**package** KickPunchBlockProject;

**import** java.util.\*;

**public** **class** KickPunchBlockProject

{

// Creates scanner object

**static** Scanner *input* = **new** Scanner(System.***in***);

// Static because variables are not used in a constructor

**static** String *playerOneName* = "";

**static** String *opponentName* = "";

**static** String *country* = "";

**static** String *countryUpperCase* = "";

**static** String *fighterAction*;

**static** String *opponentAction*;

**static** **boolean** *opponentHurt* = **false**;

**static** **boolean** *playerOneHurt* = **false**;

**static** **boolean** *counterAttack* = **false**;

**static** **boolean** *player1ValidName* = **false**;

**static** **boolean** *player2ValidName* = **false**;

**static** **boolean** *isMultiplayer* = **false**;

**static** **byte** *playerOneHealth* = 100;

**static** **byte** *opponentHealth* = 100;

**static** **int** *numOfPlayers*;

**static** **int** *fighterInput*;

**static** **int** *damageRoll*;

**static** **int** *roundNumber* = 1;

**static** **final** **byte** ***DAMAGE\_MULTIPLIER*** = 2;

**static** **final** String ***DIFFICULTY*** = "MEDIUM"; // EASY, MEDIUM, HARD

**static** String[] *actionArray* = {"Kick","Punch","Block"};

**static** String[] *playerOneArray* = {"Superman", "Batman", "Flash", "Wonderwoman", "Aquaman", "Cedric"};

**static** String[] *playerTwoArray* = {"Thor", "Iron Man", "Ant-Man", "Hulk", "Hawkeye"};

**static** String[] *countriesArray* = {"USA", "Russia", "Canada", "Germany", "Wakanda"};

**public** **static** **void** main(String[] args)

{

*introduceGame*();

*pickCountry*();

*pickNumOfPlayers*();

*pickFighter*();

*fight*();

}

**private** **static** **void** fight()

{

**do**

{

*selectActions*();

*checkActions*();

*updateHealth*();

*checkForKnockout*();

// Reset variables for next round

*opponentHurt* = **false**;

*playerOneHurt* = **false**;

*counterAttack* = **false**;

*roundNumber*++;

} **while** (*playerOneHealth* > 0 && *opponentHealth* > 0);

}

**private** **static** **void** announce(String competitor1, String damageDescription, String competitor2)

{

**for** (**int** i = 0; i < 50; i++) System.***out***.println();

*printBorder*();

System.***out***.println("Announcer says, \"" + competitor1 + damageDescription + competitor2 + "\"");

}

**private** **static** **void** printBorder()

{

System.***out***.println("================================================================================\n");

}

**private** **static** **void** checkForKnockout()

{

// Check double knock out

**if** (*playerOneHealth* <= 0 && *opponentHealth* <= 0)

{

System.***out***.println(*playerOneName* + " and " + *opponentName* + " both go down for the count!");

// Prints one to ten because fighter is knocked out

**for** (**int** i = 1; i <= 10; i++)

{

**if** (i < 6) System.***out***.println(i);

**else** System.***out***.println(i + "!");

// Delays count – from StackOverflow

**try**

{

Thread.*sleep*(500);

}

**catch** (InterruptedException e)

{

e.printStackTrace();

}

}

System.***out***.println("\n\*DING\* \*DING\* \*DING\* The match is over in round number " + *roundNumber* + "!!\n" + *playerOneName* + " and " + *opponentName* + " knocked each other out at the same time.\nWhat a weird ending!!!");

}

// Check if Player One Lost

**else** **if** (*playerOneHealth* <= 0)

{

// Prints one to ten because player one is knocked out

System.***out***.println(*playerOneName* + " is down for the count!");

**for** (**int** i = 1; i <= 10; i++)

{

**if** (i < 6) System.***out***.println(i);

**else** System.***out***.println(i + "!");

// Delays count – from StackOverflow

**try**

{

Thread.*sleep*(500);

}

**catch** (InterruptedException exception)

{

exception.printStackTrace();

}

}

// Game Over

System.***out***.println("\n\*DING\* \*DING\* \*DING\* The match is over in round number " + *roundNumber* + "!!\n" + *playerOneName* + " was knocked out, and " + *opponentName* + " still had " + *opponentHealth* + " health left. \nBetter luck next time player one!!!");

}

// Check if Player Two Lost

**else** **if** (*opponentHealth* <= 0)

{

System.***out***.println(*opponentName* + " is down for the count!");

// Prints one to ten because fighter is knocked out

**for** (**int** i = 1; i <= 10; i++)

{

**if**(i < 6)System.***out***.println(i);

**else** System.***out***.println(i + "!");

**try**

{

Thread.*sleep*(500);

