

Krung Rom Documentation

Created by

Narat Ratanawiphanon 6633062121

Sirikorn Jaisri 6633242821

Anima Ruttanamanoothit 6633278421

Ingkharut Ruttana-O-Pha 6633291521

2110215 Programming Methodology

Semester 2 Year 2023

Chulalongkorn University

PokéBattle!

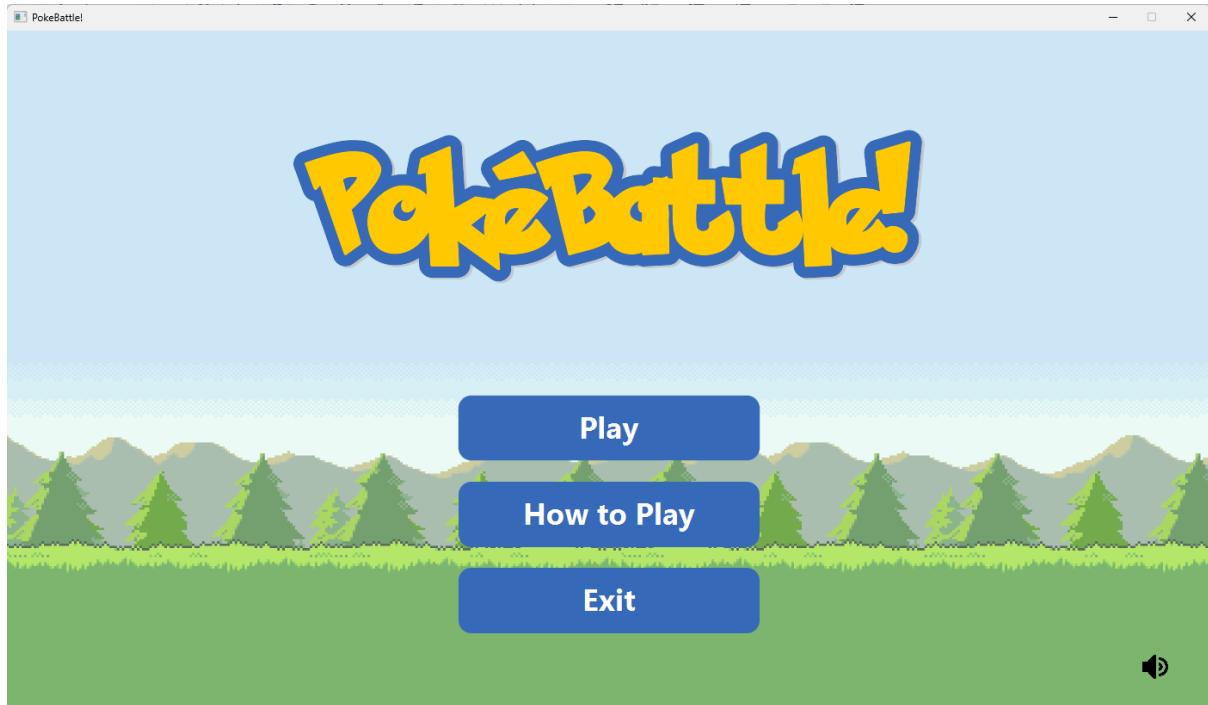
Introduction

PokéBattle! is a Pokemon battling simulator in which a player can select from a pre-selected group of Pokemon and can battle other players with it. Not much has been changed from the original game except that the maximum amount of Pokemon is 3 instead of the traditional 6.

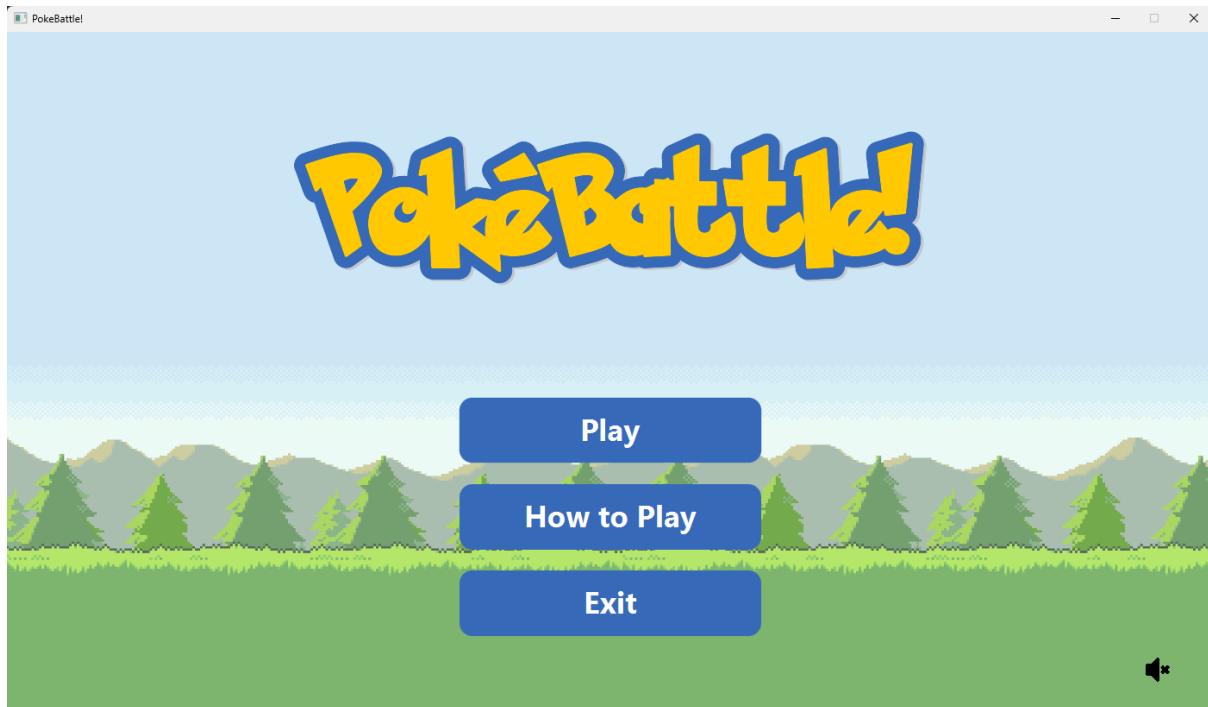
Rules

1. Each player can view the details of each Pokemon and must select 3 different Pokemon.
2. After the players have made their selection, click on "START" to begin the game.
3. First, player 1 can choose 1 of 4 actions, which are
 - a. Fight: This includes 4 movesets of the current Pokemon with the PP displayed (the amount of times this move can be used), the player can choose 1 of 4 movesets to attack the other player's Pokemon.
 - b. Item: Each player starts the game with 5 Potion, 2 Revive and 1 Full Restore, the player can choose 1 of 3 items to heal the player's Pokemon.
 - i. Potion: heals the selected Pokemon by increasing its HP by 150. However, if a Pokémon loses all of its HP, it faints and can no longer be healed with Potion.
 - ii. Revive: Revives selected Pokémon that have fainted, restoring half of their maximum HP.
 - iii. Full Restore: Revives selected Pokémon that have fainted, restoring their maximum HP, and healing from any status effects.
 - c. Switch: Switch the current Pokémon for another Pokémon that has not yet fainted.
 - d. Surrender: End the game with the victory of the opposite player.
4. After player 1's turn ends, player 2 can choose 1 of 4 actions.
5. If a player chooses to surrender or if the HP of the Pokemon on that player's side has been reduced to 0, the game ends and the other player wins the game.

