

Monster Defeated

Instruction

This project is made for subject 2110215 Programming Methodology (2018/1).

Authors : -Natthanon Manop 6031013021

-Yanika Dontong 6031010021

1.Application Flow

This application is a turn based game which have 15 stages. The look of the game main menu is shown in figure 1. There are game's name and label which when the cursor pass by the label change to begin in figure 2. when click it, it will bring you to hero selection scene in figure 3.

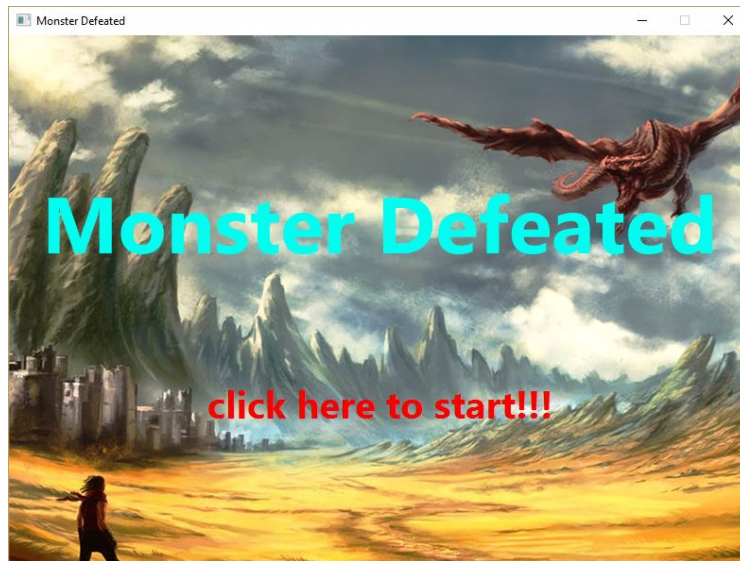


Figure 1: The game's main menu.

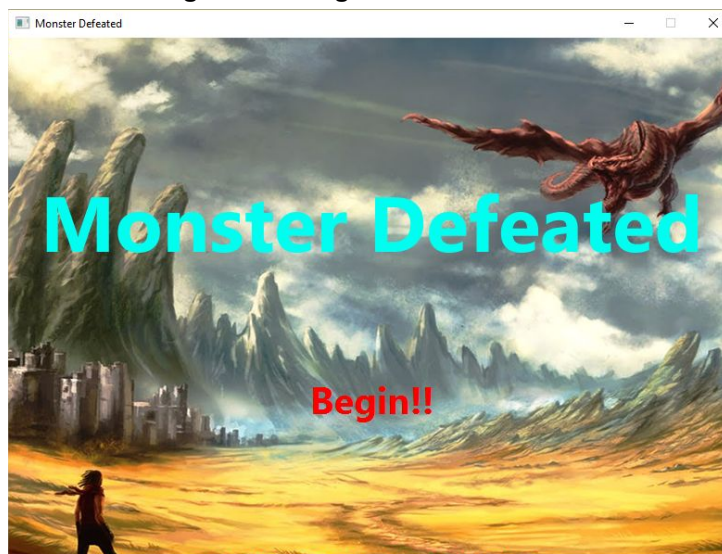


Figure 2: The game's main menu after cursor point at click here to start.

