Individual Report

ST10438528 – David Barker

Table of Contents

[List of Names for Each Component 3](#_Toc164444041)

[StartGameForm 3](#_Toc164444042)

[TurnPlayerForm 4](#_Toc164444043)

[Functionality of the Game 5](#_Toc164444044)

[What Each Form Does 5](#_Toc164444045)

[StartGameForm 5](#_Toc164444046)

[TurnPlayerForm 5](#_Toc164444047)

[User Input 5](#_Toc164444048)

[StartGameForm 5](#_Toc164444049)

[TurnPlayerForm 5](#_Toc164444050)

[Output 5](#_Toc164444051)

[StartGameForm 5](#_Toc164444052)

[TurnPlayerForm 5](#_Toc164444053)

[Why Output Is Generated 6](#_Toc164444054)

[StartGameForm 6](#_Toc164444055)

[TurnPlayerForm 6](#_Toc164444056)

[High-Level Logic 7](#_Toc164444057)

[StartGameForm 7](#_Toc164444058)

[TurnPlayerForm 12](#_Toc164444059)

# List of Names for Each Component

## StartGameForm

|  |  |  |
| --- | --- | --- |
| **Type** | **Name** | **Description** |
| Label | lblPlayer1 | Has the words “Player 1” to help player one to know where to type their information |
| Panel | pnlPlayer1 | Holds all the inputs for player one’s information |
| Label | lblPlayerName1 | Indicates where player one should input their name |
| Textbox | txtPlayerName1 | The area where player one types their name |
| Label | lblDragonName1 | Indicates where player one should input their dragon’s name |
| Textbox | txtDragonName1 | The area where player one types their dragon’s name |
| Label | lblDragonType1 | Indicates where player one can select their dragon type |
| Panel | pnlDragonType1 | Holds the radio buttons for player one to select their dragon type |
| Radio Button | rbtnFireDragon1 | Used for player one to select that their dragon is a Fire Dragon |
| Radio Button | rbtnIceDragon1 | Used for player one to select that their dragon is an Ice Dragon |
| Radio Button | rbtnWindDragon1 | Used for player one to select that their dragon is a Wind Dragon |
| Radio Button | rbtnEarthDragon1 | Used for player one to select that their dragon is an Earth Dragon |
| Picture Box | picDragon1 | Shows a picture of player one’s selected dragon |
| Button | btnSave1 | Saves player one’s information and their dragon’s information into the arrays that hold these values |
| Label | lblPlayer2 | Has the words “Player 2” to help player two to know where to type their information |
| Panel | pnlPlayer2 | Holds all the inputs for player two’s information |
| Label | lblPlayerName2 | Indicates where player one should input their name |
| Textbox | txtPlayerName2 | The area where player one types their name |
| Label | lblDragonName2 | Indicates where player one should input their dragon’s name |
| Textbox | txtDragonName2 | The area where player one types their dragon’s name |
| Label | lblDragonType2 | Indicates where player two can select their dragon type |
| Panel | pnlDragonType2 | Holds the radio buttons for player two to select their dragon type |
| Radio Button | rbtnFireDragon2 | Used for player two to select that their dragon is a Fire Dragon |
| Radio Button | rbtnIceDragon2 | Used for player two to select that their dragon is an Ice Dragon |
| Radio Button | rbtnWindDragon2 | Used for player two to select that their dragon is a Wind Dragon |
| Radio Button | rbtnEarthDragon2 | Used for player two to select that their dragon is an Earth Dragon |
| Picture Box | picDragon2 | Shows a picture of player two’s selected dragon |
| Button | btnSave2 | Saves player two’s information and their dragon’s information into the arrays that hold these values |
| Label | lblFireDragonStats | Shows a picture of a Fire Dragon and gives it statistics |
| Label | lblIceDragonStats | Shows a picture of an Ice Dragon and gives it statistics |
| Label | lblWindDragonStats | Shows a picture of a Wind Dragon and gives it statistics |
| Label | lblEarthDragonStats | Shows a picture of an Earth Dragon and gives it statistics |
| Button | btnStartGame | Saves the array values to arrays in TurnPlayerForm, hides the current form and shows the TurnPlayerForm |

