

Final Project

Documentation

Created by

Kittiphop Vichitkijja 6432015321

Naphan Choatchuangnapha 6432083021

2110215 Programming Methodology

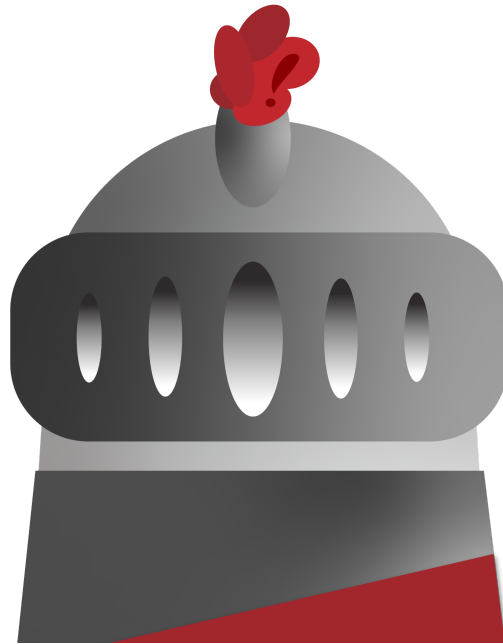
Semester 2 Year 2021

Faculty of Engineering

Chulalongkorn University

RPG Chess

Introduction



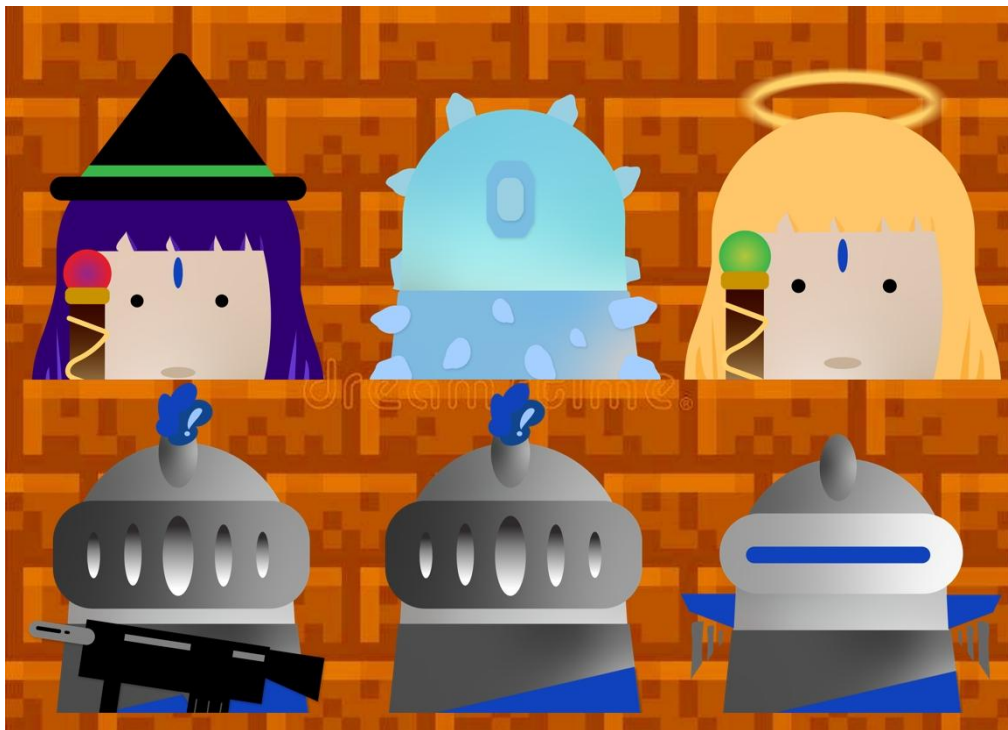
RPG Chess is a modified version of original chess. Every piece (or unit) will have its own health point (HP) and unique action. The objective of this game is to defeat all of the opponent's units.

Rules

1. Each player starts with 4 normal units and 5 special units.
2. In each turn, a player can either move a unit, attack an enemy, or use a unit's ability.

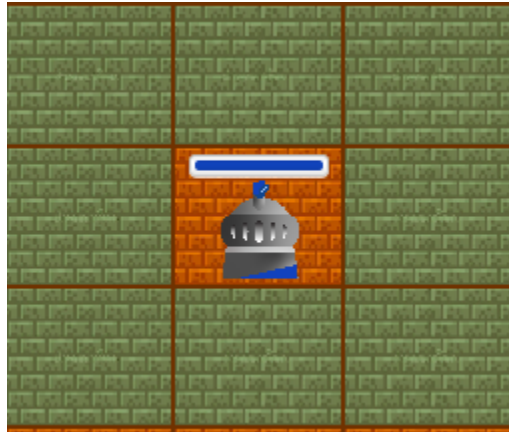
3. A player must do an action in 90 seconds or the turn will be skipped
4. When your unit kills an enemy, that unit will get upgraded.
Enhancing its attack power or ability permanently.
5. If a player doesn't have any units left on board, they lose.

