Design Overview for RPG Adventure

Name: Tran Thanh Minh
Student ID: 103809048

Summary of Program

Describe what you want the program to do… one or two paragraphs.
I want to design a RolePlay program based on Swin Adventure. It is the text-based game where the user will enter the command to play.
I also will implement the menu and monsters, …

Include a sketch of sample output to illustrate your idea.

Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: Enemy type enumerations details

Value	Notes
Monster	For class Monster
Boss	For class Boss

Table 2: interface IspecialAbilityUser details

Responsibility	Type Details	Notes
Can generate the powerful attack	Print out text	

Table 3: abstract Enemy class details

Responsibility	Type Details	Notes
Declare the public variables for the stats of the Enemy	Parameter : string name, int health, int attack , int defense, int gold, int exp	default
Take damage(int hurt)	Type : void Parameter : int hurt	Reduce the current health
IsDeath()	Type : bool Return true || false	Check whenever the enemy is dead or not
Enemyinfo()	Type: public Return string	Return all the information of the enemy

Table 4: abstract EnemyWithSpecialAbility details – duplicate

Responsibility	Type Details	Notes
UseSpecialAbility()	Type: void	
EnemyWithSpecialAbility	Default Public and inherited from Enemy	Default inherited from Enemy and use the Ispecialabilityuser

Table 5: public abstract Enemy details

Responsibility	Type Details	Notes
Name	Return string Public	Public property
Health	Return int Public	Public property
Attack	Return int Public	Public property
Defense	Return int Public	Public property
MaxHealth	Return int Public	Public property
Gold	Return int Public	Public property
Expgain	Return int Public	Public property
Enemy()	Parameters (string name, int health, int attack, int defense, int gold,int exp)	Default constructor
TakeDamage()	Parameter: int hurt Public Type: void	Decrease the amount of health for the enemy
IsDeath()	Public Return bool	Check if the enemy's health is below 0 or not
EnemyInfo()	Public Return string	Print out the current info of the boss

Table 6: Action details

Responsibility	Type Details	Notes
Attack()	Type: public static void Parameters: player, enemy	Attack the enemy
Defend()	Type: public static void Parameters: player, enemy	Defend against the enemy
Heal()	Type: public static void Parameters: player, enemy	Heal the player
EnemyAttack()	Type: public static void Parameters: player, enemy	Enemy attack

checkExp()	Type: public static void Parameters: player	Check if exp is enough to level up
LevelUp	Type: public static void Parameters: player	Level up the player

Table 7: Boss details

Responsibility	Type Details	Notes
Boss()	Type: public	Default constructor and inherited form the EnemyWithSpecialAbility
UseSpecialAbility()	Type: public override void	Print out the skills

Table 1: Program details

Responsibility	Type Details	Notes
Main()	Type: public static void Parameter: string[] args	Main program, print out the title of the game and run the program

Table 8: GamePlay details

Responsibility	Type Details	Notes
InitStates()	Type: private void	Create new states and push new mainmenu state
InitPlayer()	Type: private void	Create new playerlist
End	Type: private Return End	
Initial()	Type: private void	Set the end to false
GamePlayer()	Type: public	Default constructors and includes other Inits methods.
Run()	Type: public void	Keep the states keep updating. If there is no states, exit the program.

Table 9: State details

Responsibility	Type Details	Notes
State()	Parameters: stack<state> states	Default constructor
wantEnd()	Type: public bool Return bool	For exit the menu
Update()	Type: public virtual void	

Table 10: Main menu state details

Responsibility	Type Details	Notes
MainMenuState()	Type: public Parameters: Stack<State> states, ArrayList playerlist	Default constructors to set the local variables to equal to the parameters

Process()	Type: private void Parameter: string num	For the menu to decide on the input of the user
Update()	Type: public override void	Keep update the menu state
NewGame()	Type: private void	Process when player choose new game
Choose player()	Type: private void	To set the currentplayer for later use.

Table 11: Player State details

Responsibility	Type Details	Notes
PlayerState()	Type: public Parameters: Stack<State> states, Player player	Default constructor to set the local variables to equal to the parameters
Process()	Type: private void Parameter: string num	For process the input choice
CreatePlayer()	Type: private void	For create new player
EditPlayer()	Type: private void	For edit the stats of player
Rename()	Type: private void	
GetNewValue()	Type: private Return string Parameters: string message, string currentvalue	For get a new name
GetNewLevel()	Type: private int Parameter: string message Return int	For getting experience for leveling up
DeletePlayer()	Type: private void	For deleting player in arraylist
Update()	Type: public override void	For printing out the menu for the player

Table 22: GameState details

Responsibility	Type Details	Notes
GameState()	Type: public Parameters: Stack<State> states, Player currentplayer	Default constructor to set the local variables to the parameters
Process()	Type: private void Parameter(string num)	Process based on the input
Update()	Type: public override void	Printing out the menu content for the current state
Story()	Type: private static void Parameter: player player	Printing out the story

Table 33: SceneState details

Responsibility	Type Details	Notes
SceneState()	Type: public	Default constructor

	Parameters: Stack<State> states, Player player	
Process	Type: private void Parameter: string num	Input menu choice
Update()	Type: public override void	Keep updating method
Displayshop()	Type: private void Parameter: Player currentp	Include the ShopInstructions and SaveShop
ShopInstruction()	Type: private void	Print out shop content
SaveShop()	Type: private void Parameter: Player player	Call the Buy depend on the input of the user
Buy()	Type: private static void Parameters: string item, int cost, Player player	Check if the user's coin is enough
Equipment()	Type: private static void Parameters: Player player	

