I Love Progmeth

Documentation

Created by

Phatcharapon Jumruspun	6031040921
Phathit Sriburi	6031041521

2110215 Programming Methodology
Semester 1 Year 2018
Chulalongkorn University

I Love Progmeth Documentation

Introduction

I Love Progmeth is a turn-based game. The goal of this game is to find which player will survive. To survive in this game, a player has to fight another player by destroying all another player's characters. When the game begins, each player has to take turn choosing three characters to form their teams. Then, the players have to fight and try to destroy the opposite team. The player who can keep their characters last longer will be the winner of this game.

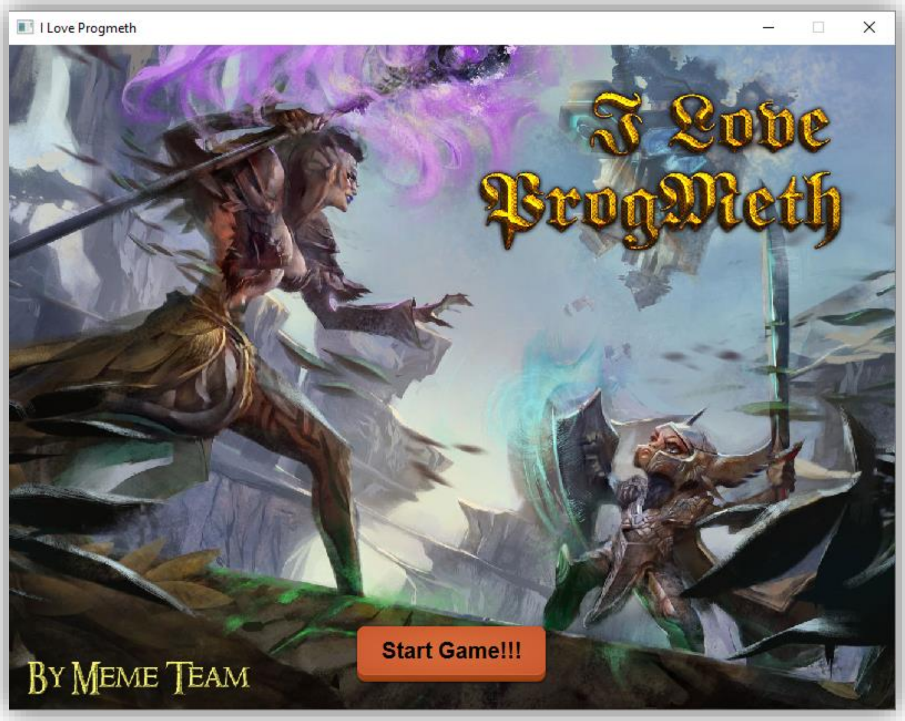
User Manual

How to install

Firstly, download runnable jar file from the link below

https://drive.google.com/open?id=1aSgkXluGHexOWfQfQ7e5N8qgWWnYqW_O

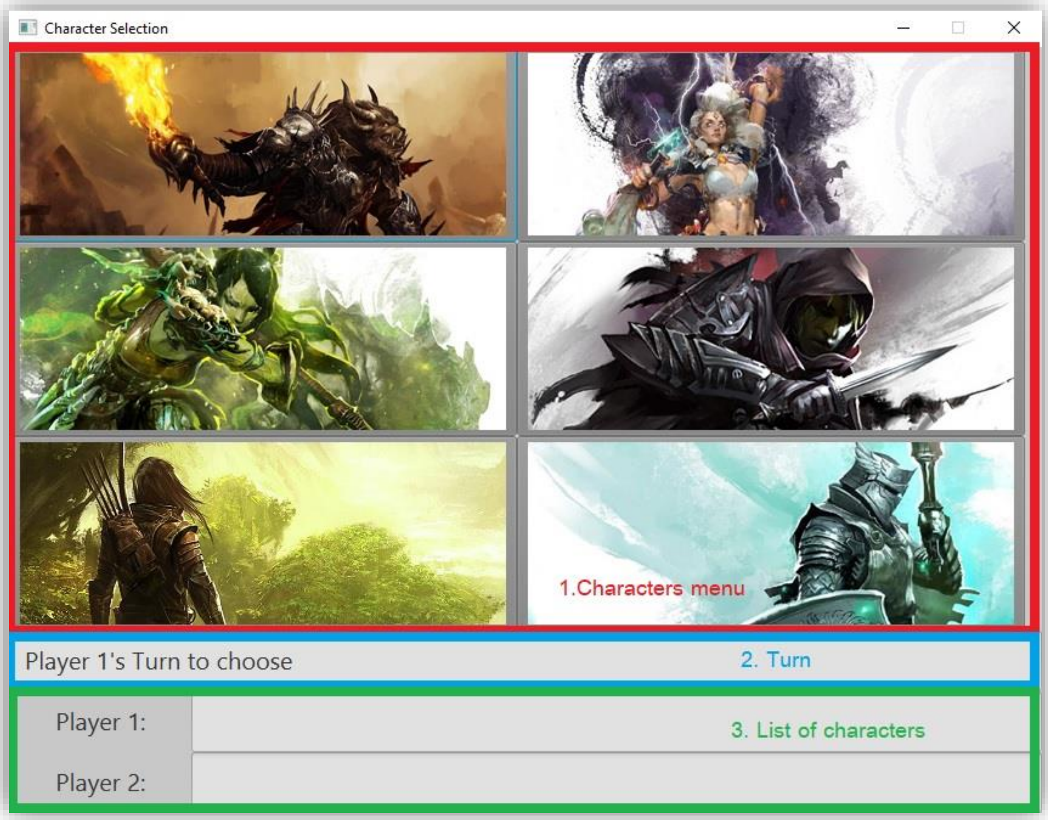
Start Scene



This is the welcome page. To start the game, click “Start Game!!!” button. After you click the button, you will be presented in the CharacterSelection scene.

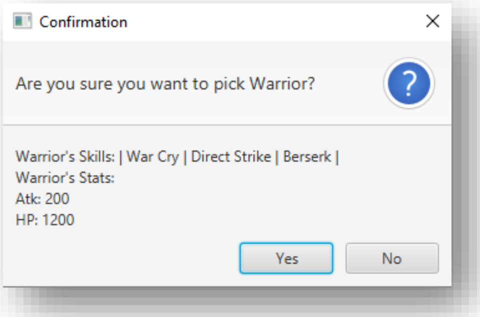
CharacterSelection Scene

The characterselection scene consists of 3 components.



1. Character menu

Character menu is the menu that show all characters in the game. The player can click on the picture of the character to get an information of the character and to confirm choosing the character if you would like to keep the character in your list of characters.


Confirmation

Are you sure you want to pick Warrior?

Warrior's Skills: | War Cry | Direct Strike | Berserk |
Warrior's Stats:
Atk: 200
HP: 1200

This box appears when you click on the picture of a character.

2. Turn

   This component display player's turn to let the player that show in this component choose the character.

