The Programming Isekai Adventure

Kin NG

PROJECT SUMMARY

The project is based on the popular Japanese mobile games called “gacha” which simulates a toy-vending machine system. The core of gacha games mainly consist of a summoning system in which players use in-game currency to obtain their desired characters. For this project there will be a feature known as banner which consist of a collection of unique characters who have a respective rate of appearance when trying to get them (1% rate for 3 stars, 4% rate for 2 stars, 5% rate for 1 star) thus simulating the gacha mechanic. Since you might not get the strongest characters at the beginning, the game will introduce a second feature which consists of an arena mode in which players can take their acquired characters into battle and earn more in-game currency from which they can summon more characters. Moreover, the arena mode will have 2 sub modes. The first will be single battle mode into which a player will go into a 1v1 battle against a randomized enemy from the available pool of enemies and earn currency to summon in the gacha while being able to experiment with each character. The second mode will be wave mode into which players select a roster of 3 characters to battle 3 waves of enemies, this mode is more challenging but it will also increase the currency earned by the user. Normally, every battle mode will terminate in a win or a loss, depending on whose health, enemy or player character, was reduced first. The battle system will be created by using a randomized turn based system from which the player can select a course of action (attack, heal, talent) for their character. In addition, each battle will be defined under the player vs. computer system.

OVERVIEW OF CLASSES

Several classes were created for this project:

1. **Character**: This class contains 6 data members and 16 member functions. Each Character object represents a character to be summoned in gacha and to be used in arena.
2. **Enemy**: This class contains 2 data members and 6 member functions. The class inherits from the Character class and each Enemy object represents an enemy that is encountered in arena mode.
3. **User**: This class contains 4 data members and 14 member functions. The User object created stores all of the aspects related to the new/current user logged in the session.
4. **Game**: This class contains 6 data members and 18 member functions. The game object stores all of the data related to the current state of the game, and provides the functionality for the two main game modes. Lastly, it loads the data previously saved into the user object in session.

CLASS DESCRIPTIONS

CHARACTER CLASS
Abstract class: No
Subclass of: N/A
Composed of: N/A
Alterations post proposal:

- Another constructor was added to be inherited by the enemy class, one that does not include rarity.

DATA MEMBERS - CHARACTER			
Variable Name	**Data Type**	**Static**	**Description**
name	string	no	Holds the name of the gacha character.
hp	int	no	Holds the base health stat of a gacha character.
atk	int	no	Holds the base attack stat of a gacha character.
talent	string	no	Holds the talent name that allows a boost to stats unique to certain characters.
rarity	int	no	The rarity of a character (3 star, 2 star, 1 star).
type	string	no	Unit type (mage, demon lord, human, angel, elf, sniper, god).

MEMBER FUNCTIONS - CHARACTER					
Signature	**Static**	**Virtual**	**Operator**	**Friend**	**Description**
Player(string ,int,int,string, int,string);	no	no	no	no	Constructor for a player object.
Player(string ,int,int,string,string);	no	no	no	no	Constructor to be inherited by enemy object.
void set_name (string);	no	no	no	no	Set the name of a gacha character.
string get_name();	no	no	no	no	Get the name of a gacha character.

void set_hp(int);	no	no	no	no	Set the health stat of a gacha character.
int get_hp() ;	no	no	no	no	Get the health stat of a gacha character.
void set_atk(int);	no	no	no	no	Set the attack stat of a gacha character.
int get_atk();	no	no	no	no	Get the attack stat of a gacha character.
void set_talent (string);	no	no	no	no	Set the talent name of a gacha character.
string get_talent();	no	no	no	no	Get the talent name of a gacha character.
void set_rarity(int);	no	no	no	no	Set the star rarity of a gacha character.
int get_rarity();	no	no	no	no	Get the star rarity of a gacha character.
void set_type (string);	no	no	no	no	Set the unit type of a gacha character.
string get_type();	no	no	no	no	Get the unit type of a gacha character.
virtual void talent_descri ption();	no	yes	no	no	Get the complete description of a particular talent unique to that character, but subclasses can format accordingly.
bool operator> (const Character& right)	no	no	yes	no	Compare if the character to the right is of least rarity than the character to the left, if rarity is the same use power formula. If it is, return true

ENEMY CLASS
Abstract class: No
Subclass of: Character
Composed of: N/A
Alterations post proposal: N/A

DATA MEMBERS - ENEMY			
Variable Name	**Data Type**	**Static**	**Description**
territory	string	no	Territory that a particular enemy governs (dark world, pixie land, the wired, flamagurr, iron empire).
weakness	string	no	Type disadvantage of an enemy.

