



**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

संयुक्त विद्यालय

**Flexi Credit Course – Programming with Java  
AIML 2021-25  
Mini Project Report**

<b>PRN</b>	<b>Name</b>	<b>Branch and Batch</b>
21070126129	Sugam Sharma	<b>AIML -B3</b>
21070126130	Roshan Kumar Yadav	
22070126509	Apurva Patil	

**Topic:-**

**Brick Breaker Game**



## Index

S. no	Topics	Page No.
1	<b>Abstract</b>	<b>3</b>
2	<b>Objective and Aim</b>	<b>4</b>
3	<b>Class Diagram</b>	<b>5</b>
4	<b>Modules and Classes</b>	<b>6</b>
5	<b>Exception handling</b>	<b>7-10</b>
6	<b>Code and Output Snippets</b>	<b>11-40</b>
7	<b>Reference</b>	<b>41-43</b>



## **ABSTRACT**

Brick Breaker is a classic arcade game where the player controls a paddle at the bottom of the screen, bouncing a ball to break a wall of bricks at the top of the screen. The game uses Java language and can be run on any platform that supports Java.

The game starts with a superficial level consisting of a few rows of bricks. The player must move the paddle left or right to keep the ball in play, bouncing it off the paddle and into the bricks to break them. The player loses a life if the ball hits the bottom of the screen. The game ends when the player runs out of lives.

As the player progresses through the levels, the game becomes more challenging with more complex brick patterns, faster ball speeds, and power-ups such as multi-ball and paddle enlargement. The game also includes sound and visual effects to enhance the player's experience.

The game is developed using Java's graphics and event-handling libraries. The graphics library draws the game elements on the screen, while the event handling library detects user input from the keyboard or mouse. The game also includes collision detection to handle the interaction of the ball, paddle, and bricks.

In summary, Brick Breaker is a fun and challenging game made with Java. It uses Java's graphics and event handling libraries to provide an engaging user experience with sound effects and visual effects.



## OUR OBJECTIVES

The objectives of the Brick Breaker game made in Java are:

- **Break all the Bricks** – The game's primary objective is to break all the bricks on the screen using a ball and a paddle. The player must hit the ball against the bricks to break them and clear the level.
- **Score Points** – The player earns points each time the ball hits a brick. The number of points earned varies depending on the type of brick hit. The player can aim to score as many points as possible.
- **Complete Levels** – The game consists of multiple levels with increasing difficulty. The player must complete each level by breaking all the bricks without losing all their lives.
- **Power-Ups** – The game includes power-ups that players can collect by hitting special bricks. Power-ups include larger oversized balls, multiple balls, and slow-motion effects that can help the player clear levels more easily.
- **Avoid Losing Lives** – The player starts with a limited number of lives. Each time the ball falls off the screen and the player fails to hit it with the paddle, they lose a life. The player needs to avoid losing all their lives to keep playing and progress through the level.



## Class Diagram



## Class And Module

1. ThemeMenu – This Java Swing class is called ThemeMenu which extends JPanel. It provides a menu for the user to select a background color for the game and saves the selected color to a file called "BackGroundColor.txt".
2. Sprite – This code defines a class called Sprite, which is used to represent a game sprite - an image that can be moved around on the screen. The class has instance variables to store the sprite's position (x and y), size (imageWidth and imageHeight), and image (image), as well as a Rectangle2D object (rectangle2D) to represent the sprite's bounding box.
3. Racket – This code defines a Racket class that extends the Sprite class. The Racket class represents a player's paddle in a game.
4. MenuScreen – This is a Java Swing application code that creates a menu screen for a game called "Brick Breaker"
5. Item-This is a class named "item" which extends the "Sprite" class. It represents an item that can be dropped in the game.
6. HighScoreBoard – This code defines a Java Swing panel named HighScoreBoard which displays the top 5 scores in a game from a text file named ScoreList.txt
7. GameBoard – This is a code for a Brick Breaker game written in Java using Swing. The code contains a GameBoard class that extends JPanel and implements the game's logic and GUI.
8. BrickBreaker - This code is a Java program that creates a game called "BrickBreaker" using Swing, which is a Java GUI toolkit.
9. Frontend.java – The GUI is created using a drag-and-drop editor in NetBeans IDE. The code has three methods that handle the action events of the buttons: jButton1ActionPerformed, jButton2ActionPerformed, and jButton3ActionPerformed. Each method hides the current JFrame (the frontend) and creates and shows another JFrame (another Java class).
10. Createaccount– This is a Java Swing application for creating user accounts. It allows users to enter their full name, username, and confirm password. When

clicking the “Create” button, the program will attempt to create a new user.

11.Loginpage- This Java Swing application implements a login page for a game called Brick Breaker. The user enters their username and password and clicks the "Login" button to authenticate themselves.

### **Exception Handling**

#### **User Entered Login Details:**

When the user clicks the "Login" button, the code establishes a connection to a MySQL database, retrieves the username and password fields from the login page, and compares them to the entries in the database. If the credentials match, the user is authenticated and allowed to proceed to the game.

The code uses the JDBC API to connect to the MySQL database and execute SQL queries. It retrieves the username and password fields using the `getText()` method of the username and password text fields, respectively. It then creates a `Statement` object to execute the SQL query and a `ResultSet` object to store the result.

#### **Code :-**

```
private void loginButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        String un = username.getText();  
        String pw = password.getText();  
  
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/brickbreaker", "root", "299792458");  
        Statement st = con.createStatement();  
  
        String sql = "SELECT * from users;";  
        ResultSet rs = st.executeQuery(sql);  
        while (rs.next()) {  
            String username = rs.getString("userNName");  
            String password = rs.getString("password");  
  
            if (un.equals(username) && pw.equals(password)) {  
                new BrickBreaker().setVisible(true);  
                return;  
            }  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```



```
        }
    }
    JOptionPane.showMessageDialog(this, "Username or Password is incorrect.");
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Connection Failed!!!");
}
}

private void return_loginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
    new frontend().setVisible(true);
}

/***
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    *
    * For details see
    * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
```



```
        } catch (IllegalAccessException ex) {  
  
java.util.logging.Logger.getLogger(loginpage.class.getName()).log(java.util.logging.Level  
.SEVERE, null, ex);  
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
  
java.util.logging.Logger.getLogger(loginpage.class.getName()).log(java.util.logging.Level  
.SEVERE, null, ex);  
    }  

```

### User Creating Account:

1. ClassNotFoundException: This exception is thrown when a specified class cannot be found. In the given code, this exception is caught if the Look and Feel class "Nimbus" cannot be found.
2. InstantiationException: This exception is thrown when an application tries to create an instance of a class using the newInstance() method in Class, but the specified class object cannot be instantiated because it is an interface or an abstract class, or it has no default constructor. In the given code, this exception is caught if an instance of the Look and Feel class "Nimbus" cannot be created.
3. IllegalAccessException: This exception is thrown when an application tries to create an instance of a class using the newInstance() method in Class, but the current method does not have access to the definition of the specified class, or the class is not accessible because it is private, protected or has no public constructor. In the given code, this exception is caught if an instance of the Look and Feel class "Nimbus" cannot be created due to illegal access.
4. UnsupportedLookAndFeelException: This exception is thrown when the specified look and feel class cannot be loaded or is not supported by the current platform. In the given code, this exception is caught if the Look and Feel class "Nimbus" is not supported by the current platform.

### Code :-

```
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  
    /* If Nimbus (introduced in Java SE 6) is unavailable, stay with the default look and  
    feel.  
     * For details see  
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  
    */
```