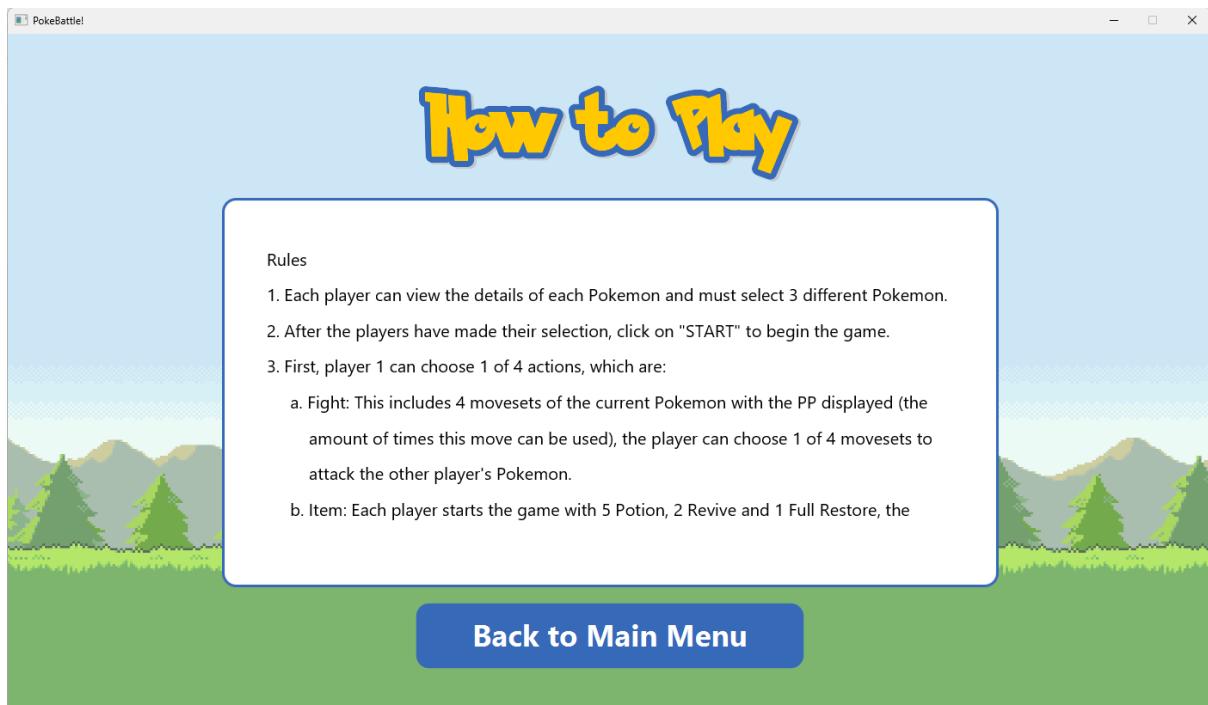
Example



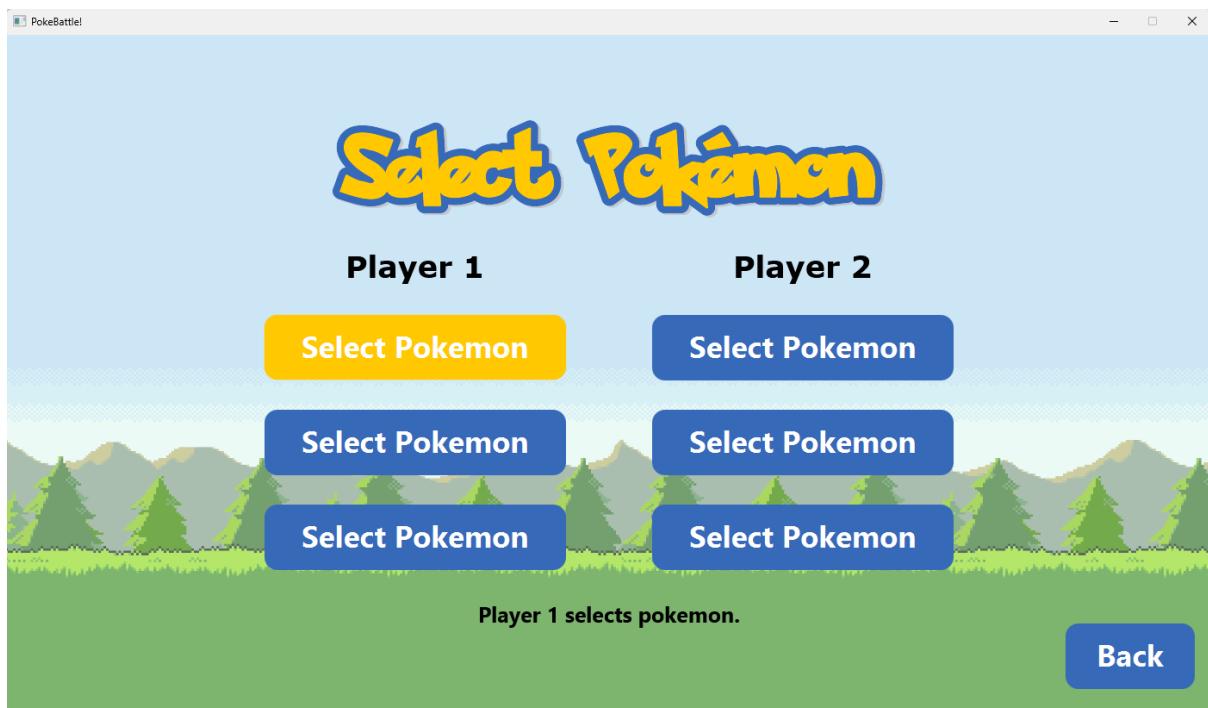
This is the main page of PokéBattle! game.



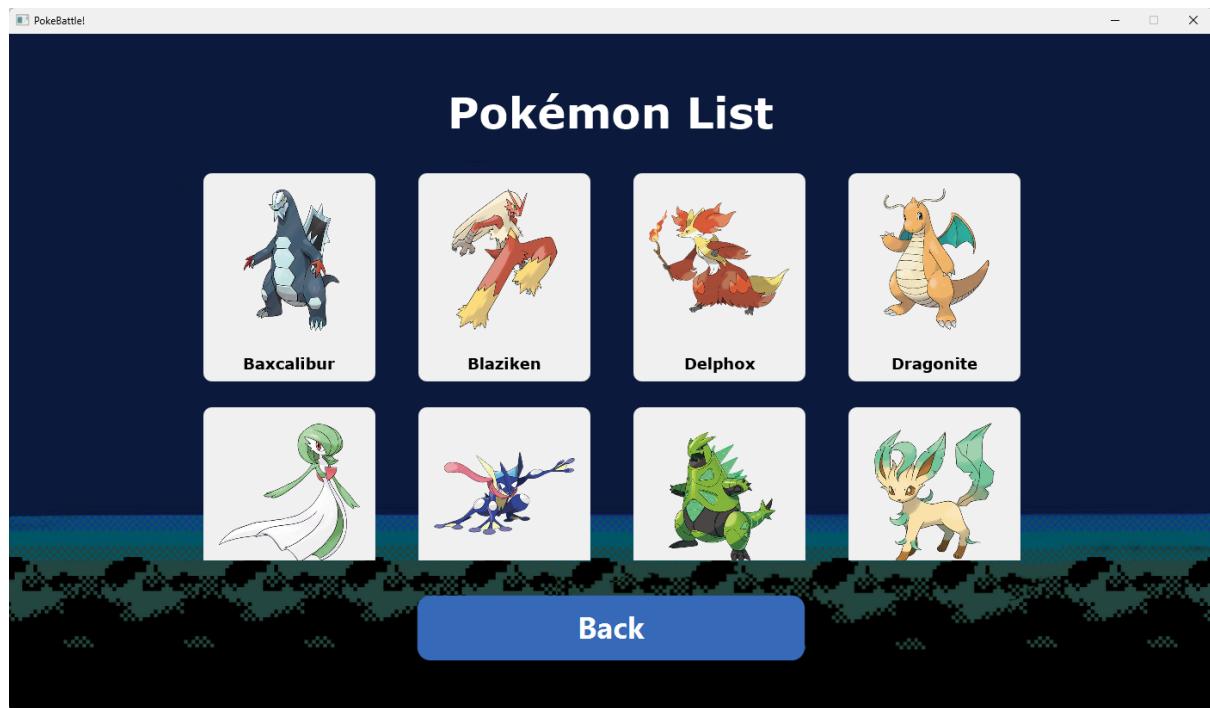
Background sound is autoplay, click on the speaker to turn it off and click again to turn on.