## TurnPlayerForm

|  |  |  |
| --- | --- | --- |
| **Type** | **Name** | **Description** |
| Label | lblDragonNameType1 | Shows the name and type of the dragon who’s turn it is |
| Panel | pnlPlayerMove | Holds the buttons that are used in a turn and information on the dragon who’s turn it is |
| Label | lblHp1 | Shows the health of the dragon who’s turn it is |
| Button | btnAttack | The current dragon attacks |
| Button | btnSpAttack | The current dragon does a special attack |
| Button | btnBlock | The current dragon prepares to block the opponent’s next attack |
| Button | btnRest | The current dragon rests, only needed the next turn after a special attack |
| Label | lblOpponent | Shows the opponent’s name |
| Panel | pnlOpponent | Holds all the components which show the opponent’s information |
| Label | lblDragonNameType2 | Shows the name and type of the opponent’s dragon |
| Label | lblHp2 | Shows the health of the opponent’s dragon |
| Picture Box | picPlayer1Dragon | Shows a picture of player one’s dragon |
| Label | lblVs | Reads “VS” to show player one vs player two |
| Picture Box | picPlayer2Dragon | Shows a picture of player two’s dragon |
| Label | lblBattleLog | Indicates where the battle log is |
| Panel | pnlBattleLog | Holds the panel which holds the battle log |
| Panel | pnlBattleLogReport | Holds the battle log and allows you to scroll through it |
| Label | lblBattle | The battle log which shows the current turn and all other turns up to that point |

# Functionality of the Game

## What Each Form Does

### StartGameForm

Allows both players to enter their names, dragon’s names and dragon’s types save them and load the TurnPlayerForm to start battling with the information they provided.

### TurnPlayerForm

Simulates combat allowing the dragons to take action and displaying how each action turns out.

## User Input

### StartGameForm

Player one can input their name through a textbox, their dragon’s name through a textbox and their dragon’s type through a group of four radio buttons. They can then save this information into the player one arrays by clicking their “Save” button.

Player two can do the same thing.

Once both “Save” buttons have been pressed, the “Start Game” button can be clicked to load the TurnPlayerForm.

### TurnPlayerForm

When it is the player’s turn, they can click either the “Attack”, “Special Attack” or the “Block” buttons to do the actions which their name’s imply.

If the player did a special attack on their previous turn, then they can only click the “Rest” button. Once they do, the dragon “rests” and play changes.

## Output

### StartGameForm

Colours change according to input. Different dragons have different coloured panels and text, the save button turns the player’s panel green, and once both players have saved the “Start Game” button turns from red to white.

### TurnPlayerForm

It shows each dragon’s health, it shows who takes initiative, who’s turn it is and what they do on that turn in the battle log.

## Why Output Is Generated

### StartGameForm

The different colours provide feedback to the players. The colours for dragons make it easier to know which dragon they’ve selected. The green panel when a player has saved lets each player know who’s saved. The red to white “Start Game” button lets the players know whether they can start the game or not.

### TurnPlayerForm

The health is shown so that the player can make decisions based on that. The battle log provides the information said previously so that players can see what is going on during the whole battle.

# High-Level Logic

## StartGameForm

start

Declarations

TextBox txtPlayerName1 = new TextBox()

TextBox txtDragonName1 = new TextBox()

RadioButton rbtnFireDragon1 = new RadioButton()

RadioButton rbtnIceDragon1 = new RadioButton()

RadioButton rbtnWindDragon1 = new RadioButton()

RadioButton rbtnEarthDragon1 = new RadioButton()

Button btnSave1 = new Button()

TextBox txtPlayerName2 = new TextBox()

TextBox txtDragonName2 = new TextBox()

RadioButton rbtnFireDragon2 = new RadioButton()

RadioButton rbtnIceDragon2 = new RadioButton()