```
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frontend.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frontend.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frontend.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(frontend.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
```

## Apart from these exceptions, we have used

1. **IOException** - a checked exception that is thrown when an input/output operation fails or is interrupted. It is a subclass of the java.lang.Exception class.
2. **FileNotFoundException** - is a checked exception in Java that is thrown when an attempt is made to access a file that does not exist or could not be found. This exception is thrown by methods like FileInputStream and FileReader when the accessed file cannot be found.
3. **NumberFormatException**- is a checked exception in Java that is thrown by the Integer.parseInt() and similar methods when they cannot parse a string argument that does not contain a valid integer. This exception occurs when a programmer tries to convert a string to a numeric type like int or double, but the string does not contain a valid representation of the expected numeric value.



## **CODE AND OUTPUT:**

## Frontend.java

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, brick - Apache NetBeans IDE 16, Search (Ctrl + I).
- Toolbar:** Standard NetBeans icons for file operations.
- Project Explorer (Projects tab):** Shows the project structure with files like frontend.java, loginpage.java, createaccount.java, Ball.java, Brick.java, BrickBreaker.java, Configurations.java, Desktop.java, Driver.java, GameBoard.java, and GameBoardTest.java.
- Code Editor:** The main window displays the Java code for the `frontend` class. The code includes imports, package declarations, class definitions, and various methods. A yellow highlight covers the area from line 20 to the end of the class definition.
- Status Bar:** Shows memory usage (1090/1924MB), a progress bar, and the current time (19:11).

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
brick - Apache NetBeans IDE 16
Search (Ctrl + I)
1090/1924MB
1
2     * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3     * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4
5 package brick;
6
7 /**
8 *
9 * @author Roshan
10 */
11 public class frontend extends javax.swing.JFrame {
12
13     /**
14      * Creates new form frontend
15      */
16     public frontend() {
17         initComponents();
18     }
19
20     /**
21      * This method is called from within the constructor to initialize the form.
22      * WARNING: Do NOT modify this code. The content of this method is always
23      * regenerated by the Form Editor.
24      */
25     @SuppressWarnings("unchecked")
26     // Generated Code
27
28     private void jButtonActionPerformed(java.awt.event.ActionEvent evt) {
29         // TODO add your handling code here:
30         this.setVisible(false);
31         new createaccount().setVisible(true);
32     }
33
34     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
35         // TODO add your handling code here:
36         this.setVisible(false);
37     }
38 }
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help  
brick - Apache NetBeans IDE 16  
Search (Ctrl+)  
frontend.java x loginpage.java x createaccount.java x Ball.java x Brickjava x BrickBreakerjava x Configurations.java x Desktop.java x Driver.java x GameBoard.java x GameBoardTest.java...  
Source Design History Tools Help  
121     this.setVisible(false);  
122     new loginpage().setVisible(true);  
123  
124 }  
125  
126     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
127         // TODO add your handling code here:  
128         this.setVisible(false);  
129         new aboutus().setVisible(true);  
130     }  
131  
132     /**  
133      * @param args the command line arguments  
134      */  
135     public static void main(String args[]) {  
136         /* Set the Nimbus look and feel */  
137         /* Look and feel setting code (optional) */  
138  
139         /* Create and display the form */  
140         java.awt.EventQueue.invokeLater(new Runnable() {  
141             public void run() {  
142                 new frontend().setVisible(true);  
143             }  
144         });  
145     }  
146  
147     // Variables declaration - do not modify  
148     private javax.swing.JButton jButton1;  
149     private javax.swing.JButton jButton2;  
150     private javax.swing.JButton jButton3;  
151     private javax.swing.JLabel jLabel1;  
152     private javax.swing.JLabel jLabel2;  
153     // End of variables declaration  
154 }  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

## Loginpage.java

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
brick - Apache NetBeans IDE 16
Search (Ctrl+I)
Projects loginpage.java <default config> createaccount.java BallJava BrickJava BrickBreaker.java Configurations.java Desktop.java Driver.java GameBoard.java GameBoardTest.java HighScoreBoard.java ...
Source Design History
1 /**
2 * Click nbfs://nbshost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 * Click nbfs://nbshost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4 */
5 package brick;
6
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.Statement;
10 import java.sql.ResultSet;
11 import javax.swing.JOptionPane;
12
13 /**
14 * @author Roshan
15 */
16 public class loginpage extends javax.swing.JFrame {
17
18     /**
19      * Creates new form loginpage
20      */
21     public loginpage() {
22         initComponents();
23     }
24
25
26     /**
27      * This method is called from within the constructor to initialize the form.
28      * WARNING: Do NOT modify this code. The content of this method is always
29      * regenerated by the Form Editor.
30      */
31     @SuppressWarnings("unchecked")
32     // Generated Code
33
34     private void passwordActionPerformed(java.awt.event.ActionEvent evt) {
35         // TODO add your handling code here:
36     }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
245
246
247
247
248
248
249
249
250
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
13
```



```
private void return_loginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
    new frontend().setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    // Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new loginpage().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JButton loginButton;
private javax.swing.JPasswordField password;
private javax.swing.JButton return_login;
private javax.swing.JTextField username;
// End of variables declaration
}
```

## Createaccount.java

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
1166/1924MB Search (Ctrl+I) ...
Projects Source Design History <default config> ...
createaccount.java BallJava BrickJava BrickBreakerJava Configurations.java DesktopJava DriverJava GameBoardJava GameBoardTestJava HighScoreBoard.java MenuScreen.java ...
brick - Apache NetBeans IDE 16
package brick;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;
import java.sql.*;

/*
 * Creates new form createaccount
 */
public createaccount() {
    initComponents();
}

/*
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
Generated Code

private void userNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void createAccountActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int userId = 0;
    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        String fn = fullName.getText();
        String un = userName.getText();
    }
}
```



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

brick - Apache NetBeans IDE 16

Search (Ctrl+F)

createaccount.java Ball.java Brick.java BrickBreaker.java Configurations.java Desktop.java Driver.java GameBoard.java GameBoardTest.java HighScoreBoard.java MenuScreen.java

Source Design History

```
158     String fn = fullName.getText();
159     String un = userName.getText();
160     String pw = passWord.getText();
161     String cpw = confirmPassword.getText();
162
163     if (pw.equals(cpw)) {
164
165         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/brickbreaker", "root", "root");
166         Statement st = con.createStatement();
167         try {
168             String sqlUserId = "SELECT userId FROM users ORDER BY userId DESC LIMIT 1";
169             ResultSet rs = st.executeQuery(sqlUserId);
170             String finalUserId = null;
171             while (rs.next()) {
172                 finalUserId = rs.getString("userId");
173             }
174             userId = Integer.parseInt(finalUserId);
175
176             userId += 1;
177         } catch (Exception e) {}
178
179         String sql = "insert into users (userId, fullName, userName, password)"
180             + "values('" + userId + "','" + fn + "','" + un + "','" + pw + "')";
181
182         try {
183             int i = st.executeUpdate(sql);
184             if (i > 0) {
185                 JOptionPane.showMessageDialog(this, "Account Created \n Please login!");
186                 new loginpage().setVisible(true);
187                 this.setVisible(false);
188             } else {
189                 JOptionPane.showMessageDialog(this, "User Already Exist");
190             }
191         } catch (Exception e) {
192             JOptionPane.showMessageDialog(this, "User Already Exist");
193         }
194     } else {
195         JOptionPane.showMessageDialog(this, "Password do not match, try again");
196     }
197 }
```

Output

Google Classroom

25:1 INS Unix (LF)

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

brick - Apache NetBeans IDE 16

Search (Ctrl+F)

createaccount.java Ball.java Brick.java BrickBreaker.java Configurations.java Desktop.java Driver.java GameBoard.java GameBoardTest.java HighScoreBoard.java MenuScreen.java

Source Design History