Units

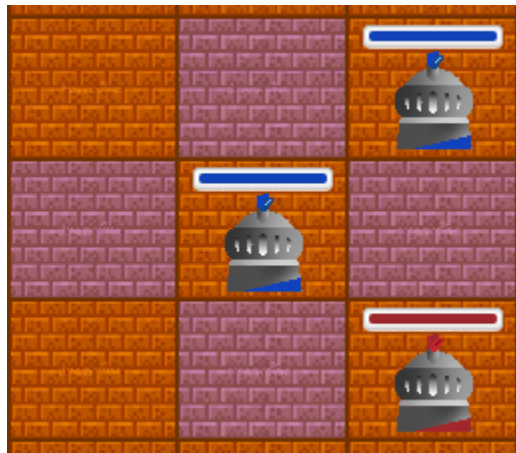


There are 6 types of units. Each type can either attack an enemy, inflict a debuff on an enemy, or buff an allied unit.

Controlling a unit

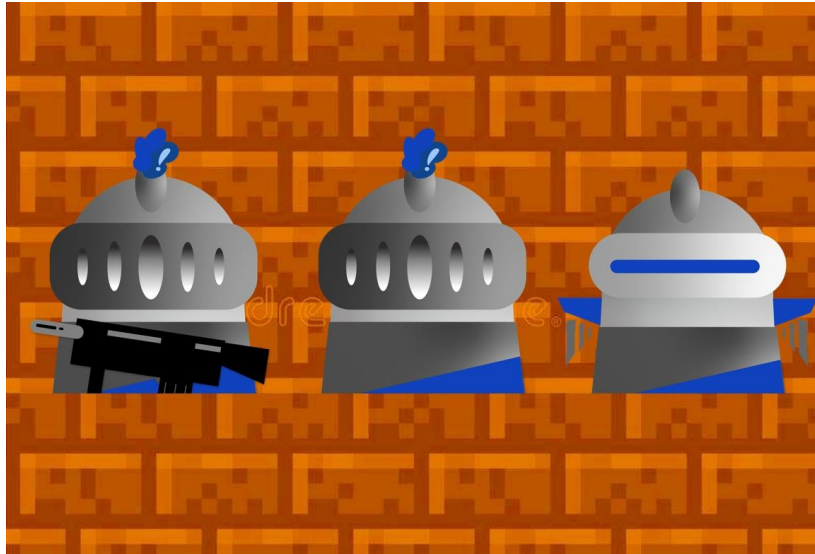


Press LEFT CLICK to show all available squares that the unit can move to. CLICK again to move to the selected square.



Press RIGHT CLICK to show all available squares that the unit can attack, inflict a debuff, or buff an ally. CLICK again to perform an action to the selected square.

Attackable Units



There are 3 types of units that can attack.

The left unit is a Shooter Unit that has a long-range attack.

The middle unit is a Normal Unit.

The right unit is a Flying Unit.

A Flying Unit cannot be attacked by a Normal Unit.

Debuffable Units



There are 2 types of units that can inflict a debuff.

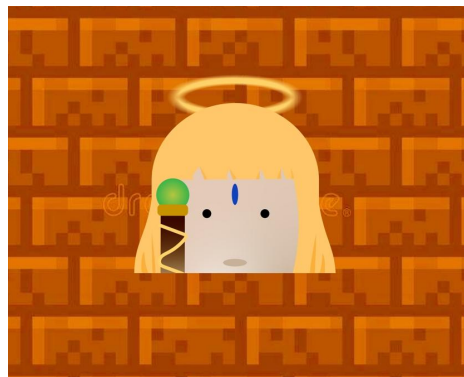
The left unit is a Venom Unit which can inflict poison.

The poisoned unit loses HP for a certain time.

The right unit is a Freezer Unit that can inflict freeze.

A Frozen Unit cannot do anything for a certain time.

Buffable Unit



There is 1 type of unit that can buff an ally.

That unit is a Healer Unit which can heal an ally for a certain HP.

Potions



While playing, there will be potions generated at the middle row of the board. Players can consume a potion by moving a unit to that square.

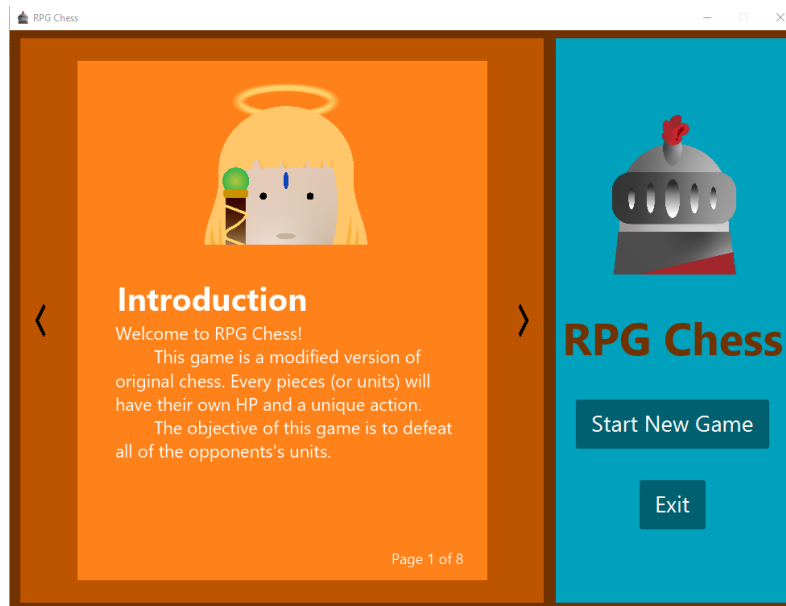
There are 3 types of potions. Each has its own expired turn.