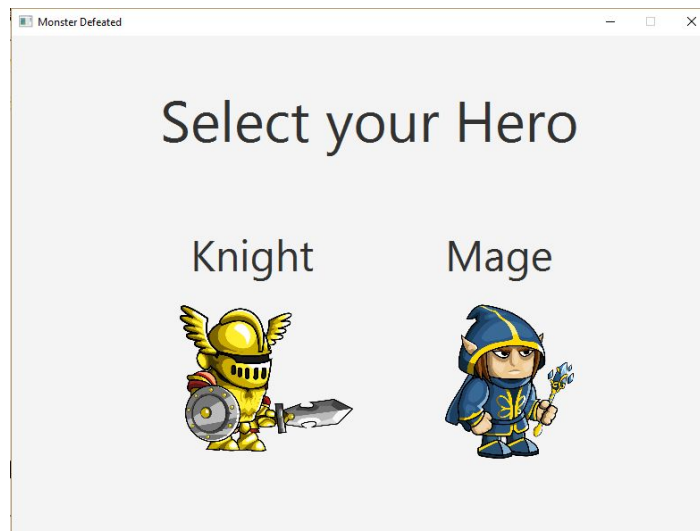


figure 3: Hero selection scene

Each hero has different status and attack pattern. After you choose your hero the battle scene will show up. It has the hero status and monsters status in figure 6.

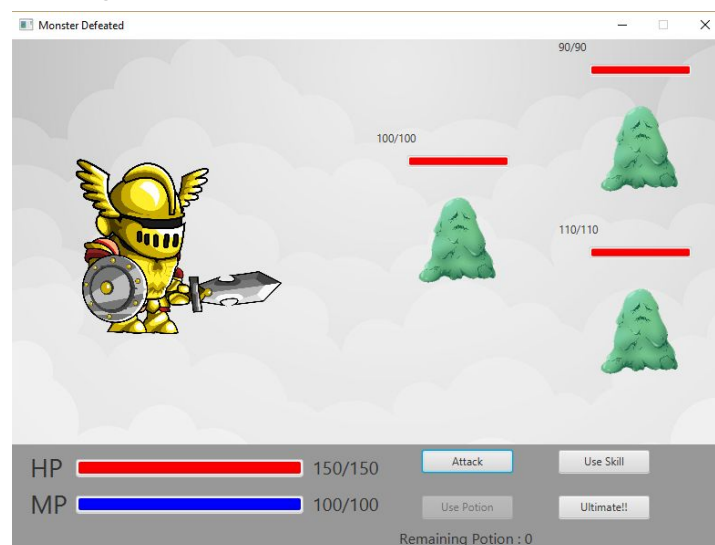


figure 4: Battle scene

There are 4 button in the scene. When you click Attack button, Use Skill button and Ultimate button, it will decrease monster health by type of hero and action that you selected. After that hero's health decrease by monsters' attack so the turn end and in each turn hero's mana increase. You can see remaining potions below the Use Potion button. When you push Use Potion button Hero health's will increase and the turn will end. When you clear a stage, a few potion will be dropped.

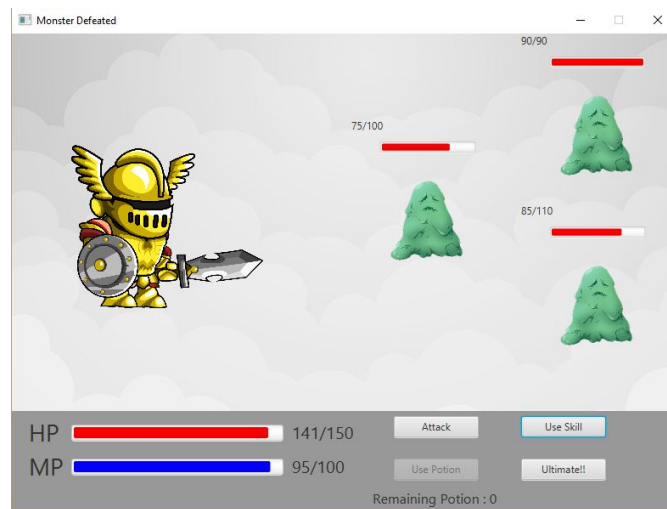


figure 5: When hero attack monster.

When you hero's health reach 0, your hero will die, the alert will pop up.