```
197     ) catch (Exception e) {
198         JOptionPane.showMessageDialog(this, "Error while establishing connection failed!!!");
199     }
200 }
201
202 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
203     // TODO add your handling code here:
204     evt.getSource().setVisible(false);
205     new frontend().setVisible(true);
206 }
207
208 /**
209 * @param args
210 */
211 public static void main(String args[]) {
212     /* Set the Nimbus look and feel */
213     /* Look and feel setting code (optional) */
214
215     /* Create and display the form */
216     java.awt.EventQueue.invokeLater(new Runnable() {
217         public void run() {
218             new createaccount().setVisible(true);
219         }
220     });
221 }
222
223 // Variables declaration - do not modify
224 private javax.swing.JPasswordField confirmPassword;
225 private javax.swing.JButton createAccount;
226 private javax.swing.JTextField fullName;
227 private javax.swing.JButton jButton1;
228 private javax.swing.JLabel jLabel1;
229 private javax.swing.JLabel jLabel2;
230 private javax.swing.JLabel jLabel3;
231 private javax.swing.JLabel jLabel4;
232 private javax.swing.JLabel jLabel5;
233 private javax.swing.JPasswordField passWord;
234 private javax.swing.JTextField userName;
235
236 // End of variables declaration
237 }
```

Output

25:1 INS Unix (LF)



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

## Ball.java :

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
1428/1924MB Search (Ctrl+I)
Projects Source History <default config> 1428/1924MB Search (Ctrl+I)
Ball.java X BrickJava X BrickBreaker.java X Configurations.java X Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java...
1 ...4 lines
2 package brick;
3 import javax.imageio.ImageIO;
4 import javax.swing.ImageIcon;
5 import java.io.IOException;
6 public class Ball extends Sprite {
7     private double xdir;
8     private double ydir;
9     public Ball() throws IOException {
10         initBall();
11     }
12     private void initBall() throws IOException {
13         xdir = 1;
14         ydir = -1;
15     }
16     loadImage();
17     getImageDimensions();
18     resetState();
19 }
20     private void loadImage() throws IOException {
21         var ii = new ImageIcon(ImageIO.read(Ball.class.getResource("images\\ball.png")));
22         image = ii.getImage();
23     }
24     void move() {
25         x += xdir;
26         y += ydir;
27         if(x + 2 >= (Configurations.WIDTH)) {
28             xdir *= -1;
29             setXDir(xdir);
30         }
31         else if(x <= 0){
32             xdir *= -1;
33             x = xdir;
34         }
35         else if(y <= 0){
36             ydir *= -1;
37         }
38     }
39     private void resetState() {
40         x = Configurations.INIT_BALL_X;
41         y = Configurations.INIT_BALL_Y;
42     }
43     void setXDir(double x) {
44         xdir = x;
45     }
46     void setYDir(double y) {
47         ydir = y;
48     }
49     double getYDir() {
50         return ydir;
51     }
52 }
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
6796/1924MB Search (Ctrl+I)
Projects Source History <default config> 6796/1924MB Search (Ctrl+I)
Ball.java X BrickJava X BrickBreaker.java X Configurations.java X Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java...
28     void move() {
29         x += xdir;
30         y += ydir;
31         if(x + 2 >= (Configurations.WIDTH)) {
32             xdir *= -1;
33             setXDir(xdir);
34         }
35         else if(x <= 0){
36             xdir *= -1;
37             x = xdir;
38         }
39         else if(y <= 0){
40             ydir *= -1;
41             y = ydir;
42         }
43     }
44     private void resetState() {
45         x = Configurations.INIT_BALL_X;
46         y = Configurations.INIT_BALL_Y;
47     }
48     void setXDir(double x) {
49         xdir = x;
50     }
51     void setYDir(double y) {
52         ydir = y;
53     }
54     double getYDir() {
55         return ydir;
56     }
57 }
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help  
brick - Apache NetBeans IDE 16  
Search (Ctrl+F)  
Projects  
Source History  1216/1924MB  
BrickJava X BrickBreaker.java X Configurations.java X Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java X Sprite.java...  
1   ...4 lines  
2 package brick;  
3 import javax.imageio.ImageIO;  
4 import javax.swing.ImageIcon;  
5 import java.io.IOException;  
6 public class Brick extends Sprite  
7 {  
8     private boolean destroyed;  
9     private int health;  
10    private boolean cement;  
11    private boolean containsItem;  
12    public Brick(int x, int y) throws IOException {  
13        initBrick(x, y);  
14    }  
15    private void initBrick(int x, int y) throws IOException {  
16        this.x = x;  
17        this.y = y;  
18        destroyed = false;  
19        cement = false;  
20        health = 1;  
21        //boolean for item drop bricks  
22        containsItem = false;  
23        loadImage(0);  
24        getImageDimensions();  
25        int random = (int) (Math.random() * 100) + 1;  
26        if (random > 50 && random <= 80) {  
27            health += 50;  
28        } else if (random > 80 && random <= 95) {  
29            health += 100;  
30        } else if (random > 95) {  
31            cement = true;  
32            loadImage(3);  
33            getImageDimensions();  
34        } else if (random < 10) {  
35            containsItem = true;  
36            loadImage(4);  
37        }  
38    }  
39 }  
40 }  
41 }  
42 }
```

The screenshot shows the Apache NetBeans IDE interface with the title bar "brick - Apache NetBeans IDE 16". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, Cut, Copy, Paste, Find, and Run. The left sidebar has a Projects tab and a Source tab (which is selected). The main editor area displays the following Java code:

```
private void loadImage(int index) throws IOException {
    if (index == 0) {
        var ii = new ImageIcon(ImageIO.read(Brick.class.getResource("images\\brick.png")));
        image = ii.getImage();
    } else if (index == 1) {
        var ii = new ImageIcon(ImageIO.read(Brick.class.getResource("images\\brick_cracked_1.png")));
        image = ii.getImage();
    } else if (index == 2) {
        var ii = new ImageIcon(ImageIO.read(Brick.class.getResource("images\\brick_cracked_2.png")));
        image = ii.getImage();
    } else if (index == 3) {
        var ii = new ImageIcon(ImageIO.read(Brick.class.getResource("images\\cement.png")));
        image = ii.getImage();
    } else if (index == 4) {
        var ii = new ImageIcon(ImageIO.read(Brick.class.getResource("images\\itemBrick.png")));
        image = ii.getImage();
    } else {
        System.out.println("Bad index passed to Brick loadImage");
    }
}

// 50% chance health = 0 (dies in one hit)
// 30% chance health = 50 (dies in two hits)
// 15% chance health = 100 (dies in three hits)
// 5% chance cement (invincible)
private int getHealth() {
    return health;
}

private void setHealth() {
    health -= 50;
}

boolean isCement() {
    return cement;
}

void doDamage() throws IOException {
    if (!isCement()) {
        setHealth();
        if (getHealth() <= 0) {
            destroyed = true;
        }
    }
}
```



**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** brick - Apache NetBeans IDE 16
- Toolbar:** Includes icons for file operations, search, and project navigation.
- Project Explorer:** Shows the current project structure with files like BrickJava, BrickBreakerJava, Configurations.java, Desktop.java, Driver.java, GameBoard.java, GameBoardTest.java, HighScoreBoard.java, MenuScreen.java, Racket.java, and Sprite.java.
- Source Editor:** Displays the Java code for the BrickBreaker class. The code handles health management, damage logic, and item detection. A cursor is visible at line 72, column 11, where the character '}' is located after the closing brace of the doDamage() method.
- Status Bar:** Shows memory usage (7704/1924MB), a search field, and system information (72:22 INS Windows (CRLF)).

```
// 5% chance cement (invincible)
private int getHealth() {
    return health;
}

private void setHealth() {
    health -= 50;
}

boolean isCement() {
    return cement;
}

void doDamage() throws IOException {
    if (!isCement()) {
        setHealth();
        if (getHealth() <= 0) {
            destroyed = true;
        } else if (getHealth() == 1) {
            loadImage(2);
            getImageDimensions();
        } else if (getHealth() == 51) {
            loadImage(1);
            getImageDimensions();
        }
    }
}

boolean isDestroyed() {
    return destroyed;
}

boolean hasItem() {
    return containsItem;
}
```

# **BrickBreaker.java**

The screenshot shows the Apache NetBeans IDE interface. The title bar reads "brick - Apache NetBeans IDE 16". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has various icons for file operations like Open, Save, Cut, Copy, Paste, Find, and Run. The Projects tab is selected, showing a tree view of the project structure with "BrickBreaker.java" as the current file. The Source tab is active, displaying the Java code for "BrickBreaker.java". The code defines a class "BrickBreaker" that extends "JFrame". It contains methods for initializing the UI and starting the game. A "main" method is also present, which creates a new "BrickBreaker" instance and makes it visible. The code uses Java Swing components and handles IOExceptions. The status bar at the bottom shows memory usage "5245/1924MB" and the current file "INS Windows (C:\P...)".