The left potion is a Healing Potion.

The middle potion is an Upgrade Potion, which enhances a unit permanently.

The right potion is a Toxic Potion.

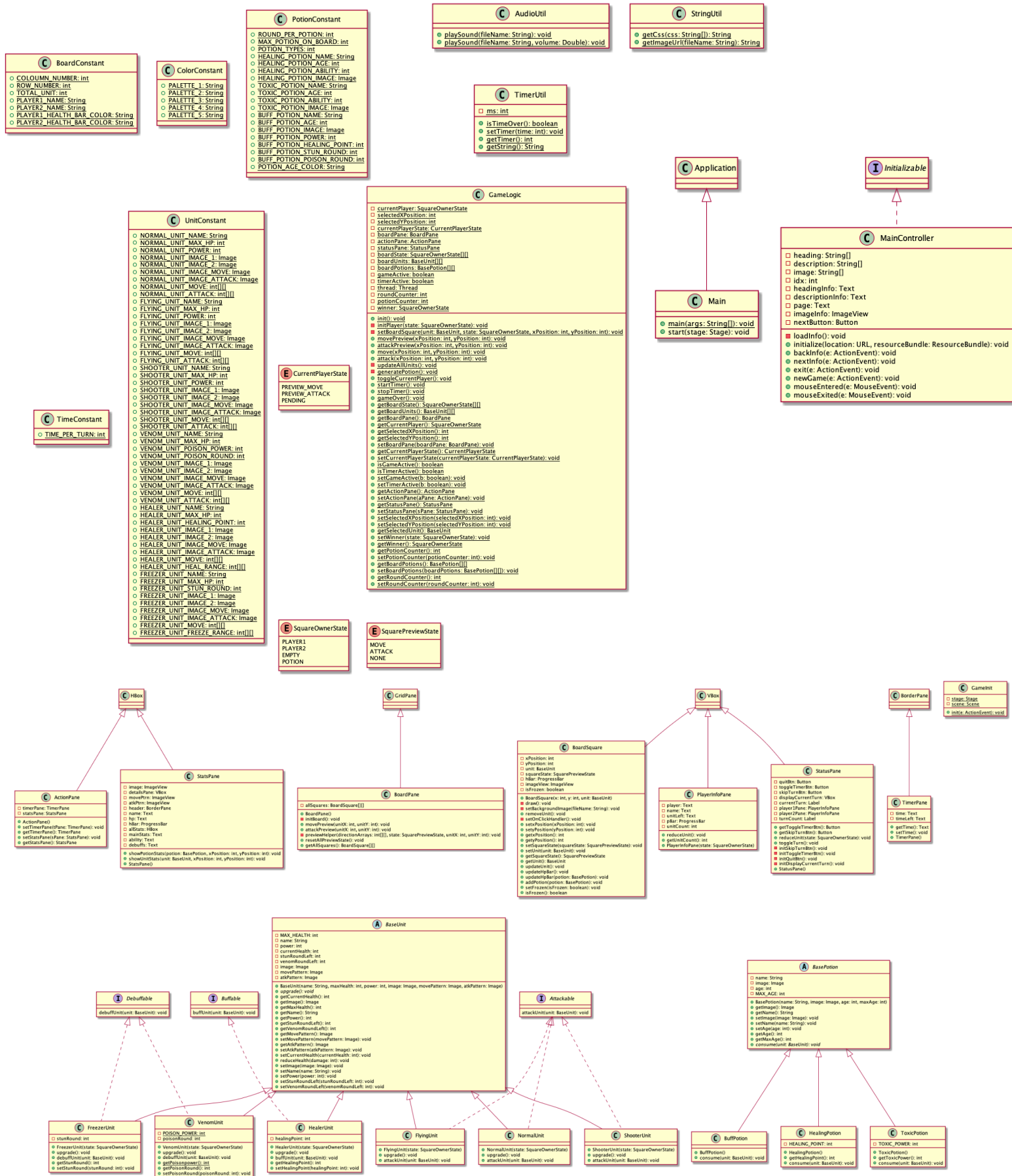
Main Menu Scene



Game Screen Scene



Class Diagram



1. Package application

1.1 class Main extends Application

1.1.1 Methods

+ static void main(String[] args);	Launch application
+ void start(Stage stage) throws Exception;	<ul style="list-style-type: none">- Load MainMenu.fxml in try catch exception- Set title, icon, and scene to stage- Set resizable to false- Show stage

