFightOrder(Team* good, Team* bad) private: void Fight(vector<Entity*> order) void transition()

Assassin

vector<Entity*> members

vector<Item*> inventory

LevelSystem:

int health int maxMana int mana int accuracy int attack int spAttack int defense int spDefense int speed int critical

Stats: int level

int maxHealth

Entity: bool isAlive() void modifyHealth(int value) void modifyMana(int value) virtual Ability* chooseAbility() virtual Entity* chooseTarget(Team* targets)

void useAbilityOnTarget(Ability* ability, Entity* target) void useAbilityOnTeam(Ability* ability, Team* targets)

void setStats(Stats addStats) private: Stats stat

Ally:

virtual Ability* chooseAbility()

virtual Entity* chooseTarget(Team* targets)

void removeltem()

Enemy: virtual Ability* chooseAbility()

virtual Entity* chooseTarget(Team* targets)

Item:

Ability:

int level

bool isAOE

void effect(Entity* target)

private: int cost

Player:

void move(char input, int curScreen) void interact(int curScreen) void addItem()

private: struct position