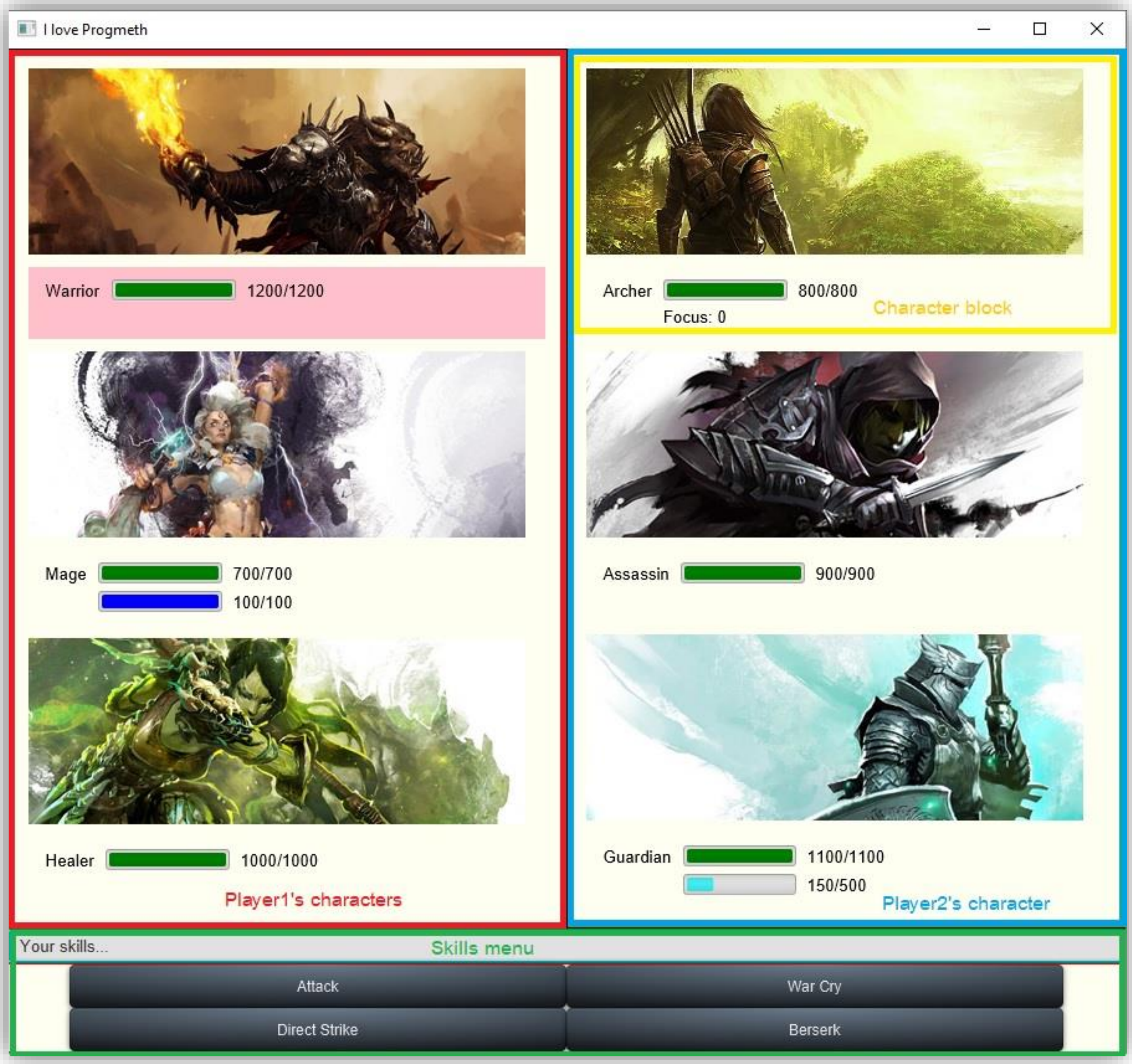
3. List of characters

   List of characters display which character was chosen by player1 or was chosen by player2.

When player1 and player2 completely choose 3 characters in each team, you will be presented in the Battle scene.

Battle Scene

The battle scene consists of 4 major components.



I love Progmeth

Warrior 1200/1200

Mage 700/700
100/100

Healer 1000/1000

Player1's characters

Archer 800/800
Focus: 0

Character block

Assassin 900/900

Guardian 1100/1100
150/500

Player2's character

Your skills...

Skills menu

1. Player1’s characters

This part of a battle scene shows all characters that was picked in characterselection scene by player1.

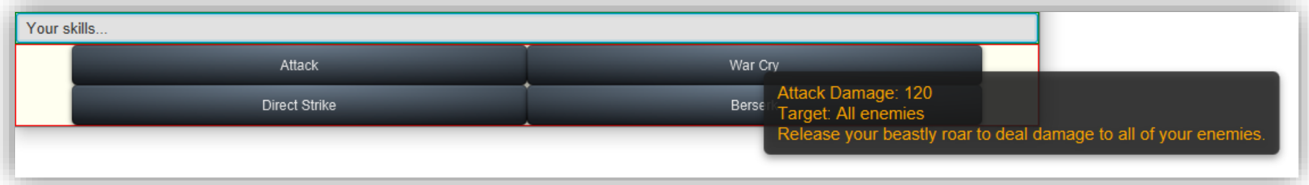
2. Player2’s characters

This part of a battle scene show all characters that was picked in characterselection scene by player2.

3. Skill menu

The skill menu shows all skills of the character that specific by red rectangle

cover the character’s name, character’s hp and character’s status.

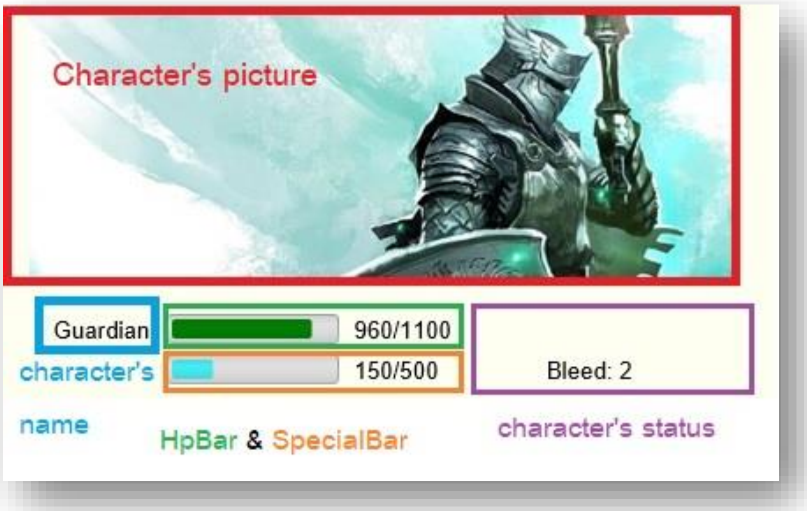


Attack Damage: 120
Target: All enemies
Release your beastly roar to deal damage to all of your enemies.

When you place the cursor on a skill bar, it will show the detail of the skill that you place the cursor on it.

4. Character block

The character block consists of 5 components



1. Character’s picture

   This component show the picture of a character. If the character is dead, the character’s picture will change the color to black and white.