RadioButton rbtnWindDragon2 = new RadioButton()

RadioButton rbtnEarthDragon2 = new RadioButton()

Button btnSave2 = new Button()

Button btnStartGame = new Button()

TurnPlayerForm turnPlayer

string p1Data[3]

string p2Data[3]

num p1Values[4]

num p2Values[4]

string FIRE\_DRAG\_NAME = "Fire Dragon"

num FIRE\_DRAG\_HP = 20

num FIRE\_DRAG\_ATK = 5

num FIRE\_DRAG\_SPATK = 12

num FIRE\_DRAG\_BLOCK = 4

string ICE\_DRAG\_NAME = "Ice Dragon"

num ICE\_DRAG\_HP = 30

num ICE\_DRAG\_ATK = 4

num ICE\_DRAG\_SPATK = 9

num ICE\_DRAG\_BLOCK = 5

string WIND\_DRAG\_NAME = "Wind Dragon"

num WIND\_DRAG\_HP = 40

num WIND\_DRAG\_ATK = 3

num WIND\_DRAG\_SPATK = 7

num WIND\_DRAG\_BLOCK = 5

string EARTH\_DRAG\_NAME = "Earth Dragon"

num EARTH\_DRAG\_HP = 50

num EARTH\_DRAG\_ATK = 2

num EARTH\_DRAG\_SPATK = 5

num EARTH\_DRAG\_BLOCK = 6

boolean player1Saved = false // Used to make sure that player 1 has clicked save, so that it is safe to load a new TurnPlayerForm

boolean player2Saved = false // Used to make sure that player 2 has clicked save, so that it is safe to load a new TurnPlayerForm

btnSave1.registerListener(btnSave1Click())

btnSave2.registerListener(btnSave2Click())

btnStartGame.registerListener(btnStartGameClick())

stop

// Calls the saveValues module using the data and values provided by player 1 as arguments

private void btnSave1Click()

Declarations

num PLAYER\_NUMBER = 1

string playerName1

string dragonName1

string type

num hp

num atk

num spatk

num block

playerName1 = txtPlayerName1.getText()

dragonName1 = txtDragonName1.getText()

if playerName1 <> "" AND dragonName1 <> "" then // Check inputs are valid

if rbtnFireDragon1.getCheckedStatus() then

type = FIRE\_DRAG\_NAME

hp = FIRE\_DRAG\_HP

atk = FIRE\_DRAG\_ATK

spatk = FIRE\_DRAG\_SPATK

block = FIRE\_DRAG\_BLOCK

endif

if rbtnIceDragon1.getCheckedStatus() then

type = ICE\_DRAG\_NAME

hp = ICE\_DRAG\_HP

atk = ICE\_DRAG\_ATK

spatk = ICE\_DRAG\_SPATK

block = ICE\_DRAG\_BLOCK

endif

if rbtnWindDragon1.getCheckedStatus() then

type = WIND\_DRAG\_NAME

hp = WIND\_DRAG\_HP

atk = WIND\_DRAG\_ATK

spatk = WIND\_DRAG\_SPATK

block = WIND\_DRAG\_BLOCK

endif

if rbtnEarthDragon1.getCheckedStatus() then

type = EARTH\_DRAG\_NAME

hp = EARTH\_DRAG\_HP

atk = EARTH\_DRAG\_ATK

spatk = EARTH\_DRAG\_SPATK

block = EARTH\_DRAG\_BLOCK

endif

saveValues(PLAYER\_NUMBER, playerName1, dragonName1, type, hp, atk, spatk, block)

player1Saved = true

endif

return

// Calls the saveValues module using the data and values provided by player 2 as arguments

private void btnSave2Click()