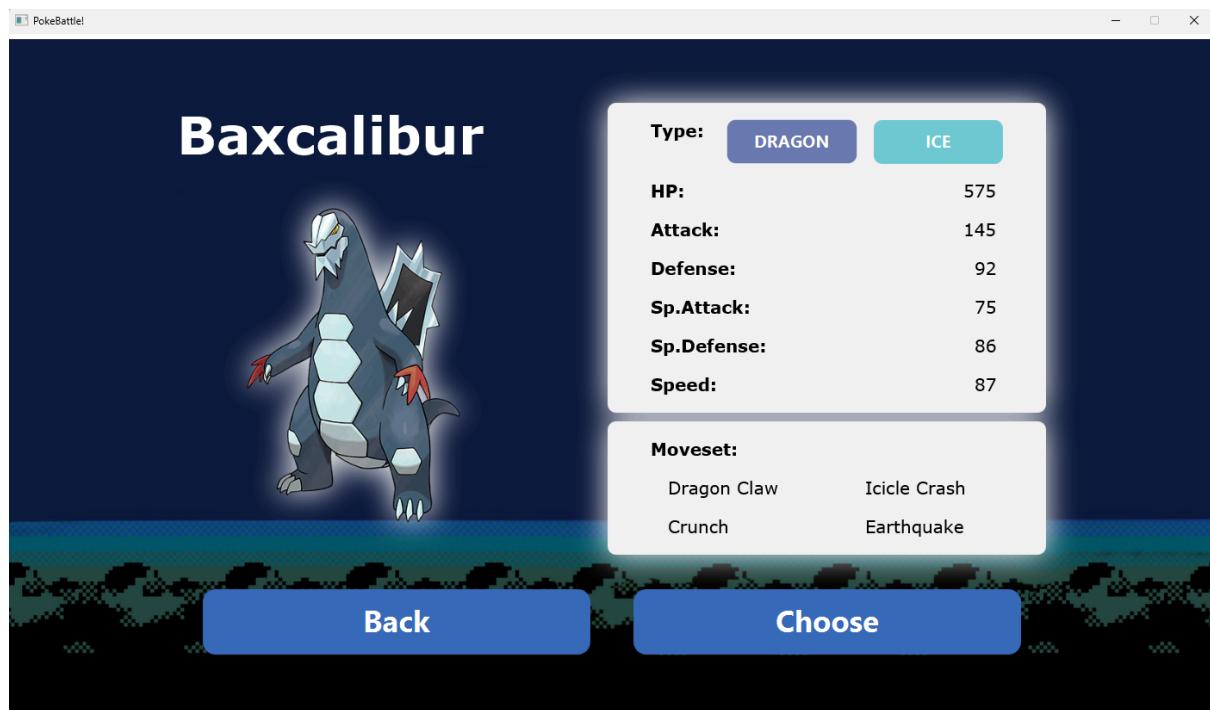
Click on the “How to Play” button on the main page to read the rules of the game.



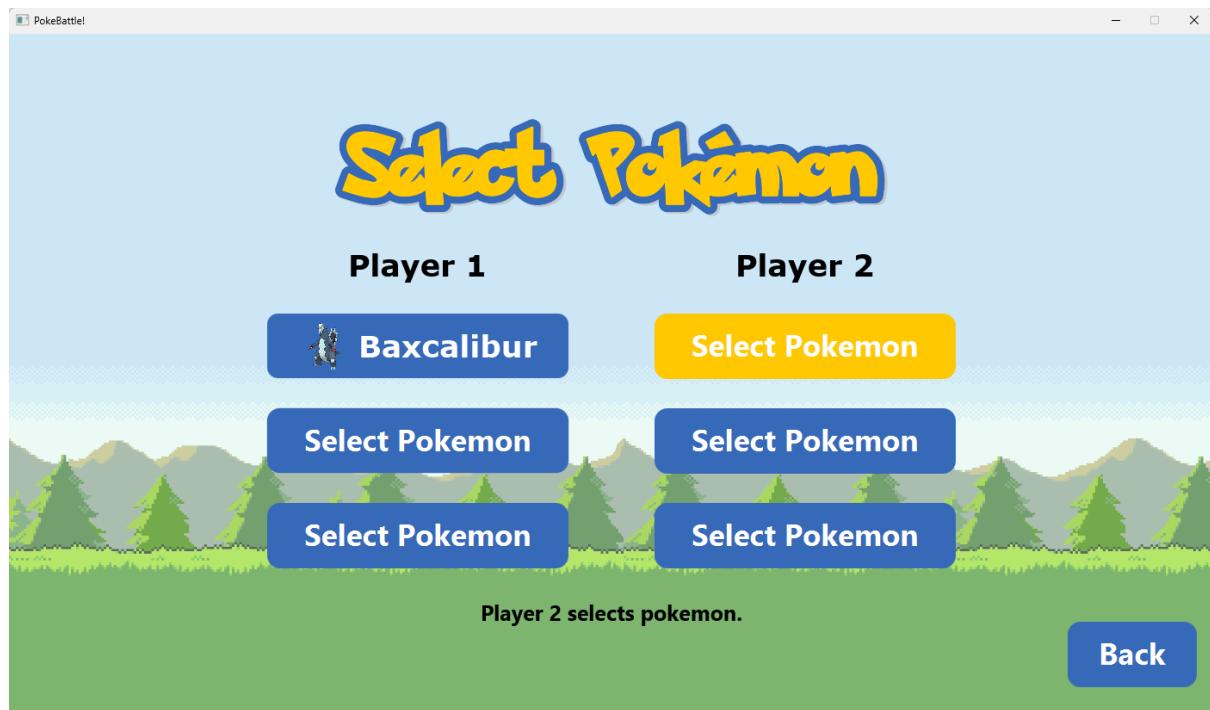
When you click on the “Play” button on the main page, the select pokemon screen will show up, you can click the highlighted “Select Pokemon” button to select your pokemon.



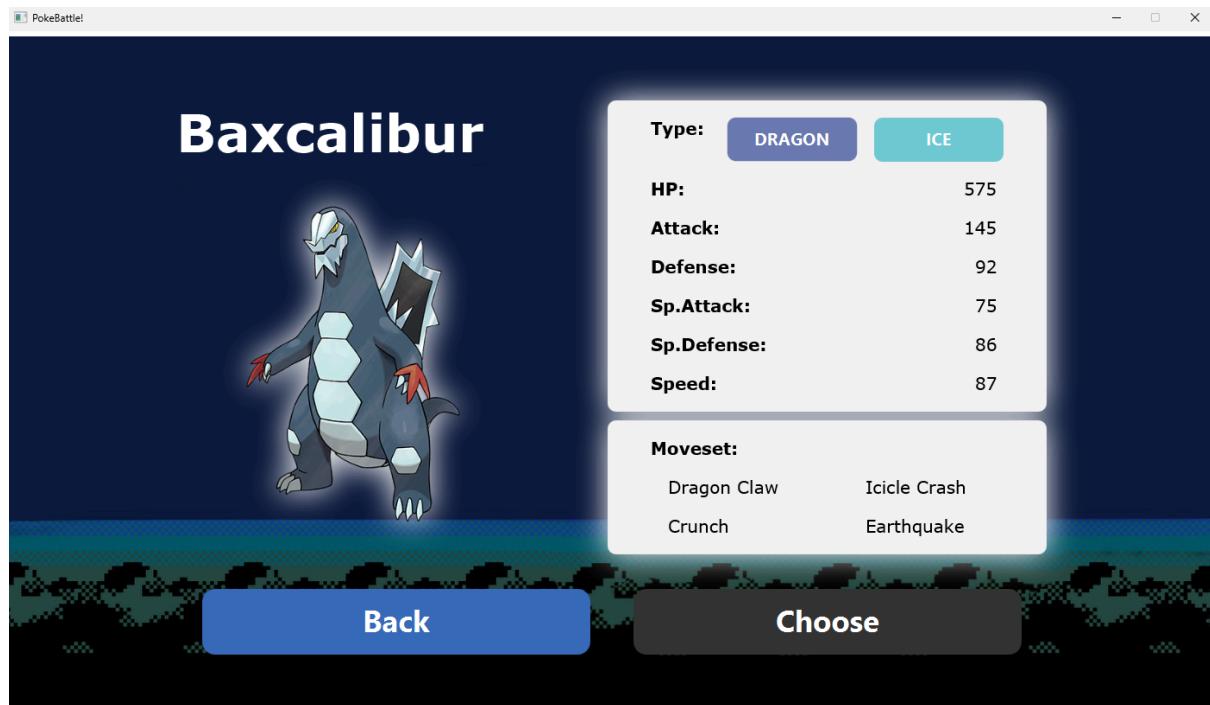
Here you can select your Pokemon from a list of pre-selected Pokemon.