2. Character’s name

   This part displays the character name.

3. HpBar

   The hpbar displays how many health points you have. When the health points of the character drop to zero, it means the character is dead.

4. SpecialBar

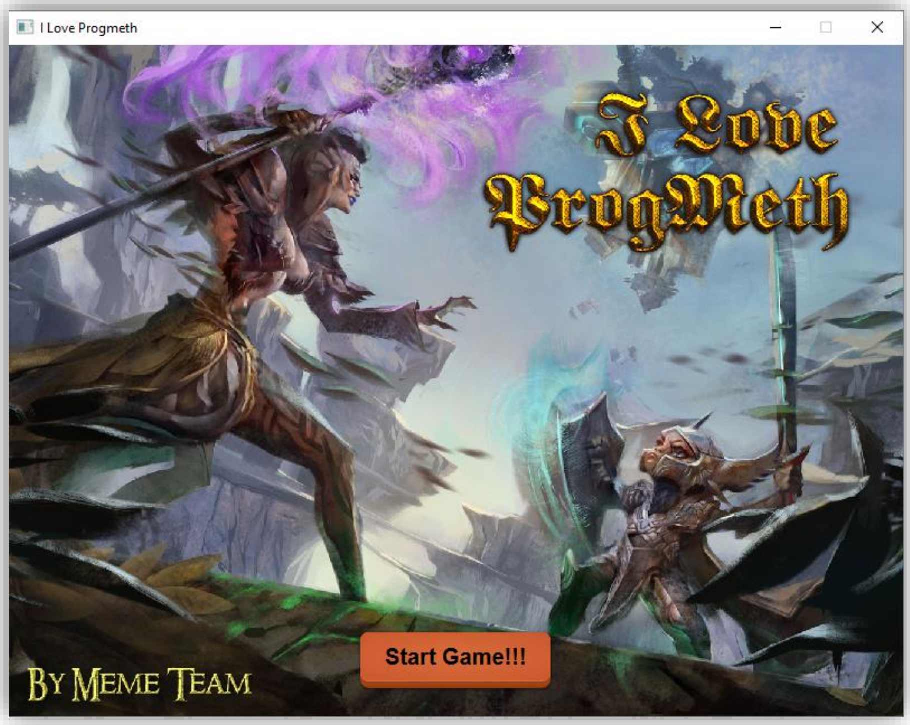
   The specialbar displays how many special points you have ,for example, mana points, focus points or virtues points. The specialbar only displays in some character block that the character has a special ability.

5. Character’s status

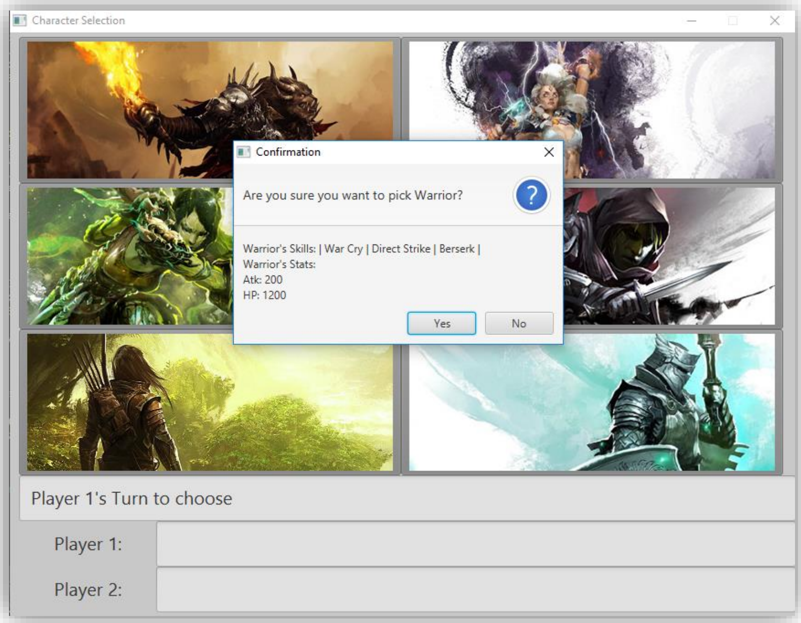
   The character’s status show that your character has any status or not. If your character has some status such as stun, freeze, burn, bleed or shield, it will be shown in this part.

Gameplay

When you enter the game, you will be presented in the start scene



After you have clicked “Start Game!!!” button on the start scene, you will be presented in the characterselection scene. You have to choose 3 characters to be your team and your friend has to choose 3 characters to be another team too by using the guide above.



Character Selection

Confirmation

Are you sure you want to pick Warrior?

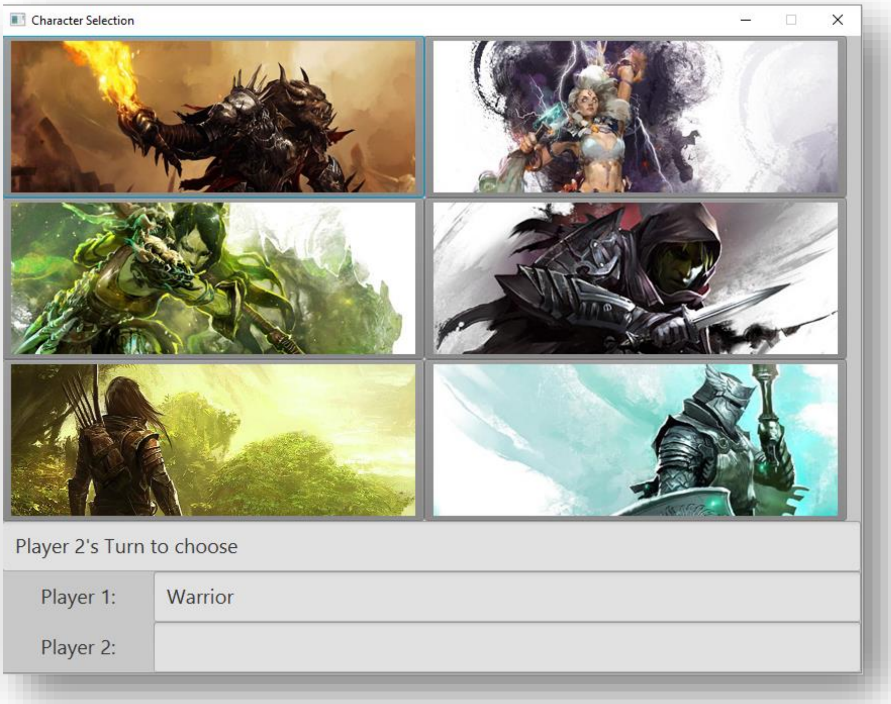
Warrior's Skills: | War Cry | Direct Strike | Berserk |
Warrior's Stats:
Atk: 200
HP: 1200

Player 1's Turn to choose

Player 1:

Player 2:

When player1 select the character, the confirmation will appear to let player1 desire to choose the character or not. If player1 choose the character, the character will be listed in player1’s team and the game will change the turn to player2’s turn.