Declarations

num PLAYER\_NUMBER = 2

string playerName2

string dragonName2

string type

num hp

num atk

num spatk

num block

playerName2 = txtPlayerName2.getText()

dragonName2 = txtDragonName2.getText()

if playerName2 <> "" AND dragonName2 <> "" then // Check inputs are valid

if rbtnFireDragon2.getCheckedStatus() then

type = FIRE\_DRAG\_NAME

hp = FIRE\_DRAG\_HP

atk = FIRE\_DRAG\_ATK

spatk = FIRE\_DRAG\_SPATK

block = FIRE\_DRAG\_BLOCK

endif

if rbtnIceDragon2.getCheckedStatus() then

type = ICE\_DRAG\_NAME

hp = ICE\_DRAG\_HP

atk = ICE\_DRAG\_ATK

spatk = ICE\_DRAG\_SPATK

block = ICE\_DRAG\_BLOCK

endif

if rbtnWindDragon2.getCheckedStatus() then

type = WIND\_DRAG\_NAME

hp = WIND\_DRAG\_HP

atk = WIND\_DRAG\_ATK

spatk = WIND\_DRAG\_SPATK

block = WIND\_DRAG\_BLOCK

endif

if rbtnEarthDragon2.getCheckedStatus() then

type = EARTH\_DRAG\_NAME

hp = EARTH\_DRAG\_HP

atk = EARTH\_DRAG\_ATK

spatk = EARTH\_DRAG\_SPATK

block = EARTH\_DRAG\_BLOCK

endif

saveValues(PLAYER\_NUMBER, playerName2, dragonName2, type, hp, atk, spatk, block)

player2Saved = true

endif

return

// If both players have saved it creates a new instance of a TurnPlayerForm, calls its saveValues module passing the arrays as arguments, switches between forms

private void btnStartGameClick()

if player1Saved AND player2Saved then

turnPlayer = new TurnPlayerForm()

turnPlayer.saveValues(p1Data, p2Data, p1Values, p2Values)

turnPlayer.show()

this.hide()

endif

return

// Saves the data and values provided by the parameters into the arrays that correspond to the player number provided by the parameters

private void saveValues(num playerNum, string playerName, string dragonName, string dragType, num dragHp, num dragAtk, num dragSpatk, num dragBlock)

if playerNum == 1 then

p1Data[0] = playerName

p1Data[1] = dragonName

p1Data[2] = dragType

p1Values[0] = dragHp

p1Values[1] = dragAtk

p1Values[2] = dragSpatk

p1Values[3] = dragBlock

else

p2Data[0] = playerName

p2Data[1] = dragonName

p2Data[2] = dragType

p2Values[0] = dragHp

p2Values[1] = dragAtk

p2Values[2] = dragSpatk

p2Values[3] = dragBlock

endif

return

## TurnPlayerForm

start

Declarations

Button btnAttack = new Button()

Button btnSpAttack = new Button()

Button btnBlock = new Button()

Button btnRest = new Button()

Label lblBattle = new Label()

Label lblDragonNameType1 = new Label()

Label lblHp1 = new Label()

Label lblOpponent = new Label()

Label lblDragonNameType2 = new Label()

Label lblHp2 = new Label()

string p1Data[]

string p2Data[]

num p1Values[]

num p2Values[]

num playerTurn

boolean p1IsResting = false // Used to check if player 1 needs to rest

boolean p2IsResting = false // Used to check if player 2 needs to rest

boolean p1IsBlocking = false // Used to check if player 1 will block player 2’s attack

boolean p2IsBlocking = false // Used to check if player 2 will block player 1’s attack

boolean p1HasPlayed = false // Used to check if player 1 has played this round, if both players have played then it’s a new round

boolean p2HasPlayed = false // Used to check if player 2 has played this round, if both players have played then it’s a new round

btnAttack.registerListener(btnAttackClick())

btnSpAttack.registerListener(btnSpAttackClick())

btnBlock.registerListener(btnBlockClick())

btnRest.registerListener(btnRestClick())

btnRest.setVisibile(false)

playerTurn = takeInitiative()

switchPlayer()

stop

// Dragon attacks their opponent when the button is clicked

private void btnAttackClick()