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help  
brick - Apache NetBeans IDE 16  
Search (Ctrl+F)  
Projects  
Source History  
BrickBreaker.java Configurations.java Desktop.java Driver.java GameBoard.java GameBoardTest.java HighScoreBoard.java MenuScreen.java Racket.java Sprite.java ThemeMen...  
<default config> 5245/1924MB  
1 ... 4 lines  
2 package brick;  
3  
4 import javax.swing.*;  
5 import java.awt.EventQueue;  
6 import java.io.IOException;  
7 import static javax.swing.WindowConstants.EXIT_ON_CLOSE;  
8  
9 /*... 4 lines */  
10 public class BrickBreaker extends JFrame {  
11     public Brickbreaker() throws IOException {  
12         initUI();  
13     }  
14     private void initUI() throws IOException {  
15         add(new MenuScreen());  
16         setTitle("Breakout");  
17         setDefaultCloseOperation(EXIT_ON_CLOSE);  
18         setLocationRelativeTo(null);  
19         setResizable(false);  
20         pack();  
21     }  
22     public static void main(String[] args) {  
23         EventQueue.invokeLater(() -> {  
24             BrickBreaker game = null;  
25             try {  
26                 game = new BrickBreaker();  
27             } catch (IOException e) {  
28                 e.printStackTrace();  
29             }  
30             game.setVisible(true);  
31         });  
32     }  
33 }  
34  
35  
36  
37  
38 }
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

## Configuration.java

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
1079/1440MB  Search (Ctrl+I)  -  X
Projects Source History 1079/1440MB  Search (Ctrl+I)  -  X
Configurations.java X Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java X Sprite.java X ThemeMenu.java X item.java X
1 ... 4 lines
2 package brick;
3
4 public interface Configurations {
5     int WIDTH = 300;
6     int HEIGHT = 400;
7     int TOP_EDGE = 10;
8     int BOTTOM_EDGE = 390;
9     int N_OF_BRICKS = 30;
10    int INIT_PADDLE_X = 200;
11    int INIT_PADDLE_Y = 360;
12    int INIT_BALL_X = 230;
13    int INIT_BALL_Y = 355;
14    int INIT_ITEM_X = 200;
15    int INIT_ITEM_Y = 360;
16    int PERIOD = 10;
17 }
18
19
20
21
22
```

## Item.java

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
452.2/1440MB  Search (Ctrl+I)  -  X
Projects Source History 452.2/1440MB  Search (Ctrl+I)  -  X
Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java X Sprite.java X ThemeMenu.java X item.java X
1 ... 4 lines
2 package brick;
3
4 import javax.imageio.ImageIO;
5 import javax.swing.ImageIcon;
6 import java.io.IOException;
7
8 public class item extends Sprite {
9     private double xdir = 1;
10    private double ydir = 1;
11    private int item_speed = 1;
12    public item(double x, double y) throws IOException {
13        initItem(x, y);
14    }
15    private void initItem(double x, double y) throws IOException {
16        this.x = x + 20;
17        this.y = y + 10;
18        loadImage();
19        getImageDimensions();
20    }
21    private void loadImage() throws IOException {
22        var ii = new ImageIcon(ImageIO.read(Ball.class.getResource("/images/itemDrop.png")));
23        image = ii.getImage();
24    }
25    void move() {
26        y += ydir;
27    }
28    void setXDir(double x) {
29        xdir = x;
30    }
31    void setYDir(double y) {
32        ydir = y;
33    }
34    double getYDir() {
35        return ydir;
36    }
37    private void resetState() {
38        x = Configurations.INIT_ITEM_X;
39        y = Configurations.INIT_ITEM_Y;
40    }
41 }
42
```



**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

## ThemeMenu.java

The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard Java development tools like New Project, Open, Save, Cut, Copy, Paste, Find, Replace, etc.
- Project Explorer (Projects tab):** Shows the current project structure with files like Desktop.java, Driver.java, GameBoard.java, GameBoardTestJava.java, HighScoreBoard.java, MenuScreenJava.java, RacketJava.java, Sprite.java, and ThemeMenu.java.
- Code Editor:** The active editor window displays the Java code for `ThemeMenu.java`. The code implements a `JPanel` with two buttons: `rtnMenuButton` and `button`. It handles button actions to show a color dialog and set the background color of the button. It also adds a `returnHandler` to the `rtnMenuButton`.
- Status Bar:** Shows memory usage (884.3 / 1440 MB) and the current time (7:34).
- Bottom Status Bar:** Shows the current file as `ThemeMenu.java` and the Windows operating system indicator.

```
1  ... 4 lines
2
3  package brick;
4
5  import javax.swing.*;
6
7  import java.awt.*;
8
9  import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import java.io.*;
12
13 public class ThemeMenu extends JPanel {
14
15     JButton rtnMenuButton = new JButton("Return");
16     JButton button = new JButton("Choose a color");
17
18     public ThemeMenu() throws IOException {
19         initMenu();
20     }
21
22     private void initMenu() throws IOException {
23
24         setFocusable(true);
25         setPreferredSize(new Dimension(Configurations.WIDTH, Configurations.HEIGHT));
26         JPanel colorPanel = new JPanel();
27         setLayout(null);
28         rtnMenuButton.setBounds(10, 10, 75, 20);
29         add(rtnMenuButton);
30         add(colorPanel, BorderLayout.CENTER);
31         button.setBounds(0, 100, Configurations.WIDTH, 50);
32         add(button);
33         setSize(425, 150);
34         setVisible(true);
35         button.addActionListener(event -> {
36             color = JColorChooser.showDialog(null, "Pick your color", color);
37             if (color == null) {
38                 color = (Color.WHITE);
39             }
40             EditBackgroundfile(color.toString());
41             button.setBackground(color);
42         });
43         add(colorPanel, BorderLayout.CENTER);
44         add(button, BorderLayout.SOUTH);
45         setSize(425, 150);
46         returnHandler rtmHandler = new returnHandler();
47         rtnMenuButton.addActionListener(rtmHandler);
48     }
49
50 }
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** brick - Apache NetBeans IDE 16
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer:** Shows files like Desktop.java, Driver.java, GameBoard.java, GameBoardTest.java, HighScoreBoard.java, MenuScreen.java, Racket.java, Sprite.java, and ThemeMenu.java.
- Code Editor:** Displays Java code for the `ThemeMenu.java` file. The code includes methods for setting up a menu screen, writing to a file for background color, and drawing text on a 2D graphics context.
- Status Bar:** Shows memory usage (469/1108MB) and other system information.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
brick - Apache NetBeans IDE 16
Search (Ctrl+F)
Desktop.java X Driver.java X GameBoard.java X GameBoardTest.java X HighScoreBoard.java X MenuScreen.java X Racket.java X Sprite.java X ThemeMenu.java X
Projects Source History
<default config> 469/1108MB
45     @Override
46     public void actionPerformed(ActionEvent e) {
47         setLayout(new BorderLayout());
48         validate();
49         repaint();
50         MenuScreen menu = new MenuScreen();
51         add(menu);
52         menu.requestFocusInWindow();
53         remove(btnMenuBar);
54         remove(button);
55     }
56 }
57 private void EditBackgroundFile(String colorChosen) {
58     FileWriter fw;
59     try {
60         fw = new FileWriter("BackGroundColor.txt");
61         BufferedWriter bw = new BufferedWriter(fw);
62         bw.write(colorChosen);
63         bw.close();
64         fw.close();
65     } catch (IOException ex) {
66         ex.printStackTrace();
67     }
68 }
69 @Override
70 public void paintComponent(Graphics g){
71     super.paintComponent(g);
72     var g2d = (Graphics2D) g;
73     try {
74         drawText(g2d);
75     } catch (IOException e) {
76         e.printStackTrace();
77     }
78 }
79
80 private void drawText(Graphics2D g2d) throws IOException {
81     g2d.drawString("Choose background color below:", 70, 50);
82     g2d.drawString("Current Background Color", 70, 80);
83 }
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

## Sprite.java

```
public class Sprite {
    // int x;
    // int y;
    double x;
    double y;
    int imageWidth;
    int imageHeight;
    Image image;
    Rectangle2D rectangle2D = new Rectangle2D.Double();
    protected void setX(double x) {
        this.x = x;
    }
    double getX() {
        return x;
    }
    protected void setY(double y) {
        this.y = y;
    }
    double getY() {
        return y;
    }
    int getImageWidth() {
        return imageWidth;
    }
    int getImageHeight() {
        return imageHeight;
    }
    Image getImage() {
        return image;
    }
    Rectangle2D getRect() {
        rectangle2D.setRect(x, y, image.getWidth(null), image.getHeight(null));
        return rectangle2D;
    }
    void getImageDimensions() {
        imageWidth = image.getWidth(null);
        imageHeight = image.getHeight(null);
    }
}
```

## Racket.java