MEMBER FUNCTIONS - Enemy					
Signature	**Static**	**Virtual**	**Operator**	**Friend**	**Description**
Enemy	no	no	no	no	Constructor for an enemy object.
void set_territor y(string);	no	no	no	no	Set the territory of an enemy.
string get_territor y();	no	no	no	no	Get the territory of an enemy.
void set_weakn ess(string) ;	no	no	no	no	Set the type disadvantage of an enemy.
string get_weakn ess();	no	no	no	no	Get the type disadvantage of an enemy
void talent_des cription() const;	no	no	no	no	Get description of the talent based on an enemy.

USER CLASS
Abstract class: No
Subclass of: N/A
Composed of: Character
Alterations post proposal:

- A Character parameter was added to the add_character function in order to facilitate the functionality of adding a character from gacha to a user's player box.
- An int parameter was added to delete_character to pinpoint the specific index that in which the Character object within the user's player box exists.
- The function retrieve_character was added because it helped with the recurrent activity of retrieving a specific character from the user's player box which was fundamental to the both game modes within arena mode.
- The inclusion of the write_to_file function was necessary to save the progress of each user within the text file every time a modification happened in a game session.

DATA MEMBERS - USER			
Variable Name	**Data Type**	**Static**	**Description**
vector<Character> player_box	Character	no	Container of the acquired gacha characters of a user.
currency	int	no	The in-game currency of the user in session.
arena_wins	int	no	The win record of a user in the arena game mode.
arena_losses	int	no	The losses record of a user in the arena game mode

MEMBER FUNCTIONS - USER					
Signature	**Static**	**Virtual**	**Operator**	**Friend**	**Description**
User(int,vector<Character>,int,int);	no	no	no	no	Constructor for a User object.
void set_ currency (int);	no	no	no	no	Set the currency of the user in session.

int get_ currency ();	no	no	no	no	Get the currency of the user in session.
void set_arena_ wins(int);	no	no	no	no	Set the arena wins of the user in session.
int get_arena_ wins() ;	no	no	no	no	Get the arena wins of the user in session.
void set_arena_ losses(int);	no	no	no	no	Set the arena losses of the user in session.
int get_arena_ losses();	no	no	no	no	Get the arena losses of the user in session.
int count_ characters();	no	no	no	no	Returns the current number of characters on player_box (max 10)
void add_ character(Charact er);	no	no	no	no	Add a character to player_box.
void delete_ character(int);	no	no	no	no	Delete a character from player_box.
void show_box();	no	no	no	no	Show the characters in player_box.
friend multiplier (User,Game)	no	no	no	yes	Increases the currency of a user object based on a win streak, works in conjunction with friend function in Game class.
Character retrieve_chara cter(int);	no	no	no	no	Get a specific character from the user's player box.
void write_to_file(stri ng);	no	no	no	no	Record the current user's data into the given filename.

GAME CLASS
Abstract class: No
Subclass of: N/A
Composed of: Character
Alterations post proposal:

- The data member banner was changed to static. This change allows the banner data member to be worked with the new function create_banner as well as the existing gacha_roll function both of which are static. Thus, by allowing banner to be static it simplified the process of keeping gacha/banner related functionalities together.
- The gacha_roll and arena functions were both given a return type of User and two parameters User and string. The first change was necessary to keep the changes made by these functions accurate to what the user sees displayed on screen. The second change is related to the user in session as it was needed to pass the current user object and record into its own personal file every time these functions were called as they are critical to the overall functionality of the game.
- A create_banner function was added. This function simplified the process of providing a brand new banner every time a gacha_roll session is initiated.
- The function check_session was added to deal with login issues that could arise at first launch of the game. Basically, check_session is in charge of checking if the user that trying to login into session already exists or not. If it exists then it returns true, otherwise false.

DATA MEMBERS - GAME			
Variable Name	**Data Type**	**Static**	**Description**
static vector<Character> banner	Character	yes	A collection of unique characters that have rate of appearance from which the user obtains the character.
mode	int	no	An integer representing the three possible modes in a game session: home, gacha and arena.
username	string	no	The name of the user that we want to access in session, or a new user.
state	bool	no	The game state, a boolean variable that is true while the game is running, if it is false it will terminate the game.
battle_mode	int	no	An integer representing the two possible sub modes of the arena mode.

streak	int	no	The win streak in arena of a particular session.