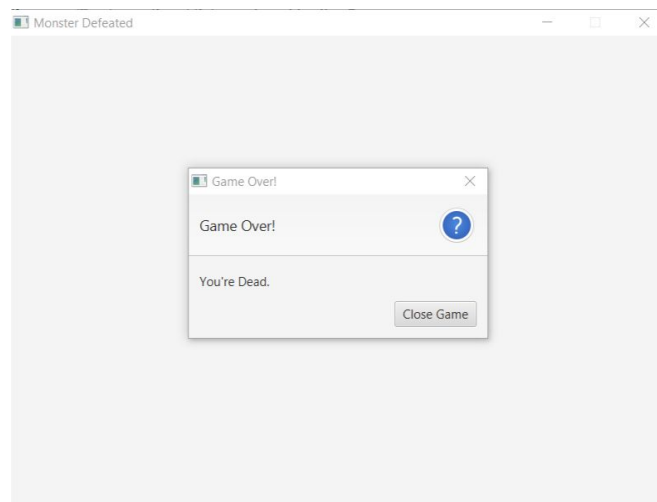


figure 6: The game over alert.

When you clear all the stage, the alert will pop up.

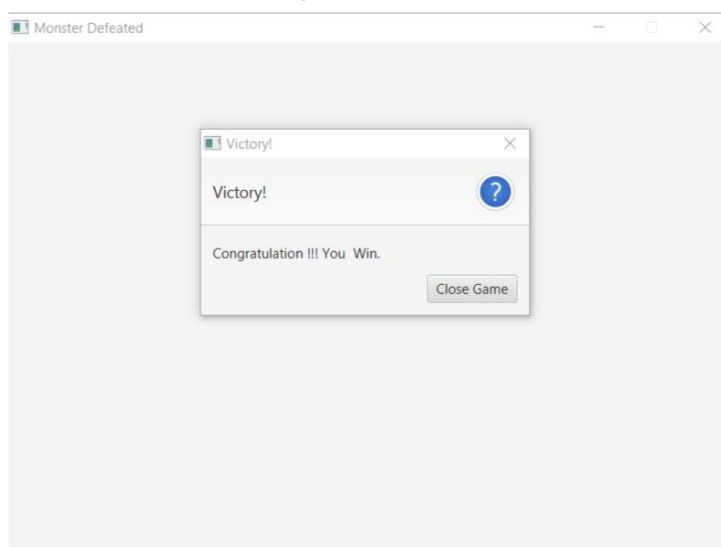
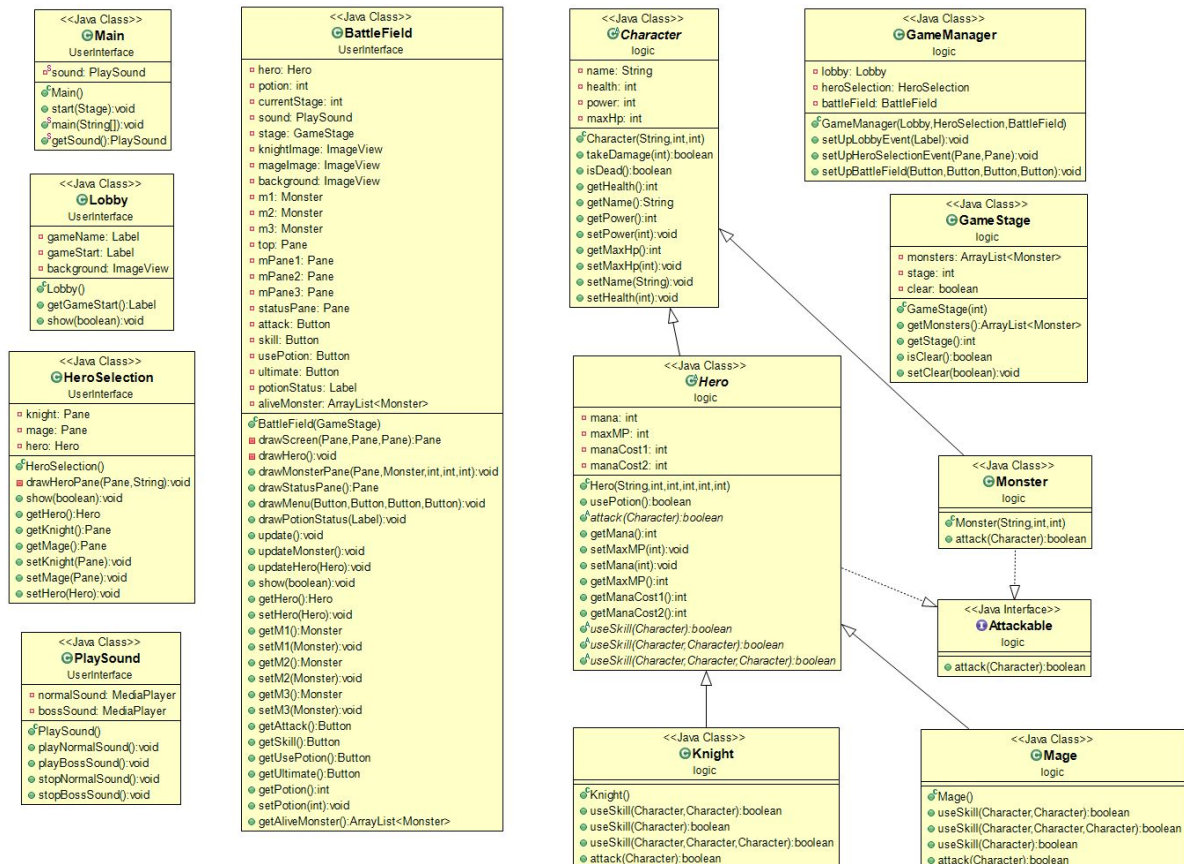