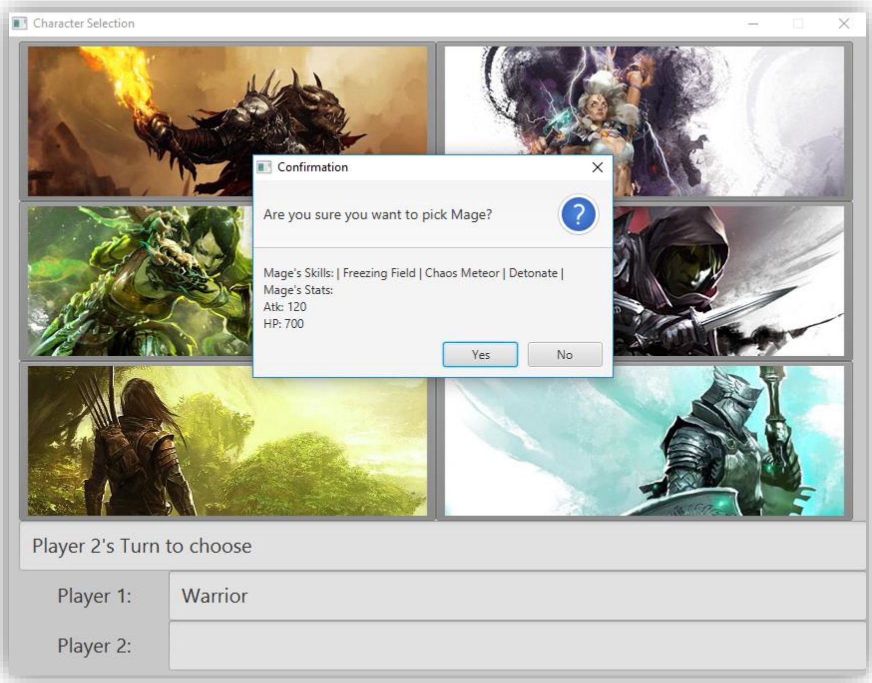
Character Selection

Player 2's Turn to choose

Player 1: Warrior

Player 2:

Player2 has to choose the character like player1 did.



Are you sure you want to pick Mage?

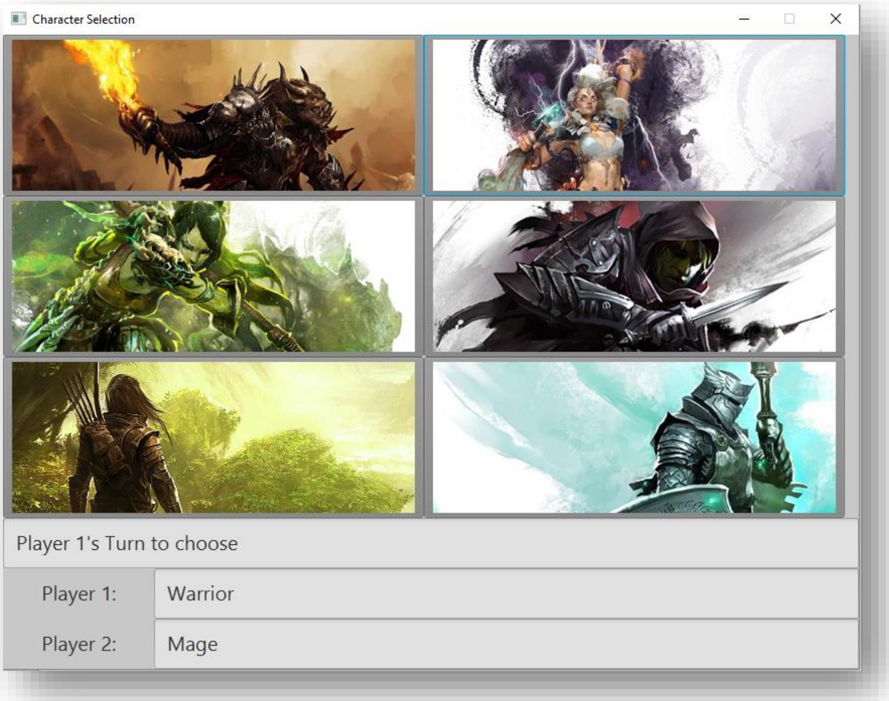
Mage's Skills: | Freezing Field | Chaos Meteor | Detonate |
Mage's Stats:
Atk: 120
HP: 700

Player 2's Turn to choose

Player 1: Warrior

Player 2:

After player2 chose the character, the character will be listed in player2's team and the game will change the turn to player1's turn.



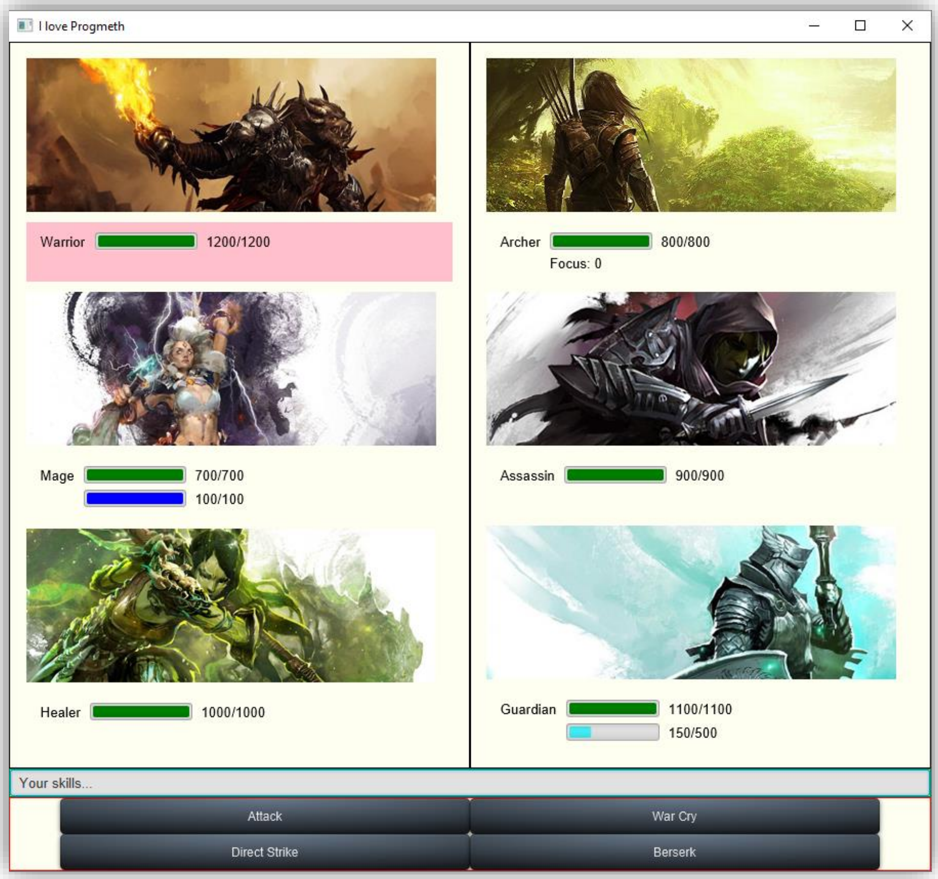
Player 1's Turn to choose

Player 1: Warrior

Player 2: Mage

The game will repeat this sequence until player1's team has 3 characters and player2's team has 3 characters too.