```
package brick;
import java.awt.event.KeyEvent;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
public class Racket extends Sprite {
    private int dx;
    public Racket(int racket) throws IOException {
        initRacket(racket);
    }
    private void initRacket(int racket) throws IOException {
        loadImage(racket);
        getImageDimensions();
        resetState();
    }
    private void loadImage(int racket) throws IOException {
        ImageIcon ii;
        if (racket == 1) {
            ii = new ImageIcon(ImageIO.read(Racket.class.getResource("images\\longPaddle.png")));
        } else if (racket == 2) {
            ii = new ImageIcon(ImageIO.read(Racket.class.getResource("images\\shortPaddle.png")));
        } else {
            ii = new ImageIcon(ImageIO.read(Racket.class.getResource("images\\paddle.png")));
        }
        image = ii.getImage();
    }
    void move() {
        x += dx;
        if (x <= 0) {
            x = 0;
        }
        if (x >= Configurations.WIDTH - imageWidth) {
            x = Configurations.WIDTH - imageWidth;
        }
    }
    void keyPressed(KeyEvent e, int select) {
        int key = e.getKeyCode();
        if (select == 0) {
            if (key == KeyEvent.VK_LEFT) {

```



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

brick - Apache NetBeans IDE 16

Search (Ctrl+I)

Desktop.java X DriverJava X GameBoard.java X GameBoardTestJava X HighScoreBoard.java X MenuScreen.java X RacketJava X

Source History

```
37     x = Configurations.WIDTH - imageWidth;
38 }
39 void keyPressed(KeyEvent e, int select) {
40     int key = e.getKeyCode();
41     if (select == 0){
42         if (key == KeyEvent.VK_LEFT) {
43             dx = -2;
44         }
45         if (key == KeyEvent.VK_RIGHT) {
46             dx = 2;
47         }
48     }else {
49         if (key == KeyEvent.VK_A) {
50             dx = -2;
51         }
52         if (key == KeyEvent.VK_D) {
53             dx = 2;
54         }
55     }
56 void keyReleased(KeyEvent e, int select) {
57     int key = e.getKeyCode();
58     if(select == 0) {
59         if (key == KeyEvent.VK_LEFT) {
60             dx = 0;
61         }
62         if (key == KeyEvent.VK_RIGHT) {
63             dx = 0;
64         }
65     }else{
66         if (key == KeyEvent.VK_A) {
67             dx = 0;
68         }
69         if (key == KeyEvent.VK_D) {
70             dx = 0;
71     }
72     private void resetState() {
73         x = Configurations.INIT_PADDLE_X;
74         y = Configurations.INIT_PADDLE_Y;
75     }
76 }
```

Output

## MenuScreen.java

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

brick - Apache NetBeans IDE 16

Search (Ctrl+I)

Desktop.java X DriverJava X GameBoard.java X GameBoardTestJava X HighScoreBoard.java X MenuScreen.java X

Source History

```
1 ...4 lines
2 package brick;
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.io.FileNotFoundException;
8 import java.io.IOException;
9 /**
10 */
11 public class MenuScreen extends JPanel {
12     JButton StartButton = new JButton("Start Game");
13     JButton HSButton = new JButton("High Scores");
14     JButton ThemeButton = new JButton("Change Theme");
15     public MenuScreen(){
16         initMenu();
17     }
18     private void initMenu() {
19         setFocusable(true);
20         setPreferredSize(new Dimension(Configurations.WIDTH, Configurations.HEIGHT));
21         setLayout(null);
22         StartButton.setBounds((Configurations.WIDTH - 120) / 2, 100, 120, 40);
23         HSButton.setBounds((Configurations.WIDTH - 120) / 2, 150, 120, 40);
24         ThemeButton.setBounds((Configurations.WIDTH - 120) / 2, 200, 120, 40);
25         add(StartButton);
26         add(HSButton);
27         add(ThemeButton);
28         StartHandler stHandler = new StartHandler();
29         StartButton.addActionListener(stHandler);
30         HSHandler hsHandler = new HSHandler();
31         HSButton.addActionListener(hsHandler);
32         ThemeHandler tHandler = new ThemeHandler();
33         ThemeButton.addActionListener(tHandler);
34     }
35     //Click listener for Start Button
36     private class StartHandler implements ActionListener {
37         @Override
38         public void actionPerformed(ActionEvent e) {
39             try {
40                 setLayout(new FlowLayout());
41             }
42         }
43     }
44 }
```

Output



**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

The screenshot shows an IDE interface with multiple tabs open. The tabs include 'GameBoard.java', 'GameBoardTestJava', 'HighScoreBoard.java', 'MenuScreen.java', and '<default config>'. The code editor displays Java code for a game application, specifically for handling button actions. It includes three private classes: HSHandler, HighScoreBoard, and ThemeHandler, each implementing ActionListener. The code uses try-catch blocks to handle exceptions like IOException and FileNotFoundException.

```
public void actionPerformed(ActionEvent e) {
    try {
        setLayout(new FlowLayout());
        validate();
        repaint();
        GameBoard gameBoard = new GameBoard();
        add(gameBoard);
        gameBoard.requestFocusInWindow();
        remove(HSButton);
        remove(StartButton);
        remove(ThemeButton);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
//Click Listener for High score button
private class HSHandler implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            setLayout(new BorderLayout());
            validate();
            repaint();
            HighScoreBoard hsBoard = new HighScoreBoard();
            add(hsBoard);
            hsBoard.requestFocusInWindow();
            remove(HSButton);
            remove(StartButton);
            remove(ThemeButton);
        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        }
    }
}
//Click Listener for High score button
private class HighScoreBoard implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            setLayout(new BorderLayout());
            validate();
        }
    }
}
//Click Listener for High score button
private class ThemeHandler implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            setLayout(new BorderLayout());
            validate();
        }
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, brick ...
- Toolbars:** Standard toolbar with icons for file operations.
- Project Bar:** Shows the current project structure with GameBoard.java, GameBoardTest.java, HighScoreBoard.java, and MenuScreen.java.
- Code Editor:** The main window displays the `MenuScreen.java` source code. The code implements a `BorderLayout` and contains methods for painting components and drawing titles. It uses `Font.BOLD`, `Color.GRAY`, and `Color.DARK_GRAY` for styling.
- Status Bar:** Shows the time as 80:30 and the system as INS Windows (CRLF).

```
79         setLayout(new BorderLayout());
80         revalidate();
81         repaint();
82         ThemeMenu tMenu = new ThemeMenu();
83         add(tMenu);
84         tMenu.requestFocusInWindow();
85         remove(HButton);
86         remove(StartButton);
87         remove(ThemeButton);
88     } catch (IOException ex) {
89         ex.printStackTrace();
90     }
91 }
92 @Override
93 public void paintComponent(Graphics g) {
94     super.paintComponent(g);
95     var g2d = (Graphics2D) g;
96     drawTitle(g2d);
97 }
98 private void drawTitle(Graphics2D g2d) {
99     var font = new Font("Verdana", Font.BOLD, 30);
100    FontMetrics fontMetrics = this.getFontMetrics(font);
101    g2d.setColor(Color.GRAY);
102    g2d.setFont(font);
103    g2d.drawString("Brick Breaker!",
104        (Configurations.WIDTH - fontMetrics.stringWidth("Brick Breaker!")) / 2,
105        50);
106    g2d.setcolor(Color.DARK_GRAY);
107    g2d.setFont(font);
108    g2d.drawString("Brick Breaker!",
109        ((Configurations.WIDTH - fontMetrics.stringWidth("Brick Breaker!")) / 2) +
110        52);
111    g2d.setcolor(Color.BLACK);
112    g2d.setFont(font);
113    g2d.drawString("Brick Breaker!",
114        ((Configurations.WIDTH - fontMetrics.stringWidth("Brick Breaker!")) / 2) +
115        54));}
```