figure 7: The congratulation alert.

2.Implementation Details



*Noted that Access Modifier Notations can be listed below

- + (public)
- # (protected)
- (private)

static will be underlined.

abstract will be italic.

2.1 Class *Logic.Character*

2.1.1 Field

- String name	This character's name.
- int health	This character's current health.
- int power	This character's attacking power; how much of target's health will reduced after attacked by this character.
- int maxHP	This character's maximum health. This character's health can't exceed this value.

2.1.2 Constructor

+ Character(String name,int maxHP,int power)	Initializes the every field of the class. Current character's health is set to maximum.
--	---

2.1.3 Method

+ boolean takeDamage(int damage)	When character is taking damage, the damage should be reduced its health. After taking the damage, return true if the health is equal or below 0, return false otherwise.
+ boolean isDead(Character c)	Return true if this character's health is no more than zero.
Getters&Setters	Every fields has its getter and setter.

2.2 Class *Logic.Hero extends Character implement Attackable*

2.2.1 Field

- int mana	This hero's current mana
- int maxMP	This hero's maximum mana.This hero's mana can't exceed this value.
- manaCost1	Mana cost when use skill.
- manaCost2	Mana cost when use ultimate.

2.2.2 Constructor

+ Hero(String name,int maxHP,int power,int maxMP,int manaCost1,int manaCost2)	Initialize all fields,current hero's mana is set to maximum.
---	--

2.2.3 Method

+ boolean usePotion()	In each stage, if hero Potion,
-----------------------	--------------------------------

	hero's current health increasing by 50 but not exceed max health then return true. If current health equals to max health or hero is dead return false.
<i>+ boolean Attack(Character target)</i>	
<i>+ boolean useSkill(Character monster)</i>	
<i>+ boolean useSkill(Character monster1, Character monster2)</i>	
<i>+ boolean useSkill(Character monster1, Character monster2, Character monster3)</i>	
Getters&Setters	Every field has its getter and setter for Mana and MaxMP.

2.3 Class Logic.Knight extends Hero

2.3.1 Constructor

<i>+ Knight()</i>	Initialize all fields.
-------------------	------------------------

2.3.2 Method

<i>+ boolean useSkill(Character monster1, Character monster2)</i>	This is use skill. If knight is dead return false. If knight use skill, monster 1 and monster 2 take damage by 25 and knight's mana decrease by 10 then return true.
<i>+ boolean useSkill(Character monster)</i>	This is ultimate. If knight is dead return false. If knight use ultimate, monster take damage by 100 and knight's mana decrease by 50 then return true.
<i>+ boolean useSkill(Character monster1, Character monster2,</i>	To implement useSkill(Character monster1, Character monster2,

Character monster3)	Character monster3) in Hero class. Return false anyway.
+ boolean attack(Character target)	This is normal attack. If target is dead return false. If knight attacks, target takes damage by knight's power then return true.