As you and your friend finishing select the characters, you and your friend will be presented in the battle scene.



Warrior 1200/1200

Archer 800/800
Focus: 0

Mage 700/700
100/100

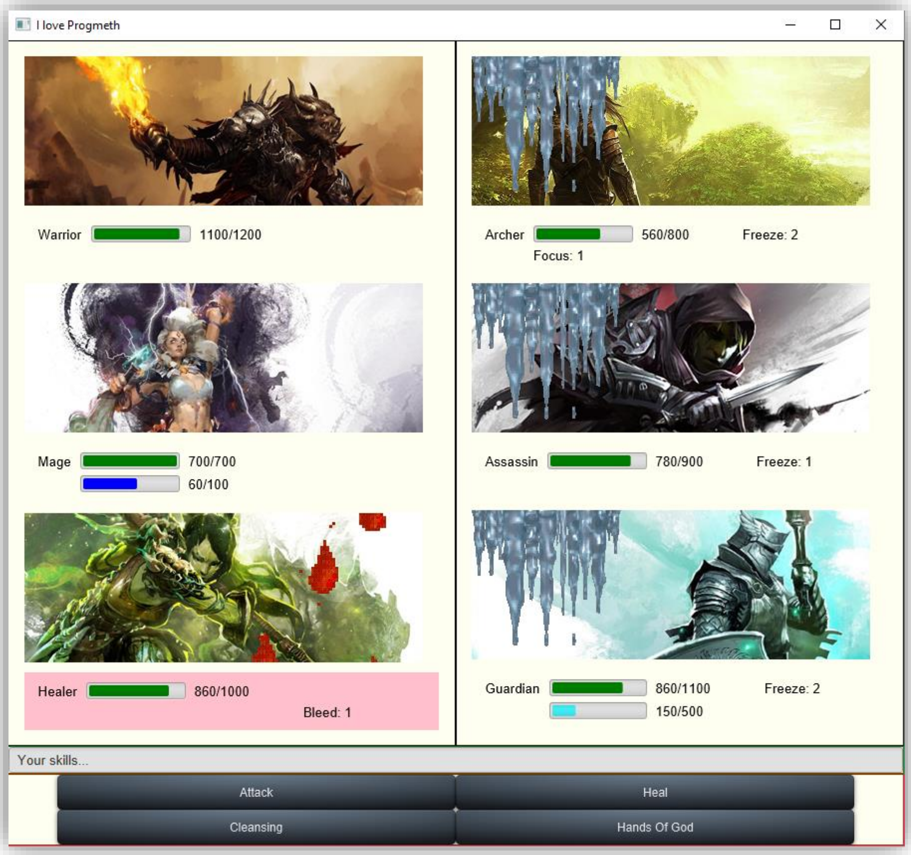
Assassin 900/900

Healer 1000/1000

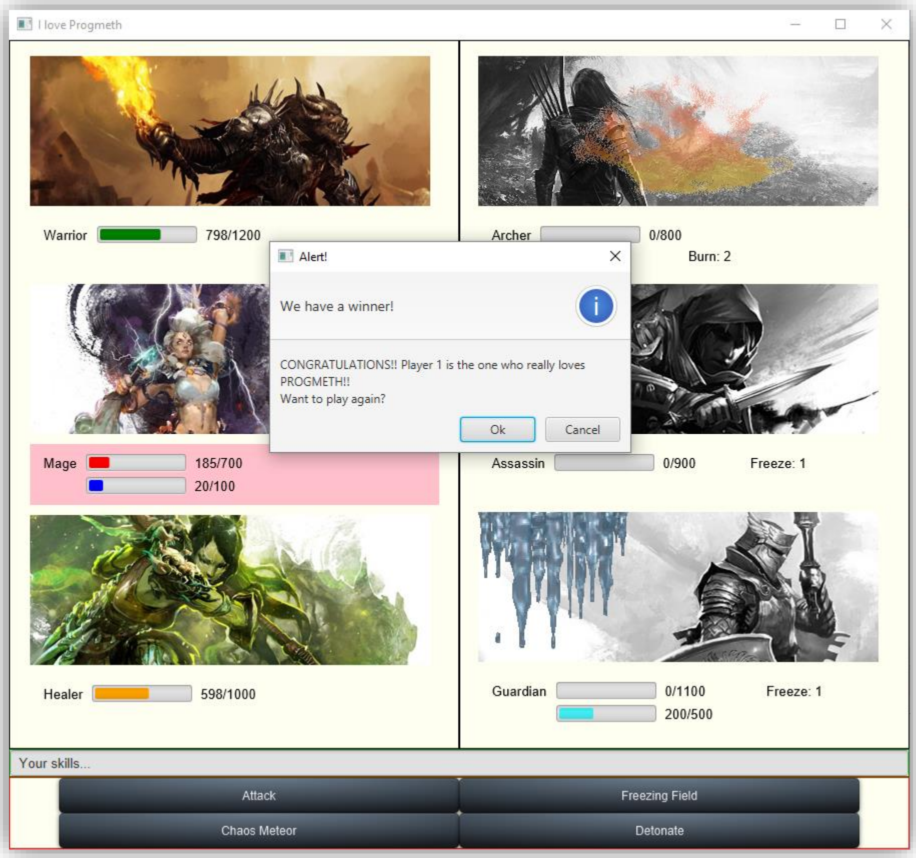
Guardian 1100/1100
150/500

Player1 is the first one who can order his/her character to do something in his/her turn. After player1 complete his/her turn, player2 can order his/her character to do something in his/her turn.

When player2 complete his/her turn, the game's turn go back to player1 again and the game will repeat this sequence until all off the characters in player1's team or player2's team are dead.



The winner of this game is the player who can eliminate another player’s team first. The winner will be promoted to be the ProgMeth lover.