## HighScoreBoard.java

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and various build and run configurations.
- Project Explorer:** Shows a single project named "brick".
- Toolbars:** Source, History, and several other toolbars with icons for navigation and search.
- Code Editor:** The main area displays the Java code for `HighScoreBoard.java`. The code implements a JPanel with a JButton and handles its action events.
- Code Snippet:** A yellow highlight covers the implementation of the `actionPerformed` method.
- Status Bar:** Shows the number of errors (3), the current time (34:53), and the file encoding (INS Windows (CRLF)).

```
1 ... 4 lines
5 package brick;
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import java.io.BufferedReader;
10 import java.io.FileNotFoundException;
11 import java.io.FileReader;
12 import java.io.IOException;
13 import java.util.ArrayList;
14 import java.util.Arrays;
15 import java.util.Collections;
16 import javax.swing.*;
17 public class HighScoreBoard extends JPanel{
18     JButton rtnMenuButton = new JButton("Return");
19     public HighScoreBoard() throws FileNotFoundException {
20         initBoard();
21     }
22     private void initBoard() {
23         setFocusable(true);
24         setPreferredSize(new Dimension(Configurations.WIDTH, Configurations.HEIGHT));
25         setLayout(null);
26         rtnMenuButton.setBounds(10, 10, 75, 20);
27         add(rtnMenuButton);
28         returnHandler rtnHandler = new returnHandler();
29         rtnMenuButton.addActionListener(rtnHandler);
30     }
31     //Click Listener for High score button
32     private class returnHandler implements ActionListener {
33         @Override
34         public void actionPerformed(ActionEvent e) {
35             setLayout(new BorderLayout());
36             revalidate();
37             repaint();
38             MenuScreen menu = new MenuScreen();
39             add(menu);
40             menu.requestFocusInWindow();
41             remove(rtnMenuButton);
        }
    }
}
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, brick ...
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, Undo, Redo, and various tool-specific icons.
- Project Explorer:** Shows a single project named "...va".
- Source Editor:** The active tab is "Source", displaying the code for **HighScoreBoard.java**. The code implements a component that displays top scores from a file. It uses `Graphics` and `Graphics2D` for drawing, and `FileReader` and `BufferedReader` for reading a file named `ScoreList.txt`.
- Code Snippet:** The code includes methods for drawing scores and handling exceptions. It uses `FontMetrics` to calculate string widths and `Integer.parseInt` to parse lines from the file.
- Status Bar:** Shows the number of errors (3), the current line (427), and the file type (INS Windows (CRLF)).

```
41     remove (rtnMenuBar);
42 }
43
44 @Override
45 public void paintComponent(Graphics g){
46     super.paintComponent(g);
47     var g2d = (Graphics2D) g;
48     try {
49         drawScores(g2d);
50     } catch (IOException e) {
51         e.printStackTrace();
52     }
53 }
54 private void drawScores(Graphics2D g2d) throws IOException {
55     FileReader in = new FileReader("ScoreList.txt");
56     BufferedReader br = new BufferedReader(in);
57     var font = new Font("Verdana", Font.BOLD, 18);
58     FontMetrics fontMetrics = this.getFontMetrics(font);
59     g2d.setColor(Color.RED);
60     g2d.setFont(font);
61     g2d.drawString("Top 5 scores",
62             (Configurations.WIDTH - fontMetrics.stringWidth("Top 5 scores")) / 2,
63             50);
64     ArrayList<Integer> scores = new ArrayList<>();
65     String line = br.readLine();
66     int y = 100;
67     if(line == null){
68         var font3 = new Font("Verdana", Font.BOLD, 15);
69         g2d.setColor(Color.BLACK);
70         g2d.setFont(font3);
71         g2d.drawString("No Scores yet!", 40, y);
72     }
73     else {
74         int nScores = 0;
75         while (line != null) {
76             scores.add(Integer.parseInt(line));
77             nScores++;
78             line = br.readLine();
79         }
80     }
81 }
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window brick ... - X
...va Desktop.java × GameBoard.java × GameBoardTest.java × HighScoreBoard.java × < default config > 
Source History Projects
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
int nScores = 0;
while (line != null) {
    scores.add(Integer.parseInt(line));
    nScores++;
    line = br.readLine();

    // If there is less than 5 scores in the score list
    if(nScores == 1){
        scores.add(0);
        scores.add(0);
        scores.add(0);
        scores.add(0);
    }
    else if(nScores == 2){
        scores.add(0);
        scores.add(0);
        scores.add(0);
    }
    else if(nScores == 3){
        scores.add(0);
        scores.add(0);
    }
    else if(nScores == 4){
        scores.add(0);
    }
    scores.sort(Collections.reverseOrder());
    var font2 = new Font("Verdana", Font.BOLD, 15);
    g2d.setColor(Color.BLACK);
    g2d.setFont(font2);
    for (int i = 0; i < 5; i++) {
        g2d.drawString("#" + (i + 1) + ": " + scores.get(i), 40, y);
        y += 30;
    }
    br.close();
    in.close();
}
}
```

Output 3 | 78:14 | INS Windows (CRLF)



## GameBoardTest.java

The screenshot shows a Java test class named GameBoardTest.java. The code uses JUnit annotations (@Before, @Test) and assertions to test the GameBoard class. It includes methods for setting up the game board, testing the initial score (zero), testing score increments after a brick is destroyed, testing score reset after a restart, and testing the initial speed (one). The IDE interface includes a toolbar, a menu bar, and a status bar at the bottom.

```
1  ... 4 lines
5  package brick;
6  import org.junit.Before;
7  import org.junit.Test;
8  import java.io.IOException;
9  import static org.junit.Assert.*;
10 import static org.junit.jupiter.api.Assertions.*;
11 public class GameBoardTest {
12     GameBoard gb;
13     @Before
14     public void setup() throws IOException {
15         gb = new GameBoard();
16     }
17     @Test
18     public void scoreZeroAtStart() {
19         int score;
20         score = gb.score;
21         assertEquals("Score should be zero", 0, score);
22     }
23     @Test
24     public void scoreIncrementsAfterBrickDestroyed() throws IOException {
25         int score;
26         gb.bricks[1].doDamage();
27         gb.checkCollision();
28         score = gb.score;
29         assertEquals("Score should be 1 after 1 brick destroyed", 1, score);
30     }
31     @Test
32     public void scoreResetsWhenRestarted() throws IOException {
33         //Add score
34         gb.score = 4;
35         //restart is clicked
36         gb.restartButton.doClick();
37         assertEquals("Score should be reset to 0 after restarted", 0, gb.score);
38     }
39     @Test
40     public void speedIsOneAtStart() {
41         assertEquals(1, gb.speed, "Should be 1");
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, New, Copy, Paste, Find, and Run.
- Project Explorer:** Shows a single project with files: Driver.java, Desktop.java, GameBoard.java, and GameBoardTest.java (selected).
- Code Editor:** Displays the content of GameBoardTest.java. The code consists of several @Test annotated methods for testing the GameBoard class. The first method, speedIsOneAtStart(), has its assertion highlighted in yellow.
- Bottom Status Bar:** Shows tabs for Output, a message icon with '2', the time 41:50, and the path INS\Windows (CRLF).

```
public void speedIsOneAtStart() {
    assertEquals(1, gb.speed, "Should be 1");
}

@Test
public void speedIsOnePointTwoAfterFiveBricksDestroyed() throws IOException {
    gb.score = 5;
    gb.checkCollision();
    assertEquals(1.2, gb.speed, "Should be 1.5");
}

@Test
public void speedIsDoubleAfterFifteenBricksDestroyed() throws IOException {
    gb.score = 15;
    gb.checkCollision();
    assertEquals(2, gb.speed, "Should be 2");
}

@Test
public void speedIsResetOnRestart() throws IOException {
    //speed set to 2
    gb.speed = 2;
    //click restart
    gb.restartButton.doClick();
    assertEquals(1, gb.speed, "Should be 1");
}

@Test
public void switchToASWDKeyboard() throws IOException{
    gb.aswdButton.doClick();
    assertEquals(1,gb.keySelect,"Value should be 1 and switch into ASWD Keyboard");
}

@Test
public void switchFromASWDKeyboardToArrowKeyboard() throws IOException{
    gb.aswdButton.doClick();
    gb.arrowButton.doClick();
    assertEquals(0,gb.keySelect,"value should be 0 and switch to arrow keyboard.");
}
```