2.4 Class Logic.Mage extends Hero

2.4.1 Constructor

+ Mage()	Initialize all fields.
----------	------------------------

2.4.2 Method

+ boolean useSkill(Character monster1, Character monster2)	This is use skill. If mage is dead return false. If mage use skill, monster 1 and monster 2 take damage by 30 and mage's mana decrease by 15 then return true.
+ boolean useSkill(Character monster1, Character monster2, Character monster3)	This is ultimate. If mage is dead return false. If mage use ultimate, monster 1, monster 2 and monster 3 take damage by 40 and mage's mana decrease by 45 then return true.
+ boolean useSkill(Character monster)	To implement useSkill(Character monster) in Hero class. Return false anyway.
+ boolean attack(Character target)	This is normal attack. If target is dead return false. If mage attacks, target takes damage by mage's power then return true. Otherwise return false.

2.5 Class Logic.Monster extends Character implements Attackable

2.5.1 Constructor

+ Monster (String name, int maxHP, int power)	Initialize all fields.
+ boolean attack(Character target)	This is monster attack. If target is dead return false. If monster attacks, target takes damage by monster's power then return true. Otherwise return false.

2.6 Class Logic.GameStage

2.5.1 Field

- ArrayList<Monster> monsters	Declare the list of all monsters in stage.
- int stage	Declare current stage.
- boolean clear	Declare stage is clear or not.

2.5.2 Constructor

+ Stage(int stage)	initialize list of all monsters in each stage.
--------------------	--

2.5.3 Method

+ isClear()	return true if this stage is cleared.
Getters&Setters	getters for monsters and stage and setter for clear.

2.7 Class Logic.GameManager

2.7.1 Field

- Lobby lobby	An instance of Lobby
- HeroSelection heroSelection	An instance of HeroSelection
- BattleField battleField	An instance of BattleField

2.7.2 Constructor

+ GameManager(Lobby lobby, HeroSelection heroSelection,	Initializes the EventManager. This class is used to bundle up the
---	---

BattleField battleFiled)	event handlers for whole game.
--------------------------	--------------------------------

2.7.3 Method

+ setUpLobbyEvent(Label gameStart)	Sets up the event handler for the gameStart label. This method is used to enter hero selection scene. This method should be called from the Main class by passing the gameStart label.
+ setUpHeroSelectionEvent(Pane pane1, Pane pane2)	Sets up the event handler for the pane1 Pane and pane2 Pane. This method is choosing hero then enter battlefield scene. This method should be called from the Main class by passing the pane1(which is knight) and pane2(which is mage).
+ setUpBattleField(Button attack, Button skill, Button usePotion, Button ultimate)	Sets up the event handler for game menu. This method is used to choose action in battlefield. This method should be called from the Main class by passing the attack Button, skill Button, usePotion Button and ultimate Button.

2.8 Interface Logic.Attackable

2.8.1 Method

+ boolean attack(Character c)	
-------------------------------	--

2.9 Class UserInterface.Lobby extends StackPane

2.9.1 Field

- Label gameName	Label for show game name.
- Label gameStart	Label for show game start that click to enter hero selection scene.
- ImageView background	Image for show background.

2.9.2 Constructor

+ Lobby()	Initializes the Lobby. This method add background Image, game name label and game start label to lobby scene.
-----------	---

2.9.3 Method

+ void show(boolean show)	If show is true sets this scene visible. If show is false sets this scene invisible.
Getters	gameStart getter.

2.10 Class UserInterface.HeroSelection extends Pane

2.10.1 Field

- Pane knight	Pane for show knight text and image.
- Pane mage	Pane for show mage text and image.
- Hero hero	Hero that player choose.