1.2 class MainController implements Initializable

1.2.1 Fields

- String[] heading;	Heading text arrays
- String[] description;	Description text arrays
- String[] image;	Image path arrays
- int idx;	Current index of arrays
- Text headingInfo;	Heading info text
- Text descriptionInfo;	Description info text
- Text page;	Current page text
- ImageView imageInfo;	Image info
- Button nextButton;	Next button

1.2.2 Methods

- void loadInfo()	Set page text, heading info, image view, and description to current index info
+ void initialize(URL location, ResourceBundle resourceBundle);	Call loadInfo method
+ void backInfo(ActionEvent e);	<ul style="list-style-type: none">- If current index is 0 set current index to arrays length- Reduce current index by 1- Call loadInfo method- Play click sound
+ void nextInfo(ActionEvent e);	<ul style="list-style-type: none">- Add current index by 1 and mod by arrays length- Call loadInfo method- Play click sound
+ void exit(ActionEvent e);	Call Platform.exit()
+ void newGame(ActionEvent e) throws IOException;	Initializes game with action event
+ void mouseEntered(MouseEvent e);	Set opacity of button to 0.5
+ void mouseExited(MouseEvent e);	Set opacity of button to 1

2. Package base

2.1 interface Attackable

2.1.1 Methods

+ void attackUnit(BaseUnit unit);	Attack other BaseUnit
-----------------------------------	-----------------------

2.2 abstract class BasePotion

2.2.1 Fields

- String name;	Name of the potion
- Image image;	Image of the potion
- int age;	Current age of the potion
- final int MAX_AGE;	Max age of the potion

2.2.2 Constructor

+ BasePotion(String name, Image image, int age, int maxAge);	Initializes name, image, MAX_AGE, and age
--	---

2.2.3 Methods

+ abstract void consume(BaseUnit unit);	Abstract method for potion consuming
+ getter/setter for each field.	getter/setter for each field except setMaxAge

2.3 abstract class BaseUnit

2.3.1 Fields

- final int MAX_HEALTH;	Max health of the unit
- String name;	Name of the unit
- int power;	Power of the unit
- int currentHealth;	Current health of the unit
- int stunRoundLeft;	Stun round left of the unit
- int venomRoundLeft;	Venom round left of the unit
- Image image;	Image of the unit
- Image movePattern;	Image of unit's move pattern
- Image atkPattern;	Image of unit's attack pattern

2.3.2 Constructor

+ BaseUnit(String name, int maxHealth, int power, Image image, Image movePattern, Image atkPattern);	<ul style="list-style-type: none">- Initializes name, MAX_HEALTH, power, image, movePattern, atkPattern,- Set stunRoundLeft = 0- Set venomRoundLeft = 0
--	---

2.3.3 Methods

+ abstract void upgrade();	Abstract method for unit upgrade
+ void reduceHealth(int damage);	Reduce currentHealth with damage
+ getter/setter for each field.	<ul style="list-style-type: none">- getter/setter for each field except setMaxHealth- if currentHealth < 0 set it to 0- if stunRoundLeft < 0 set it to 0- if venomRoundLeft < 0 set it to 0

2.4 interface Buffable

2.4.1 Methods

+ void buffUnit(BaseUnit unit);	Buff other BaseUnit
---------------------------------	---------------------

2.5 interface Debuffable

2.5.1 Methods

+ void debuffUnit(BaseUnit unit);	Debuff other BaseUnit
-----------------------------------	-----------------------

3. Package constant

3.1 class BoardConstant

3.1.1 Fields

+ static final int COLOUMN_NUMBER;	Column number constant
+ static final int ROW_NUMBER;	Row number constant
+ static final int TOTAL_UNIT;	Total unit number constant
+ static final String	1st Player name constant

PLAYER1_NAME;	
+ static final String PLAYER2_NAME;	2nd Player name constant
+ static final String PLAYER1_HEALTH_BAR_COLOR;	1st Player health bar color constant
+ static final String PLAYER2_HEALTH_BAR_COLOR;	2nd Player health bar color constant

3.2 class ColorConstant

3.2.1 Fields

+ static final String PALETTE_1;	Hex color code constant
+ static final String PALETTE_2;	Hex color code constant
+ static final String PALETTE_3;	Hex color code constant
+ static final String PALETTE_4;	Hex color code constant
+ static final String PALETTE_5;	Hex color code constant

3.3 class PotionConstant

3.3.1 Fields

+ static final int ROUND_PER_POTION;	Round per potion constant
+ static final int MAX_POTION_ON_BOARD;	Max potion on board constant
+ static final int POTION_TYPES;	The number of potion types
+ static final String HEALING_POTION_NAME;	Healing potion name
+ static final int HEALING_POTION_AGE;	Healing potion age
+ static final int HEALING_POTION_ABILITY;	Healing potion ability
+ static final Image HEALING_POTION_IMAGE;	Healing potion image
+ static final String TOXIC_POTION_NAME;	Toxic potion name
+ static final int TOXIC_POTION_AGE;	Toxic potion age
+ static final int TOXIC_POTION_ABILITY;	Toxic potion ability
+ static final Image TOXIC_POTION_IMAGE;	Toxic potion image
+ static final String	Buff potion name