## GameBoard.java

The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like New, Open, Save, Print, and various build and run actions.
- Project Explorer:** Shows a single project named "brick".
- Toolbars:** Includes Source, History, and several other toolbars with various icons.
- Code Editor:** The main window displays the `GameBoard.java` file. The code is a Java class definition for a game board, including imports for Java Swing, AWT, and IO, and a class definition for `GameBoard` extending `JPanel`.
- Status Bar:** Shows the line count (234:37), current time (234:37), and mode (INS).

```
1 ... 4 lines
5 package brick;
6 import javax.imageio.ImageIO;
7 import javax.swing.*;
8 import java.awt.*;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import java.awt.event.KeyAdapter;
12 import java.awt.event.KeyEvent;
13 import java.io.IOException;
14 import java.io.*;
15 import java.util.Objects;
16 import java.util.Scanner;
17 public class GameBoard extends JPanel {
18     private Timer timer; [Minimize]
19     private String message = "Game Over!";
20     private Ball ball;
21     public Racket racket;
22     public Brick[] bricks;
23     private item drop;
24     private boolean itemDrop;
25     public int racketType = 0;
26     private boolean inGame = true;
27     int score = 0;
28     double speed = 1;
29     String speedLevel = "x1";
30     JButton pauseButton = new JButton("Pause");
31     JButton resumeButton = new JButton("Resume");
32     JButton restartButton = new JButton("Restart");
33     boolean restartClicked = false;
34     JButton arrowButton = new JButton("Arrow");
35     JButton aswdButton = new JButton("ASWD");
36     int keySelect = 0;
37     public int livesLeft;
38     public GameBoard() throws IOException {
39         initBoard();
40     }
41     private void initBoard() throws IOException {
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows a Java IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search bar labeled <default config>.
- Project Explorer:** Shows a single project named "GameBoard.java".
- Code Editor:** Displays the source code for GameBoard.java. The code reads background color settings from a file named "BackGroundColor.txt".

```
40 }
41     private void initBoard() throws IOException {
42         PauseHandler settingHandler = new PauseHandler();
43         ResumeHandler resumeHandler = new ResumeHandler();
44         RestartHandler restartHandler = new RestartHandler();
45         ArrowKeyHandler arrowKeyHandler = new ArrowKeyHandler();
46         ASWDKeyHandler aswdKeyHandler = new ASWDKeyHandler();
47         //Read from BackGroundColor.txt to get background color
48         FileReader fr = new FileReader("BackGroundColor.txt");
49         BufferedReader br = new BufferedReader(fr);
50         String color = br.readLine();
51         // Read Color object String and convert to Color object
52         final Scanner scan = new Scanner(color);
53         scan.useDelimiter("(r|\\,|g|\\,|b)=|\\]").next(); //Use proper delimiter and ignore if
54         final int r, g, b;
55         //Verify RGB Values
56         System.out.println(r = scan.nextInt());
57         System.out.println(g = scan.nextInt());
58         System.out.println(b = scan.nextInt());
59         Color c = new Color(r, g, b);
60         setBackground(c);
61         System.out.println(color);
62         fr.close();
63         br.close();
64         JPanel buttonPane = new JPanel();
65         buttonPane.setLayout(new GridLayout(0, 3));
66         buttonPane.setPreferredSize(new Dimension(250, 30));
67         JPanel blank = new JPanel();
68         blank.setVisible(false);
69         buttonPane.add(pauseButton);
70         buttonPane.add(restartButton);
71         buttonPane.add(aswdButton);
72         add(buttonPane);
73         pauseButton.addActionListener(settingHandler);
74         resumeButton.addActionListener(resumeHandler);
75         restartButton.addActionListener(restartHandler);
76         arrowButton.addActionListener(arrowKeyHandler);
77         aswdButton.addActionListener(aswdKeyHandler);
```
- Output Tab:** Shows the output tab with a status message: 2 | 41:50 | INS Windows (CRLF).



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window brick ...

Configurations.java X Driver.java X Desktop.java X GameBoard.java X

Source History

```
76     arrowButton.addActionListener(arrowKeyHandler);
77     aswdButton.addActionListener(aswdKeyHandler);
78     pauseButton.setFocusable(false);
79     restartButton.setFocusable(false);
80     resumeButton.setFocusable(false);
81     arrowButton.setFocusable(false);
82     aswdButton.setFocusable(false);
83     addKeyListener(new TAdapter());
84     setFocusable(true);
85     setPreferredSize(new Dimension(Configurations.WIDTH, Configurations.HEIGHT));
86     gameInit();
87 }
88 private void gameInit() throws IOException {
89     bricks = new Brick[Configurations.N_OF_BRICKS];
90     ball = new Ball();
91     racket = new Racket(racketType);
92     int k = 0;
93     livesLeft = 3;
94     for (int i = 0; i < 5; i++) {
95         for (int j = 0; j < 6; j++) {
96             bricks[k] = new Brick(j * 40 + 30, i * 10 + 50);
97             k++;
98         }
99     }
100    timer = new Timer(Configurations.PERIOD, new GameCycle());
101    timer.start();
102
103    @Override
104    public void paintComponent(Graphics g) {
105        super.paintComponent(g);
106        var g2d = (Graphics2D) g;
107        Image im = new ImageIcon(GameBoard.class.getResource("images\\lava.gif")).getImage();
108        g.drawImage(im, 0, 365, null);
109        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
110            RenderingHints.VALUE_ANTIALIAS_ON);
111        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
112            RenderingHints.VALUE_RENDER_QUALITY);
113        if (inGame) {
114            try {
115                drawObjects(g2d);
```

Output 2 | 77:53 | INS Windows (CRLF)



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search icon.
- Project Explorer:** Shows a single project with files: Configurations.java, Driver.java, Desktop.java, and GameBoard.java (the active file).
- Code Editor:** Displays the Java code for GameBoard.java. The code handles drawing objects, catching exceptions, and managing game states like score and lives. A cursor is visible at line 126, column 10, under the word "Close".
- Output Tab:** Shows the Output tab is selected.
- Status Bar:** Displays the time as 113:34 and the mode as INS.

```
112         try {
113             drawObjects(g2d);
114         } catch (IOException e) {
115             e.printStackTrace();
116         }
117     } else {
118         try {
119             gameFinished(g2d);
120         } catch (IOException e) {
121             e.printStackTrace();
122         }
123     }
124     Toolkit.getDefaultToolkit().sync();
125 }
126 private void drawObjects(Graphics2D g2d) throws IOException {
127     var font = new Font("Verdana", Font.BOLD, 15);
128     FontMetrics fontMetrics = this.getFontMetrics(font);
129     g2d.setColor(Color.black);
130     g2d.setFont(font);
131     // Draw current score at bottom of panel
132     g2d.drawString("Score: " + score, 120, 390);
133     g2d.drawString("Lives: " + livesLeft, 230, 390);
134     if(restartClicked) {
135         racketType = 0;
136         speed = 1;
137         speedLevel = "x1";
138         itemDrop = false;
139         restartClicked = false;
140         racket = new Racket(racketType);
141     }
142     g2d.drawString("Speed: " + speedLevel, 10, 390);
143     g2d.drawImage(ball.getImage(), (int)ball.getX(), (int)ball.getY(),
144                   ball.getImageWidth(), ball.getImageHeight(), this);
145     g2d.drawImage(racket.getImage(), (int)racket.getX(), (int)racket.getY(),
146                   racket.getImageWidth(), racket.getImageHeight(), this);
147     if(itemDrop) {
148         g2d.drawImage(drop.getImage(), (int)drop.getX(), (int)drop.getY(),
149                     drop.getImageWidth(), drop.getImageHeight(), this);
```



**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window brick ...
Configurations.java X Driver.java X Desktop.java X GameBoard.java X
Source History
149
150     drop.drawImage(drop.getImage(), (int)drop.getX(), drop.getImageWidth(), drop.getImageHeight(), this);
151 }
152 for (int i = 0; i < Configurations.N_OF_BRICKS; i++) {
153     if (!bricks[i].isDestroyed()) {
154         g2d.drawImage(bricks[i].getImage(), (int)bricks[i].getX(),
155                     (int)bricks[i].getY(), bricks[i].getImageWidth(),
156                     bricks[i].getImageHeight(), this);
157     }
158 }
159 private void gameFinished(Graphics2D g2d) throws IOException {
160     var font = new Font("Verdana", Font.BOLD, 18);
161     FontMetrics fontMetrics = this.getFontMetrics(font);
162     g2d.setColor(Color.BLACK);
163     g2d.setFont(font);
164     g2d.drawString(message,
165                 (Configurations.WIDTH - fontMetrics.stringWidth(message)) / 2,
166                 Configurations.WIDTH / 2);
167     FileWriter out = new FileWriter("ScoreList.txt", true);
168     BufferedWriter bw = new BufferedWriter(out);
169     if (score != 0) {
170         bw.write(Integer.toString(score));
171         bw.newLine();
172     }
173     bw.close();
174     out.close();
175 }
176 private class TAdapter extends KeyAdapter {
177     @Override
178     public void keyReleased(KeyEvent e) {
179         racket.keyReleased(e, keySelect);
180     }
181     @Override
182     public void keyPressed(KeyEvent e) {
183         racket.keyPressed(e, keySelect);
184     }
185     private class GameCycle implements ActionListener {
186         @Override
187         public void actionPerformed(ActionEvent e) {
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like New, Open, Save, and a dropdown for configurations.
- Project Explorer:** Shows a single project with a source folder containing files: Configurations.java, Driver.java, Desktop.java, and GameBoard.java.
- Code Editor:** The active tab is GameBoard.java. The code implements the ActionListener interface and contains methods for game logic, collision detection, and timer management.
- Status Bar:** Shows the number of errors (2), the current time (186:53), and the operating system (INS Windows (CRLF)).