MEMBER FUNCTIONS - GAME					
Signature	**Static**	**Virtual**	**Operator**	**Friend**	**Description**
Game(int,string, bool,int, int);	no	no	no	no	Constructor for a Game object.
void set_mode (int);	no	no	no	no	Set the current game mode of the user in session.
int get_mode ();	no	no	no	no	Get the current game mode of the user in session.
void set_ username (string);	no	no	no	no	Set username to log into a session provided by the user.
string get_ username() ;	no	no	no	no	Get the username of a user.
void set_state (bool);	no	no	no	no	Set the state of the game.
bool get_state();	no	no	no	no	Get the state of the game.
void set_battle_mode ();	no	no	no	no	Set the current battle mode that the user is playing in arena.
int get_battle_mode( )	no	no	no	no	Get the latest battle mode played in arena.
void set_streak();	no	no	no	no	Set the streak of the user in game.
int get_streak();	no	no	no	no	Get the streak of the user in game.

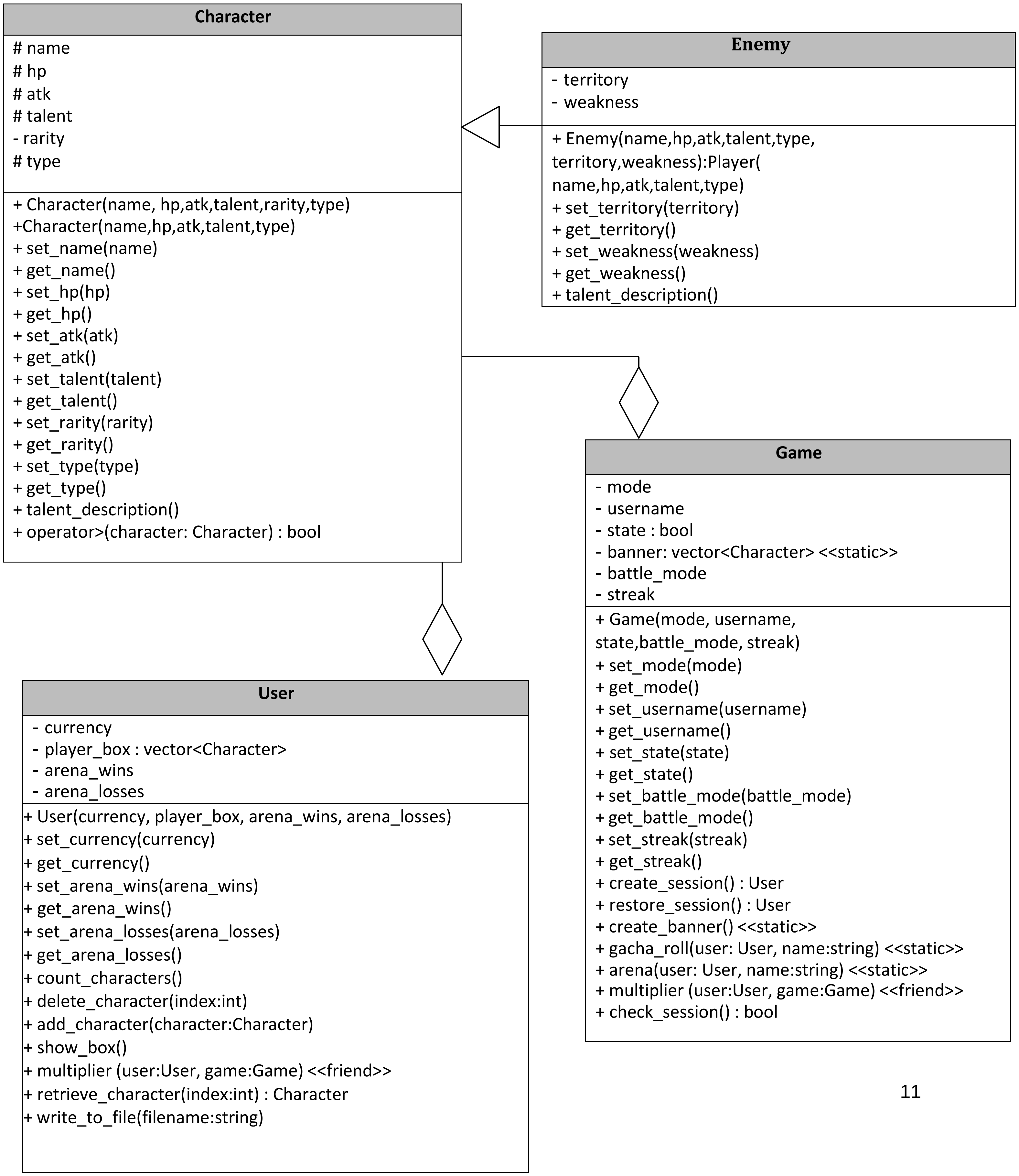
User create session();	no	no	no	no	If it is a new user it will create a new user text file to store the information of the new user.
User restore_ session();	no	no	no	no	If it is an existing user, it will restore the user text file and store the information.
static User gacha_roll (User, string);	yes	no	no	no	it will create the banner and take a user object allowing the user to get characters using currency.
static User arena(User, string);	yes	no	no	no	The arena mode. This function will be in charge of arena operations using the user's character and the enemy's text file.
friend multiplier (User,Game)	no	no	no	yes	Increases the currency of a user object based on a win streak, works in conjunction with friend function in Game class.
static void create_banner ();	yes	no	no	no	Creates a banner for a gacha_roll session.
bool check_session();	no	no	no	no	Check if the user trying to login into the game exists. If it exists, return true, otherwise false.