Declarations

num hp

num atk

num block

num damage

string attacker

string defender

string blockMessage

string battleText = lblBattle.getText()

if playerTurn == 1 then

hp = p2Values[0]

atk = p1Values[1]

attacker = p1Data[1]

defender = p2Data[1]

if p2IsBlocking then

block = p2Values[3]

damage = atk - block

if damage < 0 then

damage = 0

endif

blockingMessage = " blocks it and"

else

damage = atk

blockingMessage = ""

endif

hp = hp - damage

if hp < 0 then

hp = 0

endif

p2Values[0] = hp

p1HasPlayed = true

else

hp = p1Values[0]

atk = p2Values[1]

attacker = p2Data[1]

defender = p1Data[1]

if p1IsBlocking then

block = p1Values[3]

damage = atk - block

if damage < 0 then

damage = 0

endif

blockingMessage = " blocks it and"

else

damage = atk

blockingMessage = ""

endif

hp = hp - damage

if hp < 0 then

hp = 0

endif

p1Values[0] = hp

p2HasPlayed = true

endif

battleText = battleText, attacker, " attacks ", defender, "! ", defender, blockMessage, " takes ", damage, " damage. ", defender " is now on ", hp, " HP\n"

battleText = battleText, "--------------------------------------------------------------------------\n"

lblBattle.setText(battleText)

switchPlayer()

return

// Dragon uses a special attack on their opponent when the button is clicked

private void btnSpAttackClick()

Declarations

num hp

num spatk

num block

num damage

string attacker

string defender

string blockMessage

string battleText = lblBattle.getText()

if playerTurn == 1 then

hp = p2Values[0]

spatk = p1Values[2]

attacker = p1Data[1]

defender = p2Data[1]

if p2IsBlocking then

block = p2Values[3]

damage = spatk - block

if damage < 0 then

damage = 0

endif

blockingMessage = " blocks it and"

else

damage = spatk

blockingMessage = ""

endif

hp = hp - damage

if hp < 0 then

hp = 0

endif

p2Values[0] = hp

p1IsResting = true

p1HasPlayed = true

else

hp = p1Values[0]

spatk = p2Values[2]

attacker = p2Data[1]

defender = p1Data[1]

if p1IsBlocking then

block = p1Values[3]

damage = spatk - block

if damage < 0 then

damage = 0

endif

blockingMessage = " blocks it and"

else

damage = spatk

blockingMessage = ""

endif

hp = hp - damage

if hp < 0 then

hp = 0

endif

p1Values[0] = hp

p2IsResting = true

p2HasPlayed = true

endif

battleText = battleText, attacker, " special attacks ", defender, "! ", defender, blockMessage, " takes ", damage, " damage. ", defender " is now on ", hp, " HP\n"

battleText = battleText, "--------------------------------------------------------------------------\n"

lblBattle.setText(battleText)

switchPlayer()

return

// Enables the dragon to block their opponent’s next attack when the button is clicked

private void btnBlockClick()

Declarations

string currentDragon

string nextDragon

string battleText = lblBattle.getText()

if playerTurn == 1 then

currentDragon = p1Data[1]

nextDragon = p2Data[1]

p1IsBlocking = true

p1HasPlayed = true

else

currentDragon = p2Data[1]

nextDragon = p1Data[1]

p2IsBlocking = true

p2HasPlayed = true

endif

battleText = battleText, currentDragon, " prepares to block ", nextDragon, "'s next attack\n"

battleText = battleText, "--------------------------------------------------------------------------\n"

lblBattle.setText(battleText)

switchPlayer()

return

// Calls the rest module when the button is clicked

private void btnRestClick()

rest(playerTurn)

return

// Determines who goes first this round by using the randomRoll module, whoever has the higher roll will go first

private num takeInitiative()

Declarations

num p1Roll

num p2Roll

num startPlayer

p1Roll = randomRoll()

p2Roll = randomRoll()

while p1Roll == p2Roll

p1Roll = randomRoll()

p2Roll = randomRoll()

endwhile

if p1Roll > p2Roll then

playerTurn = 2 // Inverted so that when switchPlayer() is called after takeInitiative(), it will switch to the correct starting player (p1)

else

playerTurn = 1 // Inverted so that when switchPlayer() is called after takeInitiative(), it will switch to the correct starting player (p2)

endif

return startPlayer

// Random roll between 1 and 6 inclusive, which will be used when taking initiative

private num randomRoll()