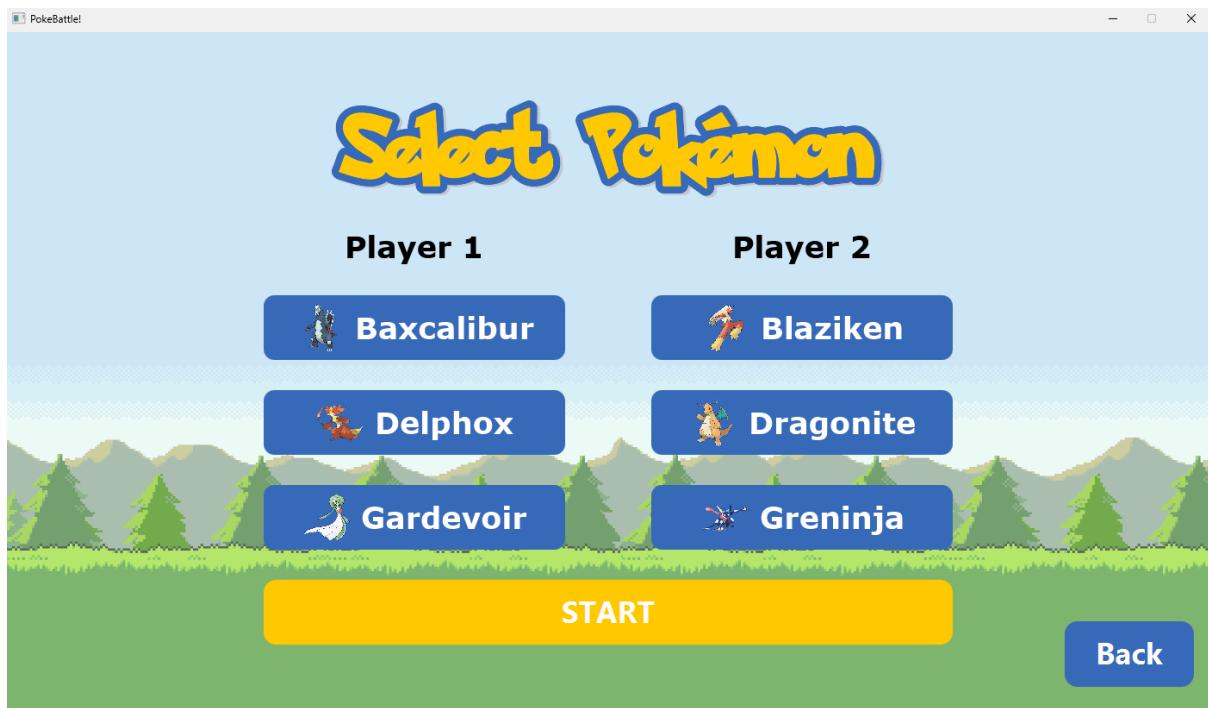
When you select a Pokemon, a different page will show up displaying the Pokemon's stats and moves to further help you in deciding, once you have decided click choose. You cannot unselect a Pokemon once it has been chosen.



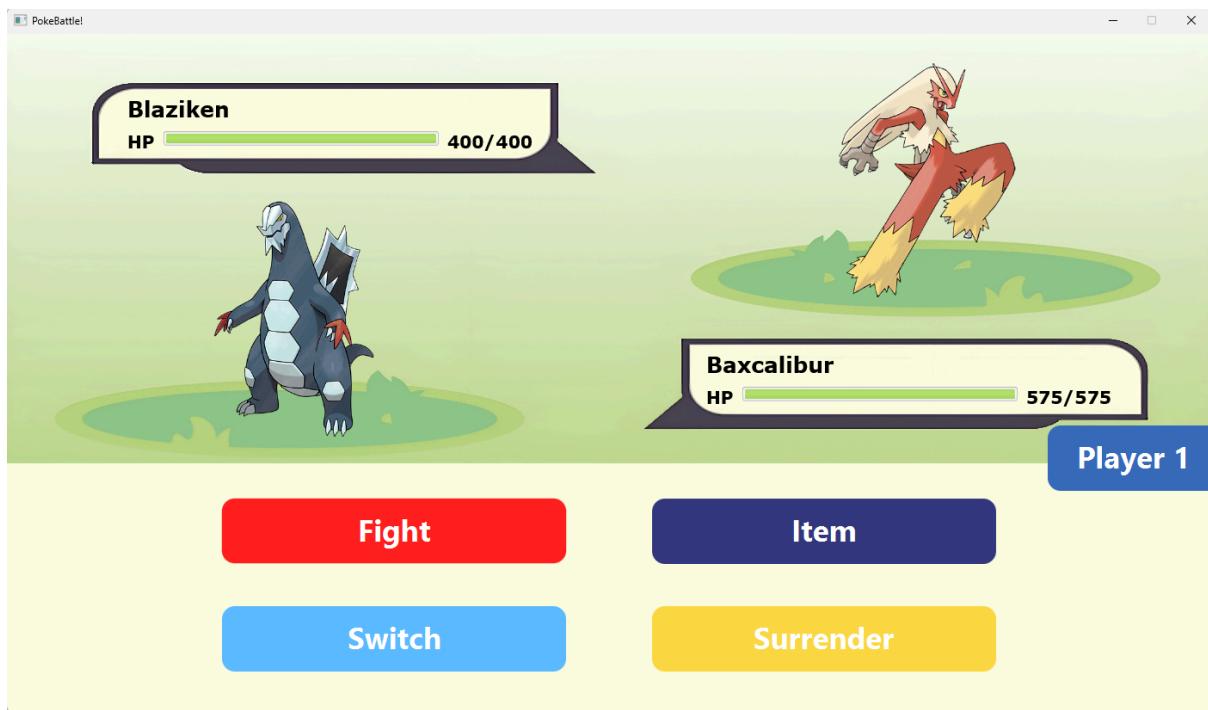
After this it's Player 2's time to select. This will alternate between Player 1 and 2 until all slots are filled.



You cannot select a Pokemon that is the same species with the ones you already chose, if you attempt to do this the choose button will black out.



Once all slots are filled, a start button will pop up. Click it and start the game!



When you start the battle, it starts from Player 1's perspective and will switch once Player 1 has selected their action. The first pokemon you and your opponent chose will be your lead pokemon. They are displayed opposed to each other, the one to the left is your own Pokemon and the one on the right is your opponent's no matter the current perspective, Here you can either Fight, use an Item, Switch to another Pokemon or surrender.



When you select “Fight”, the screen will pop up with all of your available moves. As you can see here Baxcalibur can use “Dragon Claw”, “Icicle Crash”, “Crunch” and “Earthquake”.



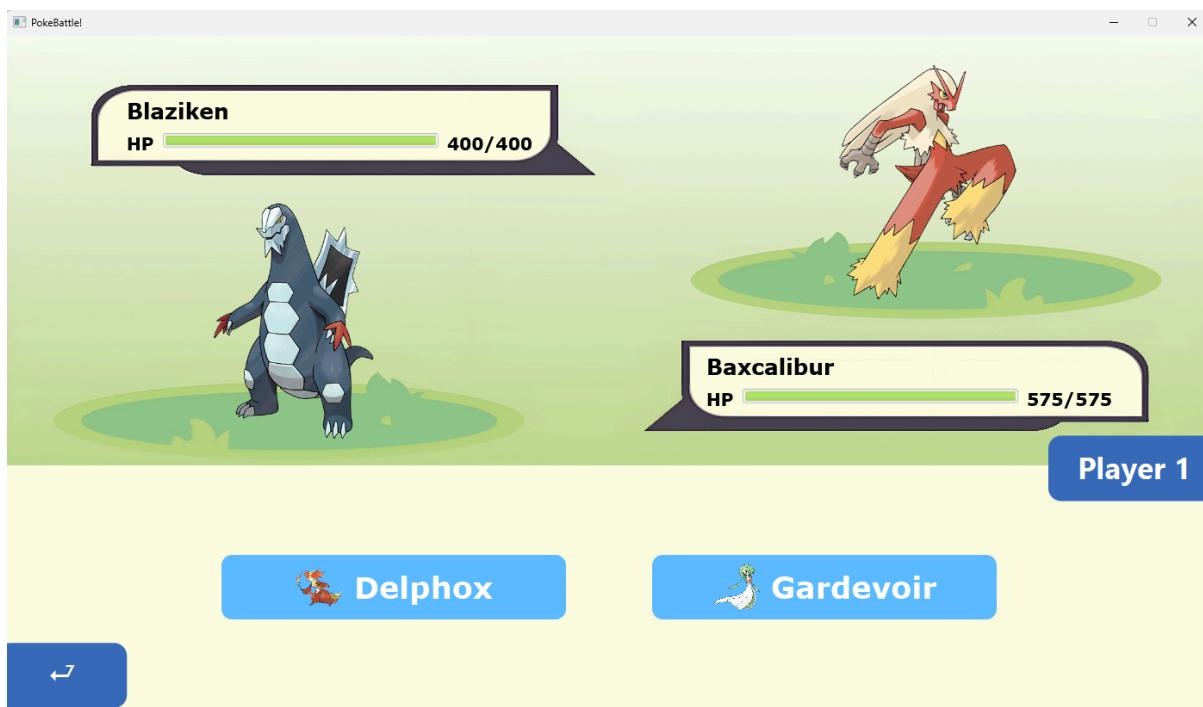
When you select “Item”, you can select from the 3 available items to use. A potion heals your pokemon, A full restore fully heals your pokemon and any status effects, A revive brings any fainted Pokemon back to half of its max HP.