BUFF_POTION_NAME;	
+ static final int BUFF_POTION_AGE;	Buff potion age
+ static final Image BUFF_POTION_IMAGE;	Buff potion image
+ static final int BUFF_POTION_POWER;	Buff attack unit power
+ static final int BUFF_POTION_HEALING_POINT;	Buff healing unit power
+ static final int BUFF_POTION_STUN_ROUND ;	Buff freezer unit power
+ static final int BUFF_POTION_POISON_ROUND;	Buff venom unit power
+ static final String POTION_AGE_COLOR;	Hex color code constant

3.4 class TimeConstant

3.4.1 Fields

+ static final int TIME_PER_TURN;	Time per turn constant in millisecond
--------------------------------------	--

3.5 class UnitConstant

3.5.1 Fields

+ static final String name for each unit.	Name of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final int max health for each unit.	Max health of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final Image unit image of 1st player.	1st player unit image of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final Image unit image of 2nd player.	2nd player unit image of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final Image move preview image.	Move preview image of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final Image attack preview image.	Attack preview image of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final int[][] move arrays	Move arrays of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer

	Unit
+ static final int[][] attack arrays	Attack arrays of Normal Unit, Flying Unit, Shooter Unit, Venom Unit, Healer Unit, and Freezer Unit
+ static final int NORMAL_UNIT_POWER;	Normal Unit power
+ static final int FLYING_UNIT_POWER;	Flying Unit power
+ static final int SHOOTER_UNIT_POWER;	Shooter Unit power
+ static final int VENOM_UNIT_POISON_POWER;	Venom Unit poison power
+ static final int VENOM_UNIT_POISON_ROUND;	Venom Unit poison round
+ static final int HEALER_UNIT_HEALING_POINT;	Healer Unit healing point
+ static final FREEZER_UNIT_STUN_ROUND;	Freezer Unit stun round

4. Package gui

4.1 class ActionPane extends HBox

4.1.1 Fields

- TimerPane timerPane;	TimerPane instant
- StatsPane statsPane;	StatsPane instant

4.1.2 Constructor

+ ActionPane();	<ul style="list-style-type: none">- Set PrefHeight to 125- Set alignment to center- Initializes TimerPane and StatsPane- Add TimerPane and StatsPane to HBox
-----------------	---

4.1.3 Methods

+ getter/setter for each field.	getter/setter for each field
---------------------------------	------------------------------

4.2 class BoardPane extends GridPane

4.2.1 Fields

- BoardSquare[][] allSquares;	Arrays of BoardSquare
-------------------------------	-----------------------

4.2.2 Constructor

+ BoardPane();	<ul style="list-style-type: none">- Call init method of GameLogic- Set padding to 5- Set styles- Set alignment to center- Set column and row constraints- Call initBoard method
----------------	--

4.2.3 Methods

- void initBoard();	Initializes BoardSquare and add to GridPane
+ void movePreview(int unitX, int unitY);	<ul style="list-style-type: none">- Get unit at unitX and unitY position and get corresponded move arrays- Call previewHelper with move arrays and move preview state

+ void attackPreview(int unitX, int unitY);	- Get unit at unitX and unitY position and get corresponded attack arrays - Call previewHelper with attack arrays and attack preview state
- void previewHelper(int[][] directionArrays, SquarePreviewState state, int unitX, int unitY);	- Get new x and y position and check if it's in the board call setSquareState method with input state
+ void resetAllPreviewState();	Call setSquareState method with none state for all squares
+ getter/setter for each field.	getter/setter for each field

4.3 class BoardSquare extends VBox

4.3.1 Fields

- int xPosition;	X position of this square
- int yPosition;	Y position of this square
- BaseUnit unit;	Unit of this square
- SquarePreviewState squareState;	Square preview state
- ProgressBar hBar;	Progress bar of this square
- ImageView imageView;	ImageView of this square
- boolean isFrozen;	Frozen state of this square

4.3.2 Constructor

+ BoardSquare(int x, int y, BaseUnit unit);	<ul style="list-style-type: none">- Initializes progress bar and image view- Set image view fit height and fit width to 50- Set square state to none- Set x and y position- Set padding to 3- Set unit- Set isFrozen to false- Call setOnClickListener method
---	--

4.3.3 Methods

- void draw();	<ul style="list-style-type: none">- Call updateHpBar method- Clear all children- Set unit image to image view- Add hBar and image view to VBox- Set VBox style correspond to isFrozen
- void setBackgroundImage(String fileName);	<ul style="list-style-type: none">- Set style and background correspond to file name