2.10.2 Constructor

+ HeroSelection()	Initializes the HeroSelection. This method set positions and alignment of all elements and add to hero selection scene.
-------------------	---

2.10.3 Method

+ void drawHeroPane(Pane pane, String type)	This method have been called by constructor. This sets all position and alignment of knight and mage Pane.
+ void show(boolean show)	If show is true sets this scene

	visible. If show is false sets this scene invisible.
Getters&Setters	Every fields has its getter and setter.

2.11 Class UserInterface.BattleField extends Pane

2.11.1 Field

- Hero hero	hero which player choose.
- int potion	show how much potion left.initialize 1.
- currentStage	which stage that player is playig.initialize 2.
- PlaySound sound	background music.
- GameStage stage	stage that player at.
- ImageView knightImage	knight's picture.
- ImageView mageImage	mage's picture.
- ImageView background	background's picture.
- Monster m1	first monster.
- Monster m2	second monster.
- Monster m3	third monster.
- Pane top	Pane that show hero and monsters.
- Pane mPane1	Pane for first monster.
- Pane mPane2	Pane for second monster.
- Pane mPane3	Pane for third monster.
- Pane StatusPane	Pane that show hero's current health and mana.
- Button attack	Button for attack monster.
- Button skill	Button for hero to use skill.

- Button usePotion	Button for hero to use potion to gain hero's health.
- Button ultimate	Button for hero to use ultimate.
- Label potionStatus	Label which show how many potion left.
- ArrayList<Monster> aliveMonster	List which contain alive monsters.

2.11.2 Constructor

+ BattleField(GameStage stage)	Initializes the BattleField. This method set positions and alignment of all elements and add to battle field scene.
--------------------------------	---

2.11.3 Method

- Pane drawScreen(Pane mPane1, Pane mPane2, Pane mPane3)	Set position and alignment of three monsters in pane. Return pane of monsters.
- void drawHero()	Set position and alignment of hero image.
- void drawMonsterPane(Pane pane, Monster m, int i, int x, int y)	This method have been called by drawScreen method. To set position and alignment of image and health bar of monster.
- Pane drawStatusPane()	This method have been called by update method. To set position, alignment and value of hero's status.
- void drawMenu(Button attack, Button skill, Button usePotion, Button ultimate)	This method have been called by Constructor. To set position and alignment of buttons.
- void drawPotionStatus(Label potionStatus)	This method have been called by Constructor and update. To set position and alignment of remaining potions.
+ void update()	This method have been called by

	GameManager class when player do action. To update hero's health and mana, monsters' health, current stage and handle event game over and game clear.
+ void updateMonster()	This method have been called by update. To update alive monster in current stage.
+ void updateHero(Hero hero)	This method have been called by GameManager class. To show hero image that player choose.
+ void show(boolean show)	If show is true sets this scene visible. If show is false sets this scene invisible.

2.12 Class UserInterface.PlaySound

2.12.1 Field

- MediaPlayer normalSound	MediaPlayer for play normal stage sound effect.
- MediaPlayer bossSound	MediaPlayer for play boss stage sound effect.

2.12.2 Constructor

+ PlaySound()	Initializes the PlaySound. This method add normal sound and boss sound.
---------------	---

2.12.3 Method

+ playNormalSound()	This method use for play normal stage sound.
+ playBossSound()	This method use for stop normal stage sound.
+ stopNormalSound()	This method use for play boss stage sound.
+ stopBossSound()	This method use for stop boss stage sound.

2.13 Class `UserInterface.Main` extends `Application`

2.13.1 Field

<u>- <code>PlaySound sound</code></u>	PlaySound for background music.
---------------------------------------	---------------------------------

2.13.2 Constructor

+ void <code>start(Stage primaryStage)</code>	The main entry point of the JavaFX application.
---	---

2.13.3 Method

+ void <u><code>main(String[] args)</code></u>	An entry point of the application.
+ <u><code>PlaySound getSound()</code></u>	return sound.