DEMONSTRATION OF OOP CONCEPTS

Demonstrated in...								
File name	Line #s	**Encapsulation**	**Inheritance**	**Polymorphism**	**Static Members**	**Friend Functions**	**Overloaded Operators**	**Text files**
Character.h	7-14	X						
Enemy.h	8-10	X						

Game.h	10-16	X						
User.h	12-16	X						
Enemy.h	7		X					
Character.h	35			X				
Enemy.cpp	27-54			X				
Game.h	11,36, 40-41				X			
multiplier.cpp	7-22					X		
Game.h	43					X		
User.h	37					X		
Character.cpp	98-121						X	
Game.cpp	65-67 134-135 344-345							X
Game.cpp	583			X				
User.cpp	99-117							X
Game.cpp	297						X	
Main.cpp	208,215					X		
Main.cpp	192,201				X			

UML DIAGRAM


Character
# name
# hp
# atk
# talent
- rarity
# type
+ Character(name, hp,atk,talent,rarity,type)
+Character(name,hp,atk,talent,type)
+ set_name(name)
+ get_name()
+ set_hp(hp)
+ get_hp()
+ set_atk(atk)
+ get_atk()
+ set_talent(talent)
+ get_talent()
+ set_rarity(rarity)
+ get_rarity()
+ set_type(type)
+ get_type()
+ talent_description()
+ operator>(character: Character) : bool
Enemy
- territory
- weakness
+ Enemy(name,hp,atk,talent,type, territory,weakness):Player(name,hp,atk,talent,type)
+ set_territory(territory)
+ get_territory()
+ set_weakness(weakness)
+ get_weakness()
+ talent_description()
Game
- mode
- username
- state : bool
- banner: vector<Character> <<static>>
- battle_mode
- streak
+ Game(mode, username, state,battle_mode, streak)
+ set_mode(mode)
+ get_mode()
+ set_username(username)
+ get_username()
+ set_state(state)
+ get_state()
+ set_battle_mode(battle_mode)
+ get_battle_mode()
+ set_streak(streak)
+ get_streak()
+ create_session() : User
+ restore_session() : User
+ create_banner() <<static>>
+ gacha_roll(user: User, name:string) <<static>>
+ arena(user: User, name:string) <<static>>
+ multiplier (user:User, game:Game) <<friend>>
+ check_session() : bool
User
- currency
- player_box : vector<Character>
- arena_wins
- arena_losses
+ User(currency, player_box, arena_wins, arena_losses)
+ set_currency(currency)
+ get_currency()
+ set_arena_wins(arena_wins)
+ get_arena_wins()
+ set_arena_losses(arena_losses)
+ get_arena_losses()
+ count_characters()
+ delete_character(index:int)
+ add_character(character:Character)
+ show_box()
+ multiplier (user:User, game:Game) <<friend>>
+ retrieve_character(index:int) : Character
+ write_to_file(filename:string)