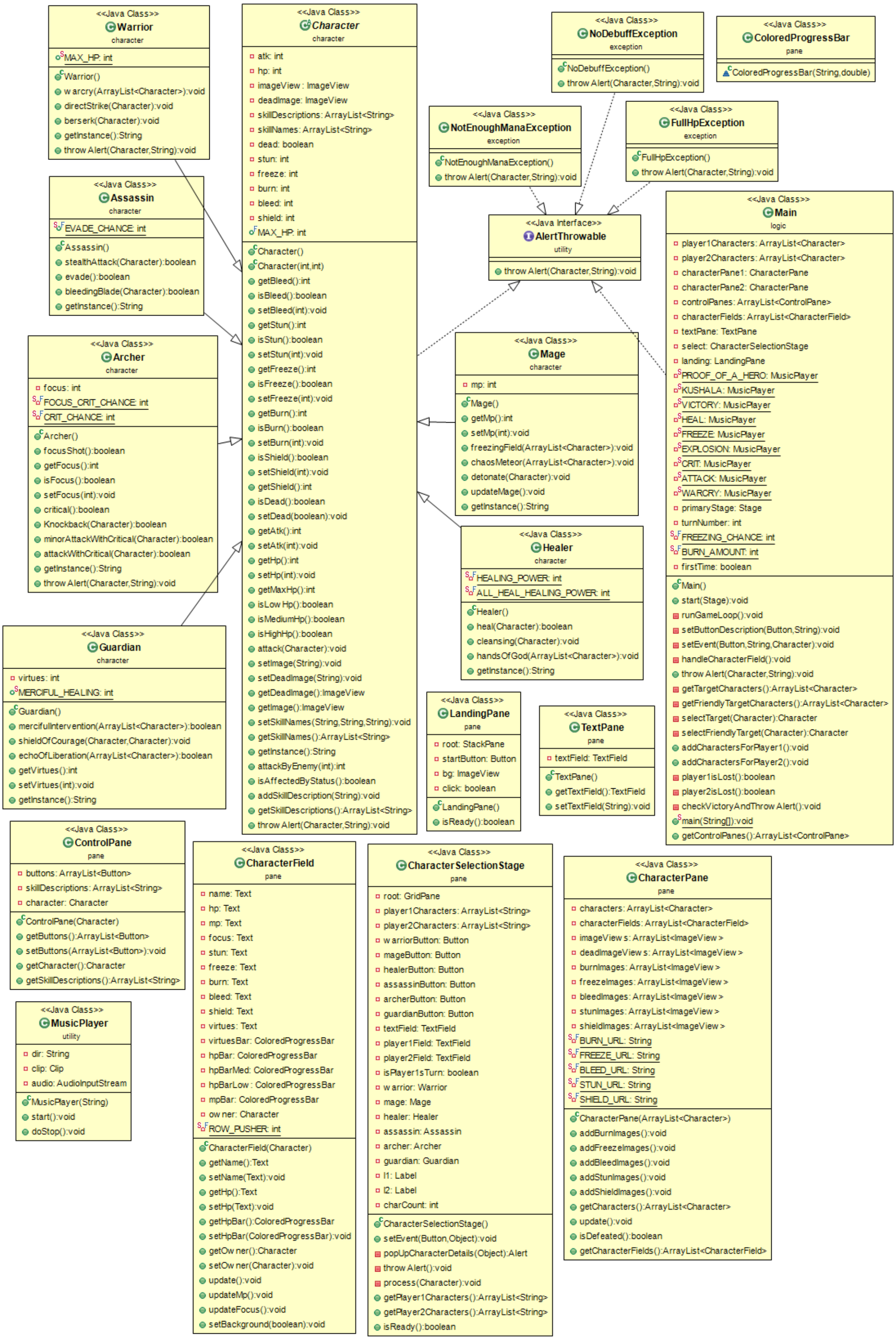


Alert!

We have a winner!

CONGRATULATIONS!! Player 1 is the one who really loves PROGMETH!!
Want to play again?

Implementation Detail



1. Package character

1.1 Class Archer

1.1.1 Field

-int focus	Store focus point
-int FOCUS_CRIT_CHANCE	Store critical chance equals to 75 when archer is focusing
-int CRIT_CHANCE	Store critical chance equals to 30 when archer is not focusing

1.1.2 Constructor

+ Archer()	Set the corresponding fields to the basic values

1.1.3 Method

+ boolean focusShot()	If archer doesn’t focus,this method will set focus equals to 2
+ int getFocus()	Return focus value
+ boolean isFocus()	Return true if focus value is more than 0 ,return false otherwise
+ void setFocus(int focus)	Set focus value
+ boolean critical()	Return critical chance randomly
+ boolean Knockback(Character enemy)	Return random chance to get 1 or 2 focus point and return random chance to stun enemy
+ boolean minorAttackWithCritical (Character enemy)	If archer attack without critical,the damage will be reduced to a half of attack point
+ boolean attackWithCritical (Character enemy)	If archer attack with critical,the damage will be multiply by 2
+ String getInstance()	Return “Archer”
+ void throwAlert(Character character,String reason)	Alert a message when the archer has some special action

1.2 Class Assasin

1.2.1 Field

-int EVADE_CHANCE	Store evade chance equals to 30

1.2.2 Constructor

+ Assasin()	Set the corresponding fields to the basic values

1.2.3 Method

+ boolean stealthAttack(Character enemy)	When enemy’s health points is lower than 35%,assassin will attack the enemy with double damage
+ boolean evade()	Return chance to evade the attacker
+ boolean bleedingBlade(Character enemy)	Make target bleed with damage
+ String getInstance()	Return “Assasin”

1.3 Class Character

This abstract class represents the basic model of all characters in this game

1.3.1 Field

-int atk	Store attack point of the character
-int hp	Store health point of the character
-imageView ImageView	Store the picture of the character
-imageView deadImage	Store the picture of the character when the character is dead
-ArrayList<String> skillDescriptions	Store all description of skills that the character has
-ArrayList<String> skillNames	Store all names of skills that the character has
-boolean dead	Store the value of the character to check that the character is dead or alive
-int stun	Store the stun values of the character
-int freeze	Store the freeze values of the character
-int burn	Store the burn values of the character
-int bleed	Store the bleed values of the character
-int shield	Store the shield values of the character
-int MAX_HP	Store max health point of the character

1.3.2 Constructor

+ Character()	Set the corresponding fields to the basic values
+ Character(int atk,int hp)	Set the corresponding fields to the parameters

1.3.3 Method

+ int getBleed()	Return bleed values
+ boolean isBleed()	Return true if the character’s bleed points is more than zero ,return false otherwise

+ void setBleed(int bleed)	Set bleed values to the parameter value
+ int getStun()	Return stun values
+ boolean isStun()	Return true if the character’s stun points is more than zero ,return false otherwise
+ void setStun(int stun)	Set stun values to the parameter value
+ int getFreeze()	Return freeze values
+ boolean isFreeze()	Return true if the character’s freeze points is more than zero ,return false otherwise
+ void setFreeze(int freeze)	Set freeze values to the parameter value
+ int getBurn()	Return burn values
+ boolean isBurn()	Return true if the character’s burn points is more than zero ,return false otherwise
+ void setBurn(int burn)	Set burn values to the parameter value
+ int getShield()	Return shield values
+ boolean isShield()	Return true if the character’s shield points is more than zero ,return false otherwise
+ void setShield(int shield)	Set shield values to the parameter value
+ boolean isDead(boolean dead)	Return true if the character is dead ,return false otherwise
+ void setDead(boolean dead)	Set dead values to the parameter value
+ int getAtk()	Return attack values
+ void setAtk(int atk)	Set attack values to the parameter value
+ int getHp()	Return health points of the character
+ void setHp(int hp)	Set health points to the parameter value
+ int getMaxHp()	Return max health points of the character
+ boolean isLowHp()	Return true if the character’s health points is lower than 35%,return false otherwise
+ boolean isMediumHp()	Return true if the character’s health points is between 35% and 59.9%,return false otherwise
+ boolean isHighHp()	Return true if the character’s health points is more than 59.9%,return false otherwise
+ void attack(Character enemy)	Attack the enemy
+ void setImage(String url)	Set the image of the character
+ void setDeadImage(String url)	Set the dead image of the character
+ ImageView getDeadImage()	Return the dead image of the character
+ ImageView getImage()	Return the image of the character
+ void setSkillNames(String s1,String s2,String s3)	Set all skill’s names
+ ArrayList<String> getSkillNames	Return arraylist of all skill’s names
+ String getInstance()	Return “Character”