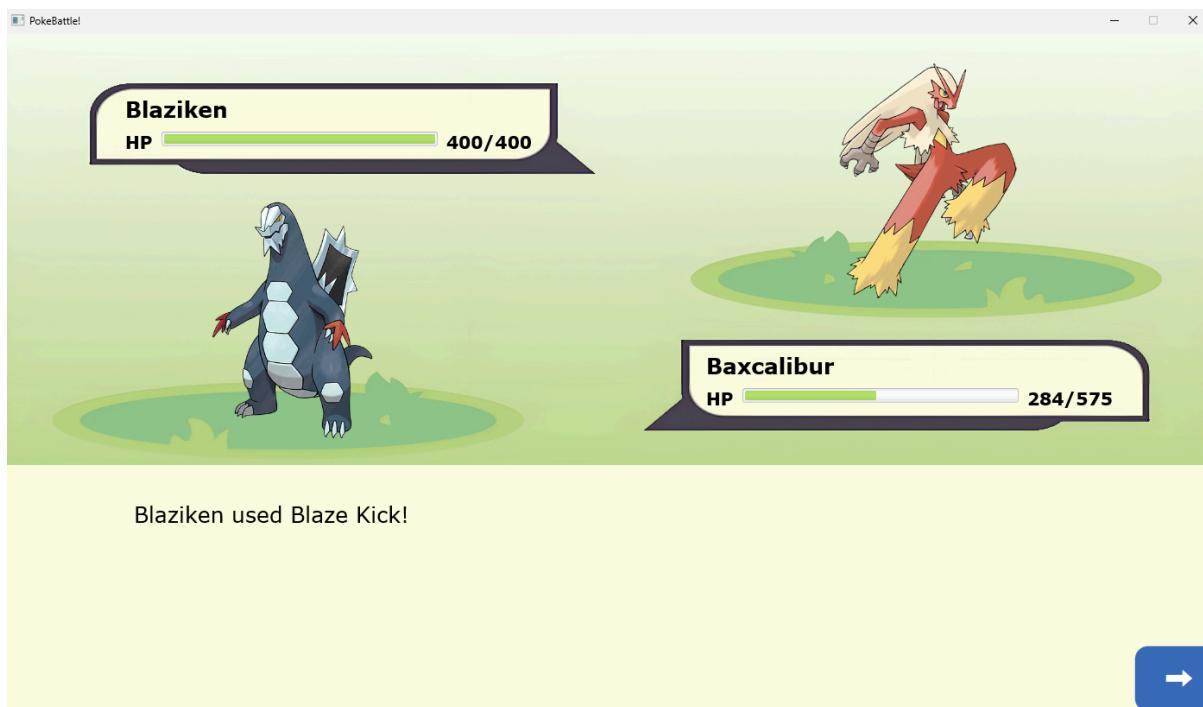
After you select which item you want to use, you can use it on any of your Pokemon.



However if your Pokemon doesn't meet the item requirements (ie: A pokemon is not fainted and you tried to use a Revive), it will black out and you cannot use the item.



When you select “Switch” you can switch from your current Pokemon to any of your other Pokemon.



Once all players have selected their actions, the battle will begin. Here the opposing Blaziken decided to use Blaze Kick, the action description will be displayed on the dialog box, and if any other actions also happen (ie: Super effective hits, status inflicted) it will also be displayed on the dialog box. You may also notice that the HP bar drops to correspond with the current amount of HP.



When Pokemon HP is equal to 0, it will display that Pokemon is fainted and remove the current Pokemon from your party.



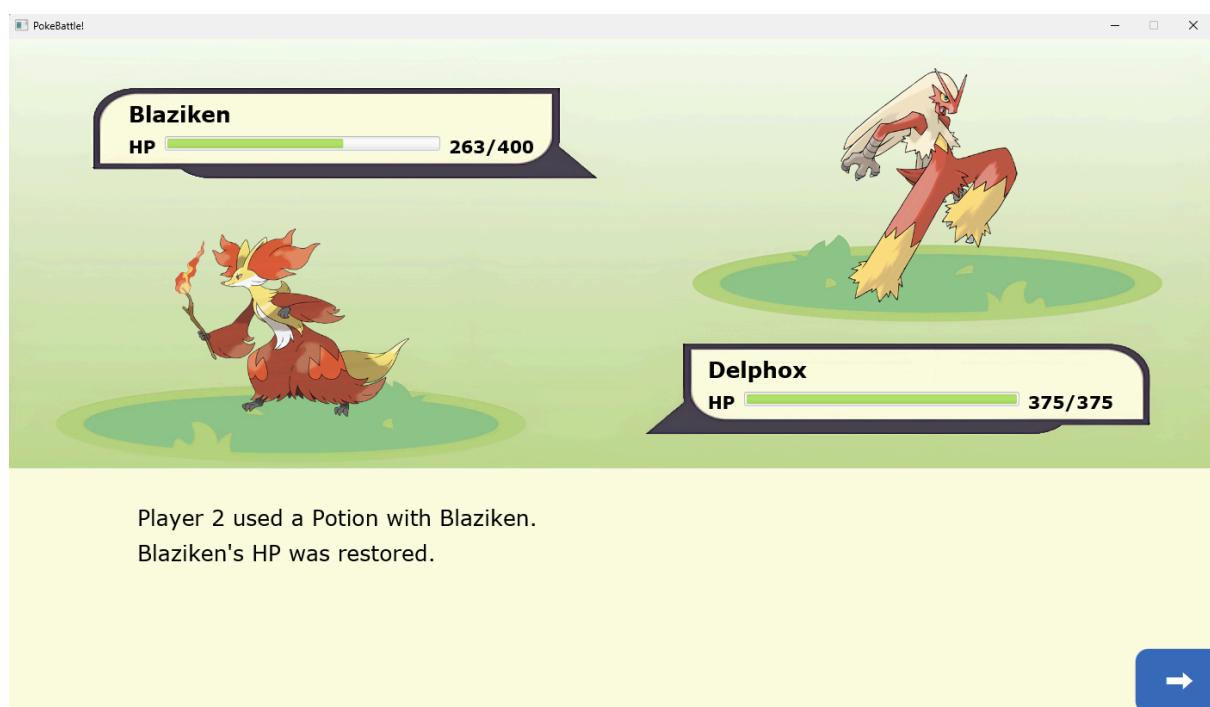
After your Pokemon faints, the game will allow you to select one of your remaining Pokemon to continue fighting.



If the Pokemon faints, you cannot switch back to that Pokemon and the button will turn black.



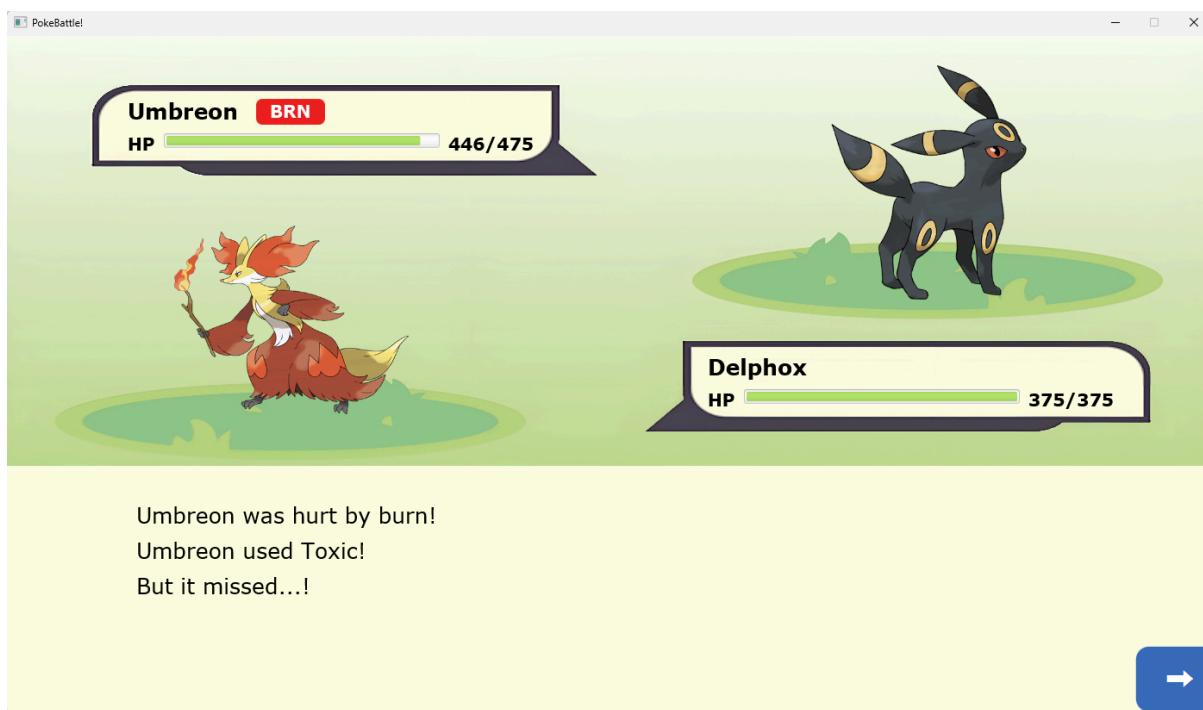
Also if you remember from earlier when we cannot use a revive on any of our Pokemon, now that one of our Pokemon has fainted, we can see that we can use it on Baxcalibur now.