+ void removeUnit();	<ul style="list-style-type: none"> - Clear all children - Set style and background to normal brick background
- void setOnClickHandler();	<ul style="list-style-type: none"> - Add mouse cliced event to VBox - If current board unit isn't null call showUnitStats - If current board state is potion call showPotionStats - If square state is move state and selected square is empty or potion then call move of GameLogic then return - If square state is attack state and selected sqare is not empty and not potion and selected square is correspond to unit type then call attack of GameLogic then return - If the event is left click then call move preview of GameLogic - If the event is right click then call attack preview of GameLogic
+ void updateUnit();	Call set unit and pass current unit from GameLogic to the funciton
+ void updateHpBar();	If the unit isn't null set progress bar to current health and set styles of progress bar

+ void updateHpBar(BasePotion potion);	Set progress bar to current potion age and set styles of progress bar
+ void addPotion(BasePotion potion);	<ul style="list-style-type: none"> - Clear all children - Set image view to potion image - Call updateHpBar with potion - Add hBar and image view to VBox
+ getter/setter for each field.	<ul style="list-style-type: none"> - getter/setter for each field - For setSqaureState call setBackgroundImage after set new value correspond to new square state - If set unit to null set frozen to false and call removeUnit method; otherwise, check unit's stun round and set frozen value then call draw method

4.4 class GameInit

4.4.1 Fields

- static Stage stage;	Game stage
- static Scene scene;	Game scene

4.4.2 Methods

+ static void init(ActionEvent e);	<ul style="list-style-type: none">- Create border pane- Set PrefSize to 980 * 720- Set status pane of GameLogic- Set status pane to the right of border pane- Set action pane of GameLogic- Set action pane to the top of border pane- Set board pane of GameLogic- Set board pane to the center of border pane- Set stage's scene to border pane- Show stage- Play game start sound- Set timer active and game active of GameLogic to true
------------------------------------	--

4.5 class PlayerInfoPane extends VBox

4.5.1 Fields

- Text player;	Player text
----------------	-------------

- Text name;	Name text
- Text unitLeft;	Unit left text
- ProgressBar pBar;	Progress bar
- int unitCount;	Unit count of the player

4.5.2 Constructor

+PlayerInfoPane(SquareOwnerState state);	<ul style="list-style-type: none"> - Set player and name correspond to state - Set unit count and unit left - Set styles of text and progress bar - Add player, name, unit left, progress bar to VBox
--	---

4.5.3 Methods

+ getter/setter for unit count.	<ul style="list-style-type: none"> - Getter for unit count - Use reduce unit 1 as a setter and update text and progress bar
---------------------------------	---

4.6 class StatsPane extends HBox

4.6.1 Fields

- ImageView image;	Image view
- VBox detailsPane;	Details pane
- ImageView movePtrn;	Move pattern image view
- ImageView atkPtrn;	Attack pattern image view
- BorderPane header;	Header pane
- Text name;	Name text
- Text hp;	Hp text
- ProgressBar hBar;	Progress bar
- HBox allStats;	All stats box
- Text mainStats;	Main stats text
- Text ability;	Ability text
- Text debuffs;	Debuffs text

4.6.2 Constructor

+ StatsPane();	- Initializes and set styles for image, move pattern, attack pattern, details pane, header, name, hp, progress bar, all stats, main stats, debuffs text, ability text
----------------	---

	<ul style="list-style-type: none"> - Add main stats, debuffs text, and ability text to all stats - Set name to header's left - Set hp to header's right - Add header, progress bar, and all stats to details pane - Add image, details pane, move pattern, and attack pattern to HBox
--	--

4.6.3 Methods

+ void showPotionStats(BasePotion potion, int xPosition, int yPosition);	<ul style="list-style-type: none"> - Set image to potion image - Set move pattern and attack pattern to null - Set name, hp, progress bar, main stats, debuffs text, and ability
+ void showUnitStats(BaseUnit unit, int xPosition, int yPosition);	<ul style="list-style-type: none"> - Set image to unit image - Set move pattern, attack patternname, hp, progress bar, main stats, debuffs text, and ability

4.7 class StatusPane extends VBox

4.7.1 Fields

- Button quitBtn;	Quit button
- Button toggleTimerBtn;	Toggle time button
- Button skipTurnBtn;	Skip turn button
- VBox displayCurrentTurn;	Current Turn VBox
- Label currentTurn;	Current turn label
- PlayerInfoPane player1Pane;	1st Player pane
- PlayerInfoPane player2Pane;	2nd Player pane
- Label turnCount;	Turn count label

4.7.2 Constructor

+ StatusPane();	<ul style="list-style-type: none">- Set styles to VBox- Call initQuitBtn, initToggleTimerBtn, initSkipTurnBtn, and initDisplayCurrentTurn methods- Create button pane and add toggle timer button, skip button, and quit button to button pane- Initializes all player pane- Add current turn vbox, button
-----------------	--

	pane, 1st player pane, and 2nd player pane to VBox
--	--

4.7.3 Methods

+ void reduceUnit(SquareOwnerState state);	- Call reduceUnit of player pane correspond to state - Check unit count and set winner if exists
+ void toggleTurn();	Toggle turn and update all related text
- void initSkipTurnBtn();	Initializes skip turn button and set styles
- void initToggleTimerBtn();	Initializes toggle timer button and set styles
- void initQuitBtn();	Initializes quit button and set styles
- void initDisplayCurrentTurn();	Initializes display current turn VBox and set styles
+ getter for toggle time button and skip turn button	getter/setter for button and skip turn button