}

**catch** (InterruptedException exception)

{

exception.printStackTrace();

}

}

// Game Over

System.***out***.println("\n\*DING\* \*DING\* \*DING\* The match is over in round number " + *roundNumber* + "!! \n" + *opponentName* + " was knocked out, and " + *playerOneName* + " still had " + *playerOneHealth* + " health left.\nCONGRATULATIONS PLAYER ONE!!!");

}

}

// KICK counters BLOCK, PUNCH counters KICK, BLOCK counters PUNCH

**private** **static** **void** checkActions()

{

System.***out***.println();

**if** (*fighterAction*.equals("Kick") && *opponentAction*.equals("Kick"))

{

*announce*(*playerOneName*," smashes shins with ", *opponentName*);

*opponentHurt* = **true**;

*playerOneHurt* = **true**;

}

**else** **if** (*fighterAction*.equals("Kick") && *opponentAction*.equals("Punch"))

{

*announce*(*opponentName*," gives a big uppercut to ", *playerOneName*);

*playerOneHurt* = **true**;

*counterAttack* = **true**;

}

**else** **if** (*fighterAction*.equals("Kick") && *opponentAction*.equals("Block"))

{

*announce*(*opponentName*," takes a big kick to the ribs from ", *playerOneName*);

*opponentHurt* = **true**;

*counterAttack* = **true**;

}

**else** **if** (*fighterAction*.equals("Punch") && *opponentAction*.equals("Kick"))

{

*announce*(*playerOneName*," gives a big uppercut to ", *opponentName*);

*opponentHurt* = **true**;

*counterAttack* = **true**;

}

**else** **if** (*fighterAction*.equals("Punch") && *opponentAction*.equals("Punch"))

{

*announce*(*playerOneName*," exchanges quick jabs with ", *opponentName*);

*opponentHurt* = **true**;

*playerOneHurt* = **true**;

}

**else** **if** (*fighterAction*.equals("Punch") && *opponentAction*.equals("Block"))

{

*announce*(*opponentName*," parries the punch and deals big damage to ", *playerOneName*);

*playerOneHurt* = **true**;

*counterAttack* = **true**;

}

**else** **if** (*fighterAction*.equals("Block") && *opponentAction*.equals("Kick"))

{

*announce*(*playerOneName*," takes a big kick to the ribs from ", *opponentName*);

*playerOneHurt* = **true**;

*counterAttack* = **true**;

}

**else** **if** (*fighterAction*.equals("Block") && *opponentAction*.equals("Punch"))

{

*announce*(*playerOneName*," parries the punch and deals big damage to ", *opponentName*);

*opponentHurt* = **true**;

*counterAttack* = **true**;

}

// Both block

**else**

{

*announce*("","Both fighters stand still, waiting for the other to make a move!","");

}

System.***out***.println();

}

**private** **static** **void** updateHealth()

{

**if** (*counterAttack*)

{

**if** (*playerOneHurt*) *playerOneHealth* -= *opponentDamageRoll*(***DIFFICULTY***) \* ***DAMAGE\_MULTIPLIER***;

**if** (*opponentHurt*) *opponentHealth* -= *playerOneDamageRoll*(***DIFFICULTY***) \* ***DAMAGE\_MULTIPLIER***;

}

**else**

{

**if** (*playerOneHurt*) *playerOneHealth* -= *opponentDamageRoll*(***DIFFICULTY***);

**if** (*opponentHurt*) *opponentHealth* -= *playerOneDamageRoll*(***DIFFICULTY***);

}

}

// Opponent Damage Roll

**private** **static** **int** opponentDamageRoll(String difficulty)

{

**if** (difficulty.equals("EASY")) **return** (**int**) (Math.*random*() \* 15 + 1); // 1 - 15 damage

**else** **if** (difficulty.equals("MEDIUM")) **return** (**int**) (Math.*random*() \* 11 + 5); // 5 - 15 damage

**else** **return** (**int**) (Math.*random*() \* 6 + 10); // 10 - 15

}

// Fighter Damage Roll

**private** **static** **int** playerOneDamageRoll(String difficulty)

{

**if** (difficulty.equals("EASY")) **return** (**int**) (Math.*random*() \* 6 + 10); // 10 - 15 damage

**else** **if** (difficulty.equals("MEDIUM")) **return** (**int**) (Math.*random*() \* 11 + 5); // 5 - 15 damage

**else** **return** (**int**) (Math.*random*() \* 15 + 1); // 1 - 15

}

// Calculate actions and damage

**private** **static** **void** selectActions()

{

// One Player

**if** (!*isMultiplayer*)

{

*printBorder*();

System.***out***.print("ROUND NUMBER " + *roundNumber* + "!\n" + *playerOneName* + " has " + *playerOneHealth* + " health left, and " + *opponentName* + " has " + *opponentHealth* + " health left.\nEnter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt();