+ int attackByEnemy(int damage)	Decrease the character’s health points with the parameter
+ boolean isAffectedByStatus()	Return true if the character get some bad status,return false otherwise
+ void addSkillDescription(String s)	Add description of the skill
+ ArrayList<String> getSkillDescriptions()	Return all skill’s descriptions
+ void throwAlert(Character character,String reason)	Alert a message when the character has some special action

1.4 Class Guardian

1.4.1 Field

-int virtues	Store virtues points of the character
-int MERCFUL_HEALING	Store merciful healing points of the character

1.4.2 Contructor

+ Guardian()	Set the corresponding fields to the basic values

1.4.3 Method

+ boolean mercifulIntervention(ArrayList<Character> friends)	Heal all of the guardian’s team and collect virtues points
+ void shieldOfCourage(Character enemy,Character friend)	Attack the enemy and add a shield point to the guardian’s friend
+ boolean echoOfLiberation(ArrayList<Character enemies>)	Attack all enemies with a half of attack points plus a half of virtues points
+ int getVirtues()	Return virtues points
+ void setVirtues(int virtues)	Set virtues points of the character to the parameter value
+ String getInstance()	Return “Guardian”

1.5 Class Healer

1.5.1 Field

-int HEALING_POWER	Store healing values
-int ALL_HEALING_POWER	Store healing all members in the team values

1.5.2 Constructor

+ Healer()	Set the corresponding fields to the basic values

1.5.3 Method

+ boolean heal(Character friend)	Recovery your friend’s health points
+ void cleansing(Character friend)	Delete all bad status of healer’s friend
+ void handsOfGod(ArrayList<Character> friends)	Recovery every member’s health points
+ String getInstance()	Return “Healer”

1.6 Class Mage

1.6.1 Field

- int mp	Store mana points

1.6.2 Constructor

+ Mage()	Set the corresponding fields to the basic values

16.3 Method

+ int getMp()	Return mana points
+ void setMp(int mp)	Set mana points to the parameter value
+ void freezingField(ArrayList<Character> characters)	Attack all enemies and set freeze status of all enemies to 2
+ void chaosMeteor(ArrayList<Character> characters)	Attack all enemies and set burn status of all enemies to 2
+ void detonate(Character character)	Attack the enemy by (100-usedMp) multiply by 5 Throw NotEnoughManaException if MP is not enough to use
+ void updateMage()	Add 15 mana points
+ String getInstance()	Return “Mage”

1.7 Class Warrior

1.7.1 Field

-int MAX_HP	Store max health points of the warrior

1.7.2 Constructor

+ Warrior()	Set the corresponding fields to the basic values

1.7.3 Method

+ void warcry(ArrayList<Character> enemies)	Attack all enemies with 120 damages
+ void directStrike(Character enemy)	Attack the enemy and stun the enemy
+ void berserk(Character enemy)	Attack the enemy with max health points minus recent health points
+ String getInstance()	Return “Warrior”
+ void throwAlert(Character character,String reason)	Alert a message when the warrior has some special action

2. Package logic

2.1 Class Main

2.1.1 Field

- ArrayList<Character> player1characters	Store player 1’s selected Characters
- ArrayList<Character> player2characters	Store player 2’s selected Characters
- CharacterPane characterPane1	Display player 1’s selected Characters
- CharacterPane characterPane2	Display player 2’s selected Characters
- ArrayList<ControlPane> controlPanes	Store all Characters’ ControlPanes
- ArrayList<CharacterField> characterFields	Store all Characters’ CharacterFields
- TextPane textPane	Store TextPane used to display “Your skills…”
- CharacterSelectionStage select	Store CharacterSelectionStage and return an ArrayList of selected Characters to be used in the gameplay
- LandingPane landing	Store a LandingPane which is use as a title screen
- MusicPlayer PROOF_OF_A_HERO	Store a MusicPlayer that contains a music “Proof of a Hero”
- MusicPlayer KUSHALA	Store a MusicPlayer that contains a music “Kushala”
- MusicPlayer VICTORY	Store a MusicPlayer that contains a music “Victory”
- MusicPlayer HEAL	Store a healing sound effect
- MusicPlayer FREEZE	Store a freezing sound effect
- MusicPlayer EXPLOSION	Store a explosion sound effect
- MusicPlayer CRIT	Store a critical sound effect
- MusicPlayer ATTACK	Store an attack sound effect
- MusicPlayer WARCRY	Store a warcry sound effect

- Stage primaryStage	Store a primaryStage from the start() method
- int turnNumber	Store current turn order
- int FREEZING_CHANCE	Store the percentage number of freezing chance
- int BURNING_AMOUNT	Store the burn tick damage
- boolean firstTime	Store a boolean the check if the player is playing for the first time or not

2.1.2 Method

+ void addCharactersForPlayer1()	Receive the returned ArrayList<String> from “select” variable and add the respective Characters into the player1Characters
+ void addCharactersForPlayer2()	Receive the returned ArrayList<String> from “select” variable and add the respective Characters into the player2Characters
- void checkVictoryAndThrowAlert()	Check if there’s a winner or not, if yes then throw alert to ask the users whether they want to continue playing or not
+ getControlPanes()	Return controlPanes
- ArrayList<Character> getFriendlyTargetCharacters()	Return the current turn owner’s allies
- ArrayList<Character> getTargetCharacters()	Return the current turn owner’s enemies
- void handleCharacterField()	Update all the CharacterField’s information
- boolean player1isLost()	Return true if player 1’s characters are all dead, else return false
- boolean player2isLost()	Return true if player 2’s characters are all dead, else return false
- void runGameLoop()	Update the turn and check the status effects on the current Character and allies its effect to them
- Character selectFriendlyTarget(Character character)	Pop up an alert to let the player choose their ally target, then return the selected ally
- Character selectTarget(Character character)	Pop up an alert to let the player choose their enemy target, then return the selected enemy
- void setButtonDescription(Button b, String description)	Set the button’s tooltip to show their skill’s description
- void setEvent(Button b, String skillName, Character character)	Set the event handler when the button is pressed to activate the respective skill
+ void start(Stage primaryStage)	Function used to start the application

+ void throwAlert(Character character, String reason)	Alert a message when the characters have some special action