4.8 class TimerPane extends BorderPane

4.8.1 Fields

- Text time;	Current time text
- Text timeLeft;	Time left text

4.8.2 Constructor

+ TimerPane();	<ul style="list-style-type: none">- Set timer of TimeUtil- Set styles of BorderPane- Initializes time and time left text- Set time left text at top- Set time at center
----------------	---

4.8.3 Methods

+ getter/setter for time	getter/setter for time
--------------------------	------------------------

5. Package logic

5.1 enum CurrentPlayerState

PREVIEW_MOVE	Preview move state
PREVIEW_ATTACK	Preview attack state
PENDING	Pending state

5.2 class GameLogic

5.2.1 Fields

- static SquareOwnerState currentPlayer;	Current player
- static int selectedXPosition;	Selected x position
- static int selectedYPosition;	Selected y position
- static CurrentPlayerState currentPlayerState;	Current player state
- static BoardPane boardPane;	Border pane
- static ActionPane actionPane;	Action pane
- static StatusPane statusPane;	Status pane
- static SquareOwnerState[][] boardState;	Arrays of board state
- static BaseUnit[][] boardUnits;	Arrays of board units

- static BasePotion[][] boardPotions;	Arrays of board potions
- static boolean gameActive;	Is game active variable
- static boolean timerActive;	Is time active variable
- static Thread thread;	Timer thread
- static int roundCounter;	Round counter
- static int potionCounter;	Potion counter
- static SquareOwnerState winner;	Game winner

5.2.2 Methods

+ static void init();	<ul style="list-style-type: none"> - Set winner to null - Set round counter to 1 - Set potion counter to 0 - Initializes board arrays - Call initPlayer method
- static void initPlayer(SquareOwnerState state);	Call setBoardSquare method with initial unit and position
- static void setBoardSquare(BaseUnit unit, SquareOwnerState state, int xPosition, int yPosition);	Set board state and board unit

+ static void movePreview(int xPosition, int yPosition);	<ul style="list-style-type: none"> - Call reset all preview state of board pane - Call move preview of board pane and set current player state to move preview - If x and y position is selected or unit get stunned or click opponent unit set current player state to pending
+ static void attackPreview(int xPosition, int yPosition);	<ul style="list-style-type: none"> - Call reset all preview state of board pane - Call attack preview of board pane and set current player state to attack preview - If x and y position is selected or unit get stunned or click opponent unit set current player state to pending
+ static void move(int xPosition, int yPosition);	<ul style="list-style-type: none"> - If x and y position has potion call consume - Move unit to x and y position and update board pane - Show new unit stats - Call toggleCurrentPlayer method - Play move sound

+ static void attack(int xPosition, int yPosition);	<ul style="list-style-type: none"> - Get selected and target unit - Check type of selected unit and call attack, buff, or debuff to target unit - Show new unit stats - Call toggleCurrentPlayer method
- static void updateAllUnits();	<ul style="list-style-type: none"> - Update all unit refer to current board arrays - Reduce potion age by 1 and set remaining potion
- static void generatePotion();	<ul style="list-style-type: none"> - Generate potion at the middle of the board if the square is empty
+ static void toggleCurrentPlayer();	<ul style="list-style-type: none"> - Set selected x and y position to -1 - Call reset all preview state of board pane - Call update all unit method - Call generate potion method every 3 rounds if the current potion doesn't exceed max potion on the board - If the winner isn't null call game over method - Toggle current player - Reset timer and update status

	pane - Add round counter by 1
+ static void startTimer();	- Create a timer thread and update GUI in thread using Platform.runLater - Start timer thread
+ static void stopTimer();	If a thread isn't null then interrupt the thread
+ static void gameOver();	Disable skip and toggle time button and show alert if the game is over.
+ static BaseUnit getSelectedUnit();	Return selected unit
+ getter/setter for each field.	getter/setter for each field

5.3 enum SquareOwnerState

PLAYER1	1st Player square
PLAYER2	2nd Player square
EMPTY	Empty square
POTION	Potion square

5.4 enum SquarePreviewState

MOVE	Move state
ATTACK	Attack state
NONE	None state

6. Package potion

6.1 class BuffPotion extends BasePotion

6.1.1 Constructor

+ BuffPotion();	Call BasePotion constructor with BuffPotion constant
-----------------	--

6.1.2 Methods

+ void consume(BaseUnit unit);	<ul style="list-style-type: none">- Upgrade unit- Play upgrade sound
--------------------------------	---

6.2 class HealingPotion extends BasePotion

6.2.1 Fields

- final int HEALING_POINT;	Healing point of the potion
----------------------------	-----------------------------

6.2.2 Constructor

+ HealingPotion();	Call BasePotion constructor and set healing point with HealingPotion constant
--------------------	---