Table 44: EnemyState details

Responsibility	Type Details	Notes
EnemyState()	Type: public	Default constructor
Process	Type: private void Parameter: string num	Input menu choice
Update()	Type: public override void	Keep updating method
Run()	Type: private void	If choice was run
DeathFlag()	Type: private void	If the player health is return 0

Table 55: Program details

Responsibility	Type Details	Notes
Main	Type: public static void Parameter: string[] args	Main program

Table 66: Player details

Responsibility	Type Details	Notes
Player()	Type: public Parameters: string namein, string descrip	Default constructor
Stats()	Type: private void	Set the initial value
Exp	Type: public Return int	Property
MaxExp	Type: public Return int	Property
Health	Type: public Return int	Property
MaxHealth	Type: public	Property

	Return int	
Name	Type: public Return string	Print out the name and desc
NamePlayer	Type: public Return string	Property
Descplayer	Type: public Return string	Property
Level	Type: public Return int	Property
Defend	Type: public Return int	Property
Potion	Type: public Return int	Property
Gold	Type: public Return int	Property
Damage	Type: public Return int	Property
DamageMax	Type: public Return int	Property
Weapon	Type: public Return string	Property
WeaponDmg	Type: public Return int	Property
Banner()	Type: public Return string	Display player info
ToString()	Type: public override Return string	Display player info
AllInfo()	Type: public Return string	Display player full info
Win()	Type: public void Parameter: Enemy enemy	If the player win
TakeDamage()	Type: public void Parameter: int damage	If the player take damage
UseHeal()	Type: public void	If the player use potion to heal
IsDefending	Type: public Return bool	If the player is defend or not

Table 77: Monster details

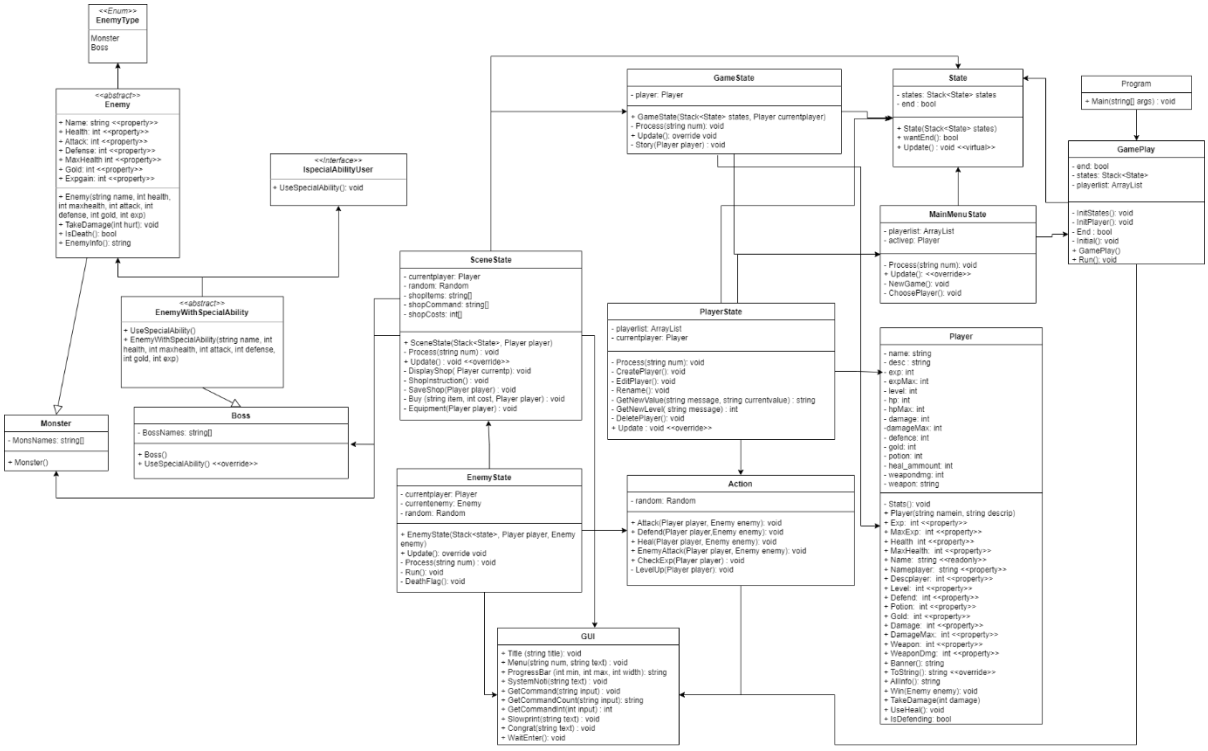
Responsibility	Type Details	Notes
Monster	Type: public	Default constructor

Table 88: GUI details

Responsibility	Type Details	Notes
Title()	Type: public static void	

	Parameter: string title	
Menu()	Type: public static void Parameters: string num, string text	
ProgressBar()	Type: public static Parameters: int min, int max, int width	
SystemNoti()	Type: public static void Parameter: string text	
GetCommand()	Type: public static void Parameter: string input	
GetCommandCount()	Type: public static Parameter: string input Return string	
GetCommandInt()	Type: public static Parameter: int input Return int	
Slowprint()	Type: public static void Parameter: string text	
Congrat()	Type: public static void Parameter: string text	
WaitEnter()	Type: public static void	

Class Diagram



Sequence Diagram