Here we can see a potion being used, we can see that Blaziken recovered 150 HP.



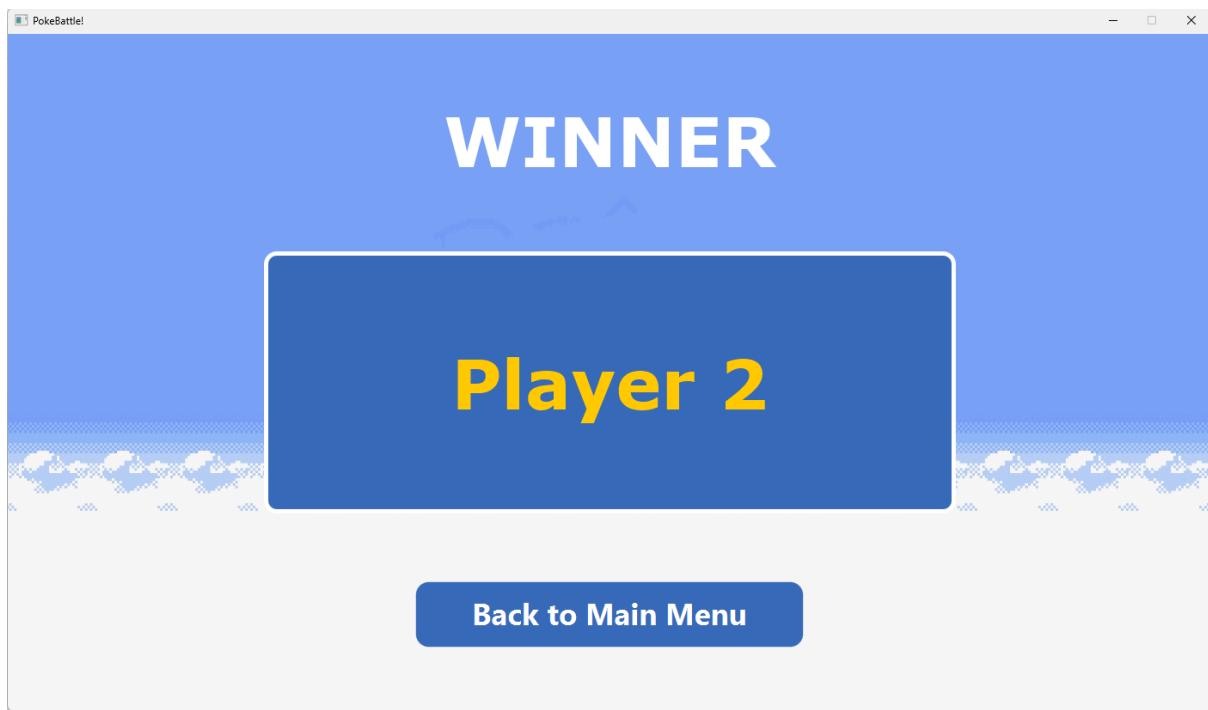
Now we will observe status moves, here Delphox uses Will-O-Wisp which burns the opponent.



There are many status effects like Burn, Poison, Paralysis. These statuses will have a negative effect on your Pokemon. As you can see here the Burn status effect damages the Pokemon overtime until it faints.

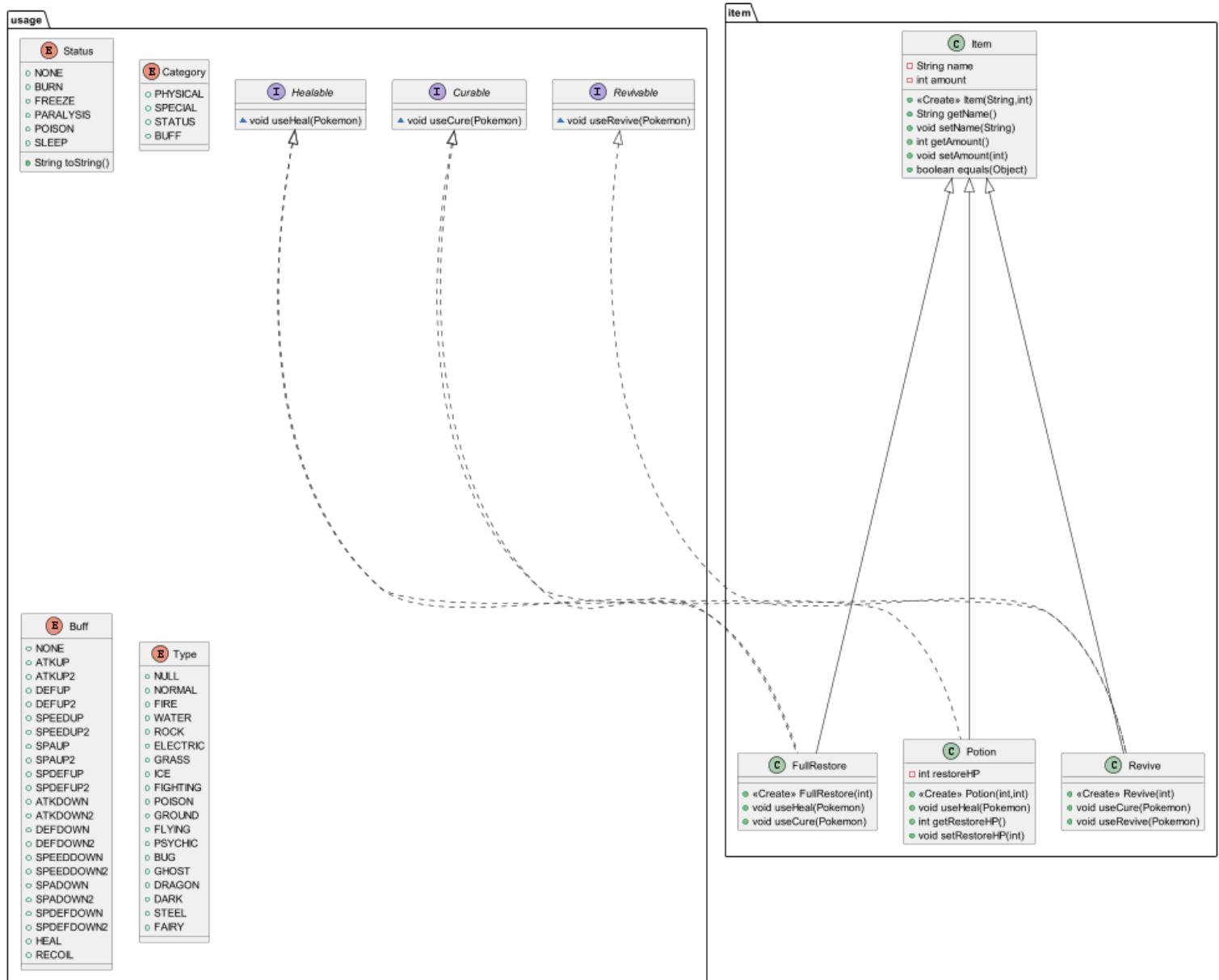


Here we see a Full Restore in action, When Player 2 uses a Full Restore on Umbreon, Umbreon's HP was restored and its status was cured.



When the game ends, the winner will be shown and players can click on "Back to Main Menu" to start a new game.