6.2.3 Methods

+ int getHealingPoint();	Return healing point
+ void consume(BaseUnit unit);	<ul style="list-style-type: none">- Add unit health with healing point- If new health exceeds unit max health set unit health to max health- Play heal sound with 0.5 volume

6.3 class ToxicPotion extends BasePotion

6.3.1 Fields

- final int TOXIC_POWER;	Toxic power of the potion
--------------------------	---------------------------

6.3.2 Constructor

+ ToxicPotion();	Call BasePotion constructor and set toxic power with ToxicPotion constant
------------------	---

6.3.3 Methods

+ int getToxicPower();	Return toxic power
+ void consume(BaseUnit unit);	- Reduce health from unit - Play poison sound

7. Package unit

7.1 class FlyingUnit extends BaseUnit implements Attackable

7.1.1 Constructor

+ FlyingUnit(SquareOwnerState state);	- Call BaseUnit constructor with FlyingUnit constant - Use image correspond to state
---------------------------------------	---

7.1.2 Methods

+ void upgrade();	Add buff potion power to unit's power
+ void attackUnit(BaseUnit unit);	- Reduce health from unit - Play attack ranged sound with 0.3 volume

7.2 class FreezerUnit extends BaseUnit implements Debuffable

7.2.1 Fields

- int stunRound;	Freezer stun round
------------------	--------------------

7.2.2 Constructor

+ FreezerUnit(SquareOwnerState	- Call BaseUnit constructor with
--------------------------------	----------------------------------

state);	FreezerUnit constant - Use image correspond to state - Set stun round with FreezerUnit constant
---------	---

7.2.3 Methods

+ void upgrade();	Add buff potion stun round to unit's stun round
+ void debuffUnit(BaseUnit unit);	- Set stun round to freezer unit stun round + 1 (Due to turn changing) - If unit stun round greater than freezer unit stun round use the old value - Play freeze sound with 0.3 volume
+ getter/setter for stun round.	getter/setter for stun round

7.3 class HealerUnit extends BaseUnit implements Buffable

7.3.1 Fields

- int healingPoint;	Healing point
---------------------	---------------

7.3.2 Constructor

+ HealerUnit(SquareOwnerState state);	<ul style="list-style-type: none">- Call BaseUnit constructor with HealerUnit constant- Use image correspond to state- Set healing point with HealerUnit constant
---------------------------------------	---

7.3.3 Methods

+ void upgrade();	Add buff potion healing point to unit's healing point
+ void buffUnit(BaseUnit unit);	<ul style="list-style-type: none">- Add unit health with healing point- If new health exceeds unit max health set unit health to max health- Play heal sound with 0.5 volume
+ getter/setter for healing point.	getter/setter for healing point

7.4 class NormalUnit extends BaseUnit implements Attackable

7.4.1 Constructor

+ NormalUnit(SquareOwnerState state);	- Call BaseUnit constructor with NormalUnit constant - Use image correspond to state
---------------------------------------	---

7.4.2 Methods

+ void upgrade();	Add buff potion power to unit's power
+ void attackUnit(BaseUnit unit);	- If the target unit isn't Flying Unit reduce health from the target - Play attack melee sound

7.5 class ShooterUnit extends BaseUnit implements Attackable

7.5.1 Constructor

+ ShooterUnit(SquareOwnerState state);	- Call BaseUnit constructor with ShooterUnit constant - Use image correspond to state
--	--

7.5.2 Methods

+ void upgrade();	Add buff potion power to unit's power
+ void attackUnit(BaseUnit unit);	<ul style="list-style-type: none">- Reduce health from the target- Play attack ranged sound with 0.3 volume

7.6 class **VenomUnit** extends **BaseUnit** implements **Debuffable**

7.6.1 Fields

- static final int POISON_POWER;	Poison power
- int poisonRound;	Poison round

7.6.2 Constructor

+ VenomUnit(SquareOwnerState state);	<ul style="list-style-type: none">- Call BaseUnit constructor with VenomUnit constant- Use image correspond to state- Set poison round with VenomUnit constant
--------------------------------------	--

7.6.3 Methods

+ void upgrade();	Add buff potion stun round to unit's stun round
+ void debuffUnit(BaseUnit unit);	- Set venom round to poison round - Play poison sound
+ getter/setter for each field.	getter/setter for each field except setPoisonPower

8. Package util

8.1 class AudioUtil

8.1.1 Methods

+ static void playSound(String fileName);	Play selected sound
+ static void playSound(String fileName, Double volume);	Play selected sound with selected volume

8.2 class StringUtil

8.2.1 Methods

+ static String getCss(String... css);	Concat all css string and return a single string
+ static String getImageUrl(String fileName);	Add "file:resources/" prefix to file name and return

8.3 class TimerUtil

8.3.1 Fields

- static int ms;	Current millisecond
------------------	---------------------

8.3.2 Methods

+ static boolean isTimeOver();	Return true if current millisecond is less than or equal to 0; otherwise, return false
+ static String getString();	Return beautiful format of current millisecond
+ static getter/setter for each field.	Static getter/setter for each field