Declarations

num roll

num MIN = 1

num MAX = 6

Random random = new Random()

roll = random.next(MIN, MAX) // Min value and max value are inclusive

return roll

// Used to save the values stored in the arrays in the StartGameForm in the arrays in the current TurnPlayerForm

public void saveValues(string p1Data[], string p2Data[], num p1Values[], num p2Values[])

this.p1Data = p1Data

this.p2Data = p2Data

this.p1Values = p1Values

this.p2Values = p2Values

return

// Used to switch players, end the current player’s blocking status and update the GUI to show the current dragon/opponent and if they can take an action or not

private void switchPlayer()

Declarations

string player

string dragon

string type

num hp

string opponent

string opponentDragon

string opponentType

num opponentHp

string battleText = lblBattle.getText()

if p1HasPlayed AND p2HasPlayed then // Begin new round

p1HasPlayed = false

p2HasPlayed = false

takeInitiative()

switchPlayer()

return // Early return since the battle log has already been updated through the previous switchPlayer() call and we don't want it to update again

endif

if playerTurn == 1 then

playerTurn = 2

player = p2Data[0]

dragon = p2Data[1]

type = p2Data[2]

hp = p2Values[0]

opponent = p1Data[0]

opponentDragon = p1Data[1]

opponentType = p1Data[2]

opponentHp = p1Values[0]

if p2IsResting then

btnAttack.setVisibile(false)

btnSpAttack.setVisibile(false)

btnBlock.setVisibile(false)

btnRest.setVisibile(true)

endif

if p2IsBlocking then

p2IsBlocking = false

endif

else

playerTurn = 1

player = p1Data[0]

dragon = p1Data[1]

type = p1Data[2]

hp = p1Values[0]

opponent = p2Data[0]

opponentDragon = p2Data[1]

opponentType = p2Data[2]

opponentHp = p2Values[0]

if p1IsResting then

btnAttack.setVisibile(false)

btnSpAttack.setVisibile(false)

btnBlock.setVisibile(false)

btnRest.setVisibile(true)

endif

if p1IsBlocking then

p1IsBlocking = false

endif

endif

if NOT(p1HasPlayed OR p2HasPlayed) then // Start of new round

battleText = battleText, \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

battleText = battleText, player, "'s dragon ", dragon, " takes inititative!\n"

battleText = battleText, "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

endif

battleText = battleText, player, "'s Turn:\n"

if hp == 0 then // Dead

btnAttack.setEnabled(false)

btnSpAttack.setEnabled(false)

btnBlock.setEnabled(false)

btnRest.setEnabled(false)

battleText = battleText, dragon, " is unable to continue. ", opponentDragon, " is the winner!"

endif

lblBattle.setText(battleText)

lblDragonNameType1.setText(dragon, ", the ", type, "'s Turn")

lblHp1.setText("HP: ", hp) // String concat

lblOpponent.setText("Opponent: ", opponent)

lblDragonNameType2.setText(opponentDragon, ", the ", opponentType)

lblHp2.setText("HP: ", opponentHp)

return

// Updates the current player’s rest status to no longer need to rest, but skips their turn

private void rest(num dragonNum)

Declarations

string dragon

string battleText = lblBattle.getText()

if dragonNum = 1 then

dragon = p1Data[1]

p1IsResting = false

p1HasPlayed = true

else

dragon = p2Data[1]

p2IsResting = false

p2HasPlayed = true

endif

battleText = battleText, dragon, " is to tired tired to fight, and rests a while\n"

battleText = battleText, "--------------------------------------------------------------------------\n"

lblBattle.setText(battleText)

btnAttack.setVisibile(true)

btnSpAttack.setVisibile(true)

btnBlock.setVisibile(true)

btnRest.setVisibile(false)

switchPlayer()

return