// Test crashes with floats or letters

**while** (*fighterInput* < 1 || *fighterInput* > 3)

{

*printBorder*();

System.***out***.print("What are you trying to do???\nYou can only enter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt();

}

// Assigns index value of action array

*fighterAction* = *actionArray*[*fighterInput* - 1];

// Opponent rolls action

*opponentAction* = *actionArray*[(**int**) (Math.*random*() \* 3)];

}

// Two Player

**else**

{

// Player1 picks action and test input

*printBorder*();

System.***out***.print("Round Number " + *roundNumber* + "!\n" + *playerOneName* + " has " + *playerOneHealth* + " health left, and " + *opponentName* + " has " + *opponentHealth* + " health left.\n\n" + *playerOneName* + " enter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt();

**while** (*fighterInput* < 1 || *fighterInput* > 3 )

{

*printBorder*();

System.***out***.print("What are you trying to do???\nYou can only enter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt(); // Throws error if input is not an int

}

*fighterAction* = *actionArray*[*fighterInput* - 1];

// Prints lines to hide player one’s action

**for**(**int** i = 0; i < 50; i++) System.***out***.println();

// Player2 picks action

System.***out***.print(*opponentName* + " enter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt();

**while** (*fighterInput* < 1 || *fighterInput* > 3 )

{

*printBorder*();

System.***out***.print("What are you trying to do???\nYou can only enter 1 to kick, 2 to punch, 3 to block: ");

*fighterInput* = *input*.nextInt();

}

*opponentAction* = *actionArray*[*fighterInput* - 1];

}

}

**private** **static** **void** pickFighter()

{

*printBorder*();

System.***out***.println("Who would like to fight as?\n");

Arrays.*sort*(*playerOneArray*);

// Prints fighter name choices

**for**(String fighters : *playerOneArray*) System.***out***.println(fighters);

**do**

{

System.***out***.print("\nPLAYER ONE - Select a fighter name from the list above: ");

*playerOneName* = *input*.next();

// Compares user's input name to each element in the array

**for** (**int** i = 0; i < *playerOneArray*.length; i++)

{

**if**(*playerOneArray*[i].equals(*playerOneName*)) *player1ValidName* = **true**;

}

} **while** (!*player1ValidName*); // Loops while user input name does not match an array element

System.***out***.println();

//Player two selects fighter from different array

**if** (*isMultiplayer*)

{

*printBorder*();

System.***out***.println("Who would like to fight as?\n");

Arrays.*sort*(*playerTwoArray*);

**for**(String fighters : *playerTwoArray*) System.***out***.println(fighters);

**do**

{

System.***out***.print("\nPLAYER TWO - Select a fighter name from the list above: ");

*opponentName* = *input*.next();

**for** (**int** i = 0; i < *playerTwoArray*.length; i++)

{

**if**(*playerTwoArray*[i].equals(*opponentName*)) *player2ValidName* = **true**;

}

} **while** (!*player2ValidName*); // Loops while user input name does not match an array element

}

**for**(**int** i = 0; i < 50; i++) System.***out***.println();

}

// Select One or Two Players

**private** **static** **void** pickNumOfPlayers()

{

*printBorder*();

System.***out***.print("Enter 1 for one player or 2 for two player: ");

*numOfPlayers* = *input*.nextInt(); // Throws error if input is not an int

**if** (*numOfPlayers* == 2) *isMultiplayer* = **true**;

System.***out***.println();

}

**private** **static** **void** pickCountry()

{

Arrays.*sort*(*countriesArray*);

**do**

{

*printBorder*();

System.***out***.println("Which country would you like to fight in?\n");

**for**(String country : *countriesArray*) System.***out***.println(country);

System.***out***.print("\nEnter a country from the list above: ");

*country* = *input*.next();

// toUpperCase returns a NEW string, doesn't change old string

*countryUpperCase* = *country*.toUpperCase();

**switch** (*countryUpperCase*)

{

**case** "USA":

*opponentName* = "Rocky";

**break**;

**case** "RUSSIA":

*opponentName* = "Drago";

**break**;

**case** "CANADA":

*opponentName* = "Horton";

**break**;

**case** "GERMANY":

*opponentName* = "Heltzenburg";

**break**;

**case** "WAKANDA":

*opponentName* = "Black Panther";

**break**;

}

} **while** (*opponentName*.equals(""));

System.***out***.println();

}

**private** **static** **void** introduceGame()

{

*printBorder*();

System.***out***.println("Welcome to Kick Punch Block!\nThis game is inspired by Rock Paper Scissors.\nAdditional features include health points, fighter names, and rounds.\n");

}

}