Class Diagram



game

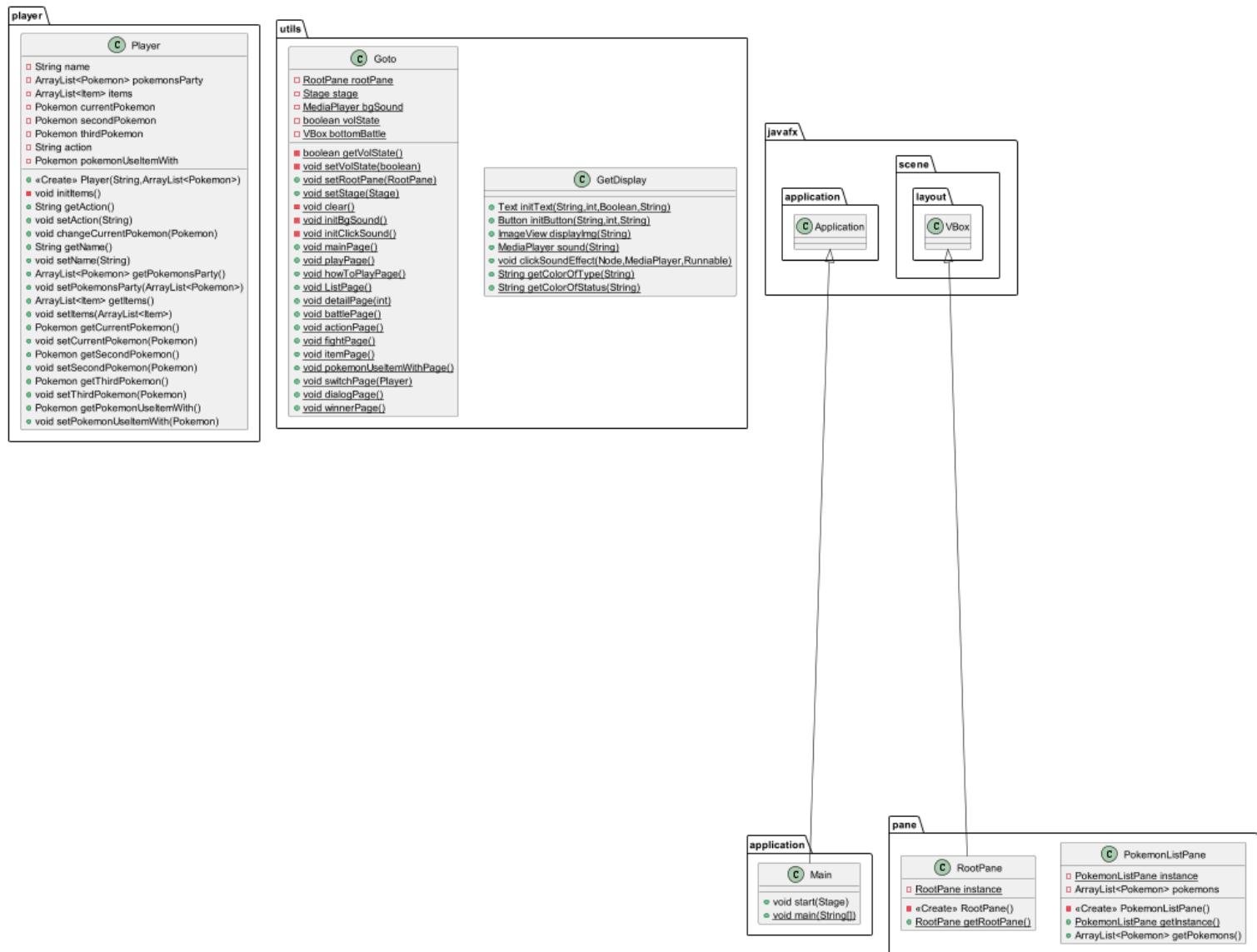


skill



pokemon\





+ (public)
 # (protected)
 - (private)
 underlined (static)
 ALL_CAPS (final variable)
italic (*abstract*)

1. Package Game

1.1 Class GameController

1.1.1 Fields

+ <u>GameController</u> instance	GameController's instance
- ArrayList<Player> players	ArrayList of all players in the game
- String playerSelectTurn	Player who selects pokemon
- String playerPlayTurn	Player who is currently playing
- int indexPlayerPlayTurn	Index of the currently playing player
- int indexRivalPlayTurn	Index of the opposing player
- ArrayList<String> actions	List of player's action
- boolean isFainted	True when currentPokemon of Player 1 or Player 2 has fainted
- boolean isGameEnded	True when all of the pokemon in the Party of Player 1 or Player 2 have fainted, or when Player 1 or Player 2 has surrendered
- String winner	Winner's name

1.1.2 Methods

- GameController()	Set players to new ArrayList, add 2 players named "Player 1" and "Player 2" to players, and call startGame() method
+ <u>GameController</u> getInstance()	Return the instance of GameController
- void startGame()	Set playerSelectTurn to "Player 1". playerPlayTurn to "Player 1", indexPlayerPlayTurn to 0, indexRivalPlayTurn to 1, actions to new ArrayList, isFainted to false, and isGameEnded to false
+ void newGame()	New GameController
+ getters/setters	Getters and setters of all fields

1.2 Class GameUtils

1.2.1 Fields

- <u>Player first, last</u>	Player who start actions first and Player who started next
-----------------------------	--

1.2.2 Methods

+ void choosePokemon(<u>Pokemon</u> pokemon)	Set player's party to that of the Pokemon they selected
+ boolean isFinishChoose()	If both players have selected their pokemon, return true
+ void switchPlayerPlay()	After the player turn is over, this will switch perspectives to the other player
+ void startAction(<u>Player</u> first, <u>Player</u> opponent)	Start actions received from the player, set actions in GameController and output them onto the dialog page.
+ void startTurn(<u>Player</u> p1, <u>Player</u> p2)	Starting turn by utilizing method from <u>whoFirst</u> and <u>startAction</u>
+ void whoFirst(<u>Player</u> p1, <u>Player</u> p2)	Checks which side goes first, if a Player switches or uses an item, it will always go first. If a player uses an attacking move, check the speed of both pokemon
+ void setPokemonInParty()	Check which Pokemon are in the 2nd and 3rd place when you enter the switch menu
+ getters/setters	Getters and setters of all fields

2. Package Item

2.1 Class Item

2.1.1 Field

- String name	Name of the item
- int amount	Amount of the Item

2.1.2 Constructor

+ Item(String name, int amount)	Set fields to the given parameter values
---------------------------------	--

2.1.3 Methods

+ getters/setters	Getters and setters of all fields
-------------------	-----------------------------------

2.2 Class Potion extends Item implements Healable

2.2.1 Field

- restoreHP	Amount of restore pokemon's Hp
-------------	--------------------------------

2.2.2 Constructor

+ Potion(int restoreHP, int amount)	Initialize the Potion Set name to "Potion", restoreHp to restoreHP, and amount to amount. (use super from item)
-------------------------------------	---

2.2.3 Methods

+ void useHeal(Pokemon target)	Heals target Pokemon.
+ getters/setters	Getters and setters of all fields.

2.3 Class Revive extends Item implements Revivable, Healable, Curable

2.3.1 Constructor

+ Revive(int amount)	Initialize the Revive Set name to "Revive" and amount to amount. (use super from item)
----------------------	--

2.3.2 Methods

+ void useCure(Pokemon target)	Cures status of target Pokemon.
+ void useRevive(Pokemon target)	Revives target Pokemon and heals it to half of maxHP.

2.4 Class FullRestore extends Item implements Healable, Curable

2.4.1 Constructor

+ FullRestore(int amount)	Initialize the FullRestore. Set name to "Full Restore" and amount to amount. (use super from item)
---------------------------	--

2.4.2 Methods

+ void useHeal(Pokemon target)	Heals target Pokemon to maxHP.
+ void useCure(Pokemon target)	Cures status of target Pokemon.

3. Package Pokemon

3.1 Class Pokemon

3.1.1 Field

- String name	Name of the Pokemon.
- Type type	Type of the Pokemon.
- Type type2	Type of the Pokemon.
- int hp	HP of the Pokemon.
- int maxHp	MaxHP of the Pokemon.
- int atk	Atk of the Pokemon, used to calculate physical moves.
- int def	Def of the Pokemon, used to calculate physical damage.
- int spAtk	Special Atk of the Pokemon, used to calculate special moves.
- int spDef	Special Def of the Pokemon, used to calculate special damage.
- int spd	Speed of the Pokemon.
- String imgsrc	Image path of the Pokemon.
- Status status	Status of the Pokemon.
- ArrayList<BaseSkill> moves	4 moves that the Pokemon can use.

3.1.2 Constructor

+ Pokemon(String name, Type type, Type type2, int hp, int atk, int def, int spAtk, int spDef, int spd, String imgsrc)	Set fields to the given parameter values.
---	---

3.1.3 Methods

+ void checkStatus	Check if the Pokemon has a status, if yes does the status effect
+void checkSleep	Check if the Pokemon has the “sleep” status, if yes it will immobilize it for some turns
+void checkFrozen	Check if the Pokemon has the “freeze” status, if yes it will immobilize it for some turns
+ getters/setters	Getters and setters of all fields.

4. Package Player

4.1 Class Player

4.1.1 Fields

- String name	The player's name.
- ArrayList<Pokemon> pokemonsParty	The list of player's pokemons that can be chosen in a player's turn.
- ArrayList<Item> items	The list of player's items that can be used in a player's turn.
- Pokemon currentPokemon	Current pokemon that is used in a player's turn.
- Pokemon secondPokemon	The second pokemon in party.
- Pokemon thirdPokemon	The second pokemon in party.
- String action	Action of the player in each turn.
- Pokemon pokemonUseItemWith	The Pokémon that the player wants to select to use an item.