3. Package pane

3.1 Class CharacterField

3.1.1 Field

-Text name	Store name of the characterfield in Text form
-Text hp	Store health points of the characterfield in Text form
-Text mp	Store mana points of the characterfield in Text form
-Text focus	Store focus points of the characterfield in Text form
-Text stun	Store stun points of the characterfield in Text form
-Text freeze	Store freeze points of the characterfield in Text form
-Text burn	Store burn points of the characterfield in Text form
-Text bleed	Store bleed points of the characterfield in Text form
-Text shield	Store shield points of the characterfield in Text form
-Text virtues	Store virtues points of the characterfield in Text form
-ColoredProgressBar virtuesBar	Store virtues points of the characterfield in ColoredProgressBar form
-ColoredProgressBar hpBar	Store health points of the characterfield in ColoredProgressBar form
-ColoredProgressBar hpBarMed	Store health points of the characterfield in ColoredProgressBar form
-ColoredProgressBar HpBarLow	Store health points of the characterfield in ColoredProgressBar form
-ColoredProgressBar mpBar	Store mana points of the characterfield in ColoredProgressBar form
-Character owner	Store owner of the characterfield
-int ROW_PUSHER	Store row of the game

3.1.2 Constructor

+ CharacterField(Character character)	Set the corresponding fields to the parameter value

3.1.3 Method

+ Text getName()	Return name of the characterfield
+ void setName(Text name)	Set name values to the parameter value
+ Text getHp()	Return health points of the characterfield
+ void setHp(Text hp)	Set Health points to the parameter value
+ ColoredProgressBar getHpBar()	Return ColoredProgressBar of health points
+ void setHpBar(ColoredProgressBar hpBar)	Set ColoredProgressBar of health points to the parameter value
+ Character getOwner()	Return the owner of the characterfield
+ void setOwner(Character owner)	Set the owner value to the parameter value
+ void update()	Update the characterfield
+ void updateMp()	Update mp of the characterfield
+ void updateFocus()	Update focus points of the characterfield
+ void setBackground(boolean b)	Set background of the characterfield

3.2 Class CharacterPane

3.2.1 Field

-ArrayList<Character> characters	Store all characters
-ArrayList<Character> characterFields	Store all characterfields
-ArrayList<ImageView> imageViews	Store all character’s images
-ArrayList<ImageView> deadImageViews	Store all character’s dead images
-ArrayList<ImageView> burnImages	Store burning images
-ArrayList<ImageView> freezeImages	Store freezing images
-ArrayList<ImageView> bleedImages	Store bleeding images
-ArrayList<ImageView> stunImages	Store stuning images
-ArrayList<ImageView> shieldImages	Store shield images
-String BURN_URL	Store location of a burning image
-String FREEZE_URL	Store location of a freezing image
-String BLEED_URL	Store location of a bleeding image
-String STUN_URL	Store location of a stunning image
-String SHIELD_URL	Store location of a shield image

3.2.2 Constructor

+ CharacterPane(ArrayList<Character> characters)	Set the corresponding fields to the parameter value

3.2.3 Method

+ void addBurnImages()	Add burning images
+ void addFreezeImages()	Add freezing images
+ void addBleedImages()	Add bleeding images
+ void addStunImages()	Add stunning images
+ void addShieldImages()	Add shield images
+ ArrayList<Character> getCharacters()	Return all characters
+ void update()	Update the characterpane
+ boolean isDefeated()	Return true if this character is defeated,return false otherwise
+ ArrayList<CharacterField> getCharacterFields()	Return all characterfields

3.3 Class CharacterSelectionStage

3.3.1 Field

-GridPane root	Store gridpane
-ArrayList<String> player1Characters	Store player1's characters
-ArrayList<String> player2Characters	Store player2's characters
-Button warriorButton	Store a button with warrior's picture
-Button mageButton	Store a button with mage's picture
-Button healerButton	Store a button with healer's picture
-Button assasinButton	Store a button with assasin's picture
-Button archerButton	Store a button with archer's picture
-Button guardianButton	Store a button with guardian's picture
-TextField textField	Store textfield
-TextField player1Field	Store textfield of player1
-TextField player2Field	Store textfield of player2
-boolean isPlayer1sTurn	Store turn of a player
-Warrior warrior	Store warrior class
-Mage mage	Store mage class
-Healer healer	Store healer class
-Assasin assassin	Store assassin class
-Archer archer	Store archer class
-Guardian guardian	Store guardian class
-Label l1	Store label values
-Label l2	Store label values
-int charCount	Store the number of characters

3.3.2 Constructor

+ CharacterSelectionState()	Set the corresponding fields to the basic value

3.3.3 Method

+ void setEvent(Button b,Object character)	Set event to the button
-Alert popUpCharacterDetails(Object o)	Alert popup
-void throwAlert()	Alert error popup
-void process(character.Character character)	Capture the result and add to arraylist
+ ArrayList<String> getPlayer1Characters()	Return player1's characters
+ ArrayList<String> getPlayer2Characters()	Return player2's characters
+ boolean isReady()	Return true if the number of characters equal to 6 , return false otherwise

3.4 Class ColoredProgressBar

3.4.1 Constructor

+ ColoredProgressBar(String styleClass,double progress)	Set the corresponding fields to the parameter value

3.5 Class ControlPane

3.5.1 Field

-ArrayList<Button> buttons	Store all buttons
-ArrayList<String> skillDescriptions	Store all skill's descriptions
-Character character	Store a character

3.5.2 Constructor

+ ControlPane(Character character)	Set the corresponding fields to the parameter value

3.5.3 Method

+ ArrayList<Button> getButtons()	Return all buttons
+ void setButtons(ArrayList<Button> buttons)	Set all buttons to parameter values
+ Character getCharacter()	Return the character
+ ArrayList<String> getSkillDescriptions()	Return all skill's descriptions

3.6 Class LandingPane

3.6.1 Field

-StackPane root	Store stackpane
-Button startButton	Store starting button
-ImageView bg	Store background image
-boolean click	Store click value

3.6.2 Constructor

+ LandingPane()	Set the corresponding fields to the basic value

3.6.3 Method

+ isReady()	Return true if a player click startbutton ,return false otherwise

3.7 Class TextPane

3.7.1 Field

-TextField textField	Store textfield values

3.7.2 Constructor

+ TextPane()	Set the corresponding fields to the basic value

3.7.3 Method

+ TextField getTextField()	Return the textfield of this class
+ void setTextField(String text)	Set textfield value to the parameter value

4. Package utility

4.1 Interface AlertThrowable

This Interface represents the ability of the class to be able to throw out the alert

4.1.1 Method

+ void throwAlert(Character character, String reason)	Alert a message when the characters have some special action

4.2 Class MusicPlayer

4.2.1 Field

- String dir	Store .wav’s file directory
- Clip clip	Store Clip variable
- AudioInputStream audio	Store AudioInputStream variable

4.2.2 Constructor

+ MusicPlayer(String dir)	Construct a MusicPlayer object with the instantiated directory

4.2.3 Method

+ void start()	Start playing a music that is loaded in the class
+ void doStop()	Stop playing a currently playing music

5. Package exception

5.1 Class NotEnoughManaException

Throw this exception when a mage trying to use a skill but she doesn’t have enough mana

5.2 Class NoDebuffException

Throw this exception when a healer trying to cleanse a debuff but the target doesn’t have one

5.3 Class FullHpException

Throw this exception when a healer trying to use “Hands of God” skill while all of her teammates are still at full HP