High Level Analysis

Fantasy Rpg:
Three classes - Mage, Rogue, Fighter.

GameLoop:
Player faces a Fight, Puzzle, And Riddle.
Randomize order.

Gold:
player accumulates gold via Events.

Stretch goal:
Items

Create a Character.
Assign that Character a class.
first challenge, (random), fight.
Take player & npc hp, attack, defense, use randomness plus these stats to decide winner.
Next challenge, random excluding fight. (example Riddle)
takes player obj, tells riddle (output) if player instances mage, has option to do two riddles. if riddle is answered wrong, lose 25pt health.
final challenge, (Example puzzle)
random math problem. wrong answer results in loss of 25pt of health.

(player dies at any point) goes back to beginning, option to restart or exit.
(Endgame) If Player doesnt have enough treasure, they reHeal and restart.
If Player does have enough treasure, Player wins.

Jake

App:
-final int treasureGoal

- player creation
- player eventLoop
- player

EndGame:
Treasure check for win.

Player must have enough player.

EventType enum

INIT,
CHALLENGE(Challenge),
TAKE_DAMAGE(GameCharacter entity),
HEAL(GameCharacter entity),
NEXT_CHALLENGE,
RESTART,
EXIT,

Osvaldo

abstract GameCharacter Model:
Int currHealth;
int maxHealth;
String name;
int attack;
int Defense;

boolean isAlive();
void takeDamage(int);
Void recoverHP(int);

Atk/def is set in class player gives just name and class type.

Game Character AbilityClass

Interface:
special_ability

Player

Int treasure;

Boolean usedAbility;

void setClass(class)

Enemy
String type;
init randomized atk/def

Mage
More atk
Less def
Norm health

Rogue
Norm health
Norm def

Fighter

more max health.
Better ask/def.

Jacob

<Abstract>
Challenge

Player player;

checkAbility;

Riddle

Multiple choice riddle.

map<String, String[]>

Puzzle

Int[] operands;
Char operator;

createProblem;
int checkAnswer(int);

Fight

Enemy enemy;

method initFight;
Method fightLoop;
Method fight;

Enum Treasure

Coin
Satchel
Chest