4.1.2 Constructor

+ Player(String name, ArrayList<Pokemon> pokemonsParty)	Set fields to the given parameter values, and Starting item for player.
---	---

4.1.3 Methods

- void initItems()	Initializes the item list.
+ void changeCurrentPokemon(Pokemon newPokemon)	Changes the current Pokemon.
+ getters/setters	Getters and setters of all fields.

5. Package Skill

5.1 Class BaseSkill

5.1.1 Field

- String name	The skill's name
- Type type	Type of the skill
- Category category	Category of the Skill
- Status status	For skills that implements a debuff, buff or a status effect on the opponent
- int pp	Amount of PP
- int power	Amount of power

- int accuracy	Amount of accuracy
- int statusChance	If the status attack deals damage, this calculates the chance for the status taking effect

5.1.2 Constructor

+ BaseSkill(String name, Type type, Category category, Status status, int pp, int power, int statusChance, int accuracy)	Set fields to the given parameter values
--	--

5.1.3 Methods

+ void useSkill(Pokemon opponent, Pokemon user)	Use the attack on the opponent's Pokemon
+ getters/setters	Getters and setters of all fields

6. Package Application

6.1 Class Main extends Application

6.1.1 Methods

+ void start(Stage stage)	Set scene to RootPane. Show stage
+ static void main(String[] args)	Main application

7. Package Pane

7.1 Class RootPane

7.1.1 Field

- RootPane instance	RootPane's instance
---------------------	---------------------

7.1.2 Constructor

- RootPane()	Set background color to white Set alignment to TOP_CENTER Set Goto's rootPane to this Goto mainPage
--------------	--

7.1.3 Methods

+ RootPane getRootPane()	Return the instance of RootPane
--------------------------	---------------------------------

7.2 Class PokemonListPane

7.2.1 Field

- PokemonListPane instance	PokemonListPane's instance
- ArrayList<Pokemon> pokemons	All pokemons

7.2.2 Methods

- <u>PokemonListPane()</u>	Add new Pokemon to pokemons
+ getters	Getters of all fields

8. Package Usage

8.1 Interface Curable

8.1.1 Methods

- void useCure(Pokemon target)	Cures status of target Pokemon
--------------------------------	--------------------------------

8.2 Interface Healable

8.2.1 Methods

- void useHeal(Pokemon target)	Heals target Pokemon
--------------------------------	----------------------

8.3 Interface Revivable

8.3.1 Methods

- void useRevive(Pokemon target)	Revives target Pokemon
----------------------------------	------------------------

8.4 Enum Status

This enum is for Pokemon status. There are NONE, BURN, FREEZE, PARALYSIS, POISON, and SLEEP.

8.5 Enum Type

This enum is for Pokemon type. There are NULL, NORMAL, FIRE, WATER, ELECTRIC, GRASS, ICE, FIGHTING, POISON, GROUND, FLYING, PSYCHIC, BUG, ROCK, GHOST, DRAGON, DARK, STEEL, and FAIRY.

8.6 Enum Category

This enum is for the category of Pokemon moves, it is split into PHYSICAL, SPECIAL and STATUS

8.7 Buff Category

This enum is for the buffs and debuffs Pokemon can receive, it ranges from increasing and decreasing stats, healing and recoil

9. Package Utils

9.1 Class GetDisplay

9.1.1 Methods

+ <u>Text initText(String title, int size, Boolean bold, String fontFamily)</u>	Create a new Text with the string title Set text color to black Set font family to fontFamily If bold is true, set font weight to bold Otherwise, set font weight to normal Set font size to size Return created Text
---	---

+ <u>Button initButton(String text, int width, String bgcolor)</u>	Create a new Button with the string text Set background radius to 15px. Set background color to bgcolor. Set text color to white. Set the font family to null. Set font weight to BOLD. Set font size to 35. Set prefers width to width. Return created Button.
+ <u>ImageView displayImg(String imgPath)</u>	Create a new ImageView. Set image of ImageView to Image from imgPath. Return created ImageView.
+ <u>MediaPlayer sound(String soundPath)</u>	Create a new Media from soundPath. Create a new MediaPlayer from Media that has been created. Return created MediaPlayer
+ <u>void clickSoundEffect(Node clickNode, MediaPlayer clickSound, Runnable onReleaseAction)</u>	SetOnMousePressed of clickNode to run clickSound.stop() and clickSound.play(). SetOnMouseReleased of clickNode to run onReleaseAction.
+ <u>String getColorOfType(String type)</u>	Return color code based on type input.
+ <u>String getColorOfStatus(String status)</u>	Return color code based on status input.

9.2 Class Goto

9.1.1 Fields

- <u>RootPane rootPane</u>	RootPane's instance
- <u>Stage stage</u>	Main stage of application
- <u>MediaPlayer bgSound</u>	Background sound
- <u>MediaPlayer clickSound</u>	Click sound
- <u>boolean volState</u>	State of background sound
- <u>VBox bottomBattle</u>	The bottom part from the battle page.

9.1.2 Methods

+ <u>void setRootPane(RootPane rootPane)</u>	Setter for rootPane field.
+ <u>void setStage(Stage stage)</u>	Setter for stage field.
- <u>void clear()</u>	Remove all children from the RootPane's children.
- <u>void initBgSound()</u>	Init Background Sound

- <u>void initClickSound()</u>	Init sound when clicked.
+ <u>void mainPage()</u>	Remove all children from the RootPane's children, and show the main page. There are 3 buttons: Play which takes you to playPage, How to play which takes you to howToPlayPage() and Exit which exists the program
- <u>void playPage()</u>	Remove all children from the RootPane's children, and display the Select pokemon page. The Pokemon slots will show "Select Pokemon" until the player selects a pokemon, after that it will show the Pokemon's name and image
- <u>void howToPlayPage()</u>	Remove all children from the RootPane's children, and display instructions on how to play
- <u>void listPage()</u>	Remove all children from the RootPane's children, and show the list of all Pokemon available
- <u>void detailPage()</u>	Remove all children from the RootPane's children, and show the selected Pokemon's details
- <u>void battlePage()</u>	Remove all children from the RootPane's children. <ul style="list-style-type: none"> - Display details of battle's gameplay such as pokemon status,pokemon hp. - Set image and name of both pokemon, the pokemon on the left side is the player's side. the pokemon on the right side is the opponent's side
- <u>void actionPage()</u>	Remove all children from the RootPane's children. <ul style="list-style-type: none"> - Display the 4 main options a Player can choose: Fight, Item, Switch and Surrender clicking on one of these buttons will send you to the corresponding page.
- <u>void fightPage()</u>	Remove all children from the RootPane's children. <ul style="list-style-type: none"> - Display the 4 moves a Pokemon can you, as well as the amount of PP left - If a skill's PP reaches 0, the color will turn

	black and you cannot select the move
- <u>void itemPage()</u>	<p>Remove all children from the RootPane's children.</p> <p>-Display the 3 items in player's item list, and if the number of items still hasn't reached 0, the player can select it</p> <p>-When a player selects an item, navigate to <u>pokemonUseItemWithPage()</u> to allow the player to choose a pokemon to use the item with.</p>
- <u>void pokemonUseItemWithPage()</u>	<p>Remove all children from the RootPane's children.</p> <p>-Display all the Pokemon currently in the player's party, selecting one of them will use the item currently selected on the Pokemon.</p> <p>-If the player selects a potion or a full restore, they can only use it on pokemon that are not fainted. If the player chooses to use a revive, they can only use it on pokemon that have already fainted.</p>
- <u>void switchPage()</u>	<p>Remove all children from the RootPane's children.</p> <p>-Display all the Pokemon currently in the player's party, selecting one of them will switch the player's current Pokemon to that Pokemon</p>
- <u>void dialogPage()</u>	<p>Remove all children from the bottomBattle's children</p> <p>-Show detail of actions that players have done in a round</p> <p>-Init new button</p> <p>If the game ends, navigate to <u>winnerPage()</u></p> <p>If the last player's action does not equal null and the last player's pokemon hasn't yet fainted, navigate to <u>actionPage()</u> of the first player</p> <p>If the first player's pokemon faints, navigate to <u>switchPage()</u> of the first player</p> <p>If the last player's pokemon faints, navigate to <u>switchPage()</u> of the last player</p> <p>If <u>getAction()</u> of both players has no children, navigate to <u>actionPage()</u></p>
- <u>void winnerPage()</u>	<p>Remove all children from the RootPane's children</p> <p>Show the winner player</p> <p>Set button to navigate to <u>mainPage()</u></p>