```
private class GameCycle implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            doGameCycle();
        } catch (IOException ioException) {
            ioException.printStackTrace();
        }
    }

    private void doGameCycle() throws IOException {
        ball.move();
        racket.move();
        if(itemDrop) {
            drop.move();
            if (drop.y > Configurations.INIT_PADDLE_Y) {
                itemDrop = false;
            }
        }
        checkCollision();
        repaint();
    }

    private void stopGame() throws IOException {
        livesLeft--;
        itemDrop = false;
        racketType = 0;
        if(livesLeft == 0) {
            inGame = false;
            timer.stop();
        }
        ball = new Ball();
        racket = new Racket(racketType);
        timer.stop();
        timer = new Timer(Configurations.PERIOD, new GameCycle());
        timer.start();
    }

    // pause game once click on pause button
    private class PauseHandler implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            pauseGame();
        }
    }
}
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window brick ... - X
<default config> Projects GameBoard.java
Source History
public void actionPerformed(ActionEvent e) {
    pauseGame();
}
// method to pause game
private void pauseGame() {
    Container parent = pauseButton.getParent();
    parent.add(resumeButton, 0, 0);
    parent.remove(pauseButton);
    parent.revalidate();
    parent.repaint();
    timer.stop();
}
private class ResumeHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        resumeGame();
    }
}
private void resumeGame() {
    Container parent = resumeButton.getParent();
    parent.add(pauseButton, 0, 0);
    parent.remove(resumeButton);
    parent.revalidate();
    parent.repaint();
    timer.stop();
    timer = new Timer(Configurations.PERIOD, new GameCycle());
    timer.start();
}
private class RestartHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            speed = 1;
            speedLevel = "x1";
            restartClicked = true;
            inGame = true;
            timer.stop();
        }
    }
}

```

Output 222:25 INS Windows (CRLF)



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, New, and Build.
- Project Explorer:** Shows a single project with a file named "GameBoard.java" selected.
- Code Editor:** Displays the Java code for "GameBoard.java". The code handles game logic, including collision detection and key handling for movement.
- Status Bar:** Shows the number of errors (2), the current time (258:30), and the operating system (INS Windows (CRLF)).

```
200     restart();
201     inGame = true;
202     timer.stop();
203     gameInit();
204 } catch (IOException er) {
205     er.printStackTrace();
206 }
207 }
208
209 private class ArrowKeyHandler implements ActionListener{
210     @Override
211     public void actionPerformed(ActionEvent e) {
212         selectArrowKey();
213         --keySelect;
214     }
215     private void selectArrowKey(){
216         Container key = arrowButton.getParent();
217         key.add(aswdButton, 3);
218         key.remove(arrowButton);
219         key.revalidate();
220         key.repaint();
221     }
222     private class ASWDKeyHandler implements ActionListener{
223         @Override
224         public void actionPerformed(ActionEvent e) {
225             selectASWDKey();
226             ++keySelect;
227         }
228     }
229     private void selectASWDKey(){
230         Container key = aswdButton.getParent();
231         key.add(arrowButton, 0, 3);
232         key.remove(aswdButton);
233         key.getComponentAt(250,30);
234         key.revalidate();
235         key.repaint();
236     }
237     public void checkCollision() throws IOException {
238         if (ball.getRect().getMaxY() > Configurations.BOTTOM_EDGE) {
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, New, Copy, Paste, Find, and Run.
- Project Explorer:** Shows "GameBoard.java" as the active source file.
- Code Editor:** Displays Java code for a game board. The code handles ball collision detection, score keeping, and game logic. It uses variables like ball, racket, Configurations, and bricks.
- Status Bar:** Shows "294:69" and "INS Windows (CRLF)".

```
public void checkCollision() throws IOException {
    if (ball.getRect().getMaxY() > Configurations.BOTTOM_EDGE) {
        stopGame();
    }
    // game over when the ball hit the top edge
    if(ball.getRect().getMaxY() < Configurations.TOP_EDGE) {stopGame();}
    // Speeds up the ball every time
    // 5 bricks are destroyed until the 15th destroyed brick
    if(score >= 5 && score < 10){
        speed = 1.2;
        speedLevel = "x1.2";
    }
    else if(score >= 10 && score < 15){
        speed = 1.5;
        speedLevel = "x1.5";
    }
    else if(score >= 15){
        speed = 2;
        speedLevel = "x2";}
    for (int i = 0, j = 0; i < Configurations.N_OF_BRICKS; i++) {
        if (bricks[i].isDestroyed()) {
            j++;
        }
        //added score keeper
        score = j;
        if (j == Configurations.N_OF_BRICKS) {
            message = "Victory";
            stopGame();
        }
    }
    if ((ball.getRect()).intersects(racket.getRect())) {
        int paddleLPos = (int) racket.getRect().getMinX();
        int ballLPos = (int) ball.getRect().getMinX();
        int first = paddleLPos + 8;
        int second = paddleLPos + 16;
        int third = paddleLPos + 24;
        int fourth = paddleLPos + 32;
        if (ballLPos < first) {
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search bar labeled <default config>.
- Project Explorer:** Shows a single project with files: Configurations.java, Driver.java, Desktop.java, and GameBoard.java (which is the active file).
- Code Editor:** Displays the Java code for GameBoard.java. The code handles ball movement logic based on its position relative to a paddle's segments (first, second, third, fourth). It also checks for item drops and updates racket type or lives left accordingly.
- Output Tab:** Shows the Output tab at the bottom left.
- Status Bar:** At the bottom right, it shows the time (330:36), operating system (INS Windows (CRLF)), and a small icon with the number 2.

```
int first = paddleLPos + 24;
int fourth = paddleLPos + 32;
if (ballLPos < first) {
    ball.setXDir(-speed);
    ball.setYDir(-speed);
}
if (ballLPos >= first && ballLPos < second) {
    ball.setXDir(-speed);
    ball.setYDir(-speed * ball.getYDir());
}
if (ballLPos >= second && ballLPos < third) {
    ball.setXDir(0);
    ball.setYDir(-speed);
}
if (ballLPos >= third && ballLPos < fourth) {
    ball.setXDir(speed);
    ball.setYDir(-speed * ball.getYDir());
}
if (ballLPos > fourth) {
    ball.setXDir(speed);
    ball.setYDir(-speed);
}

//check if the user caught a dropped item
if (itemDrop && (drop.getRect()).intersects(racket.getRect())) {
    int random = (int) (Math.random() * 100) + 1;
    //case 1: lengthen paddle
    if(random < 50) {
        racketType = 1;
    }
    else if(random > 50 && random < 80){
        livesLeft++;
    }
    //case 2: shorten paddle
    else {
        racketType = 2;
    }
    //have new racket appear under ball
    double temp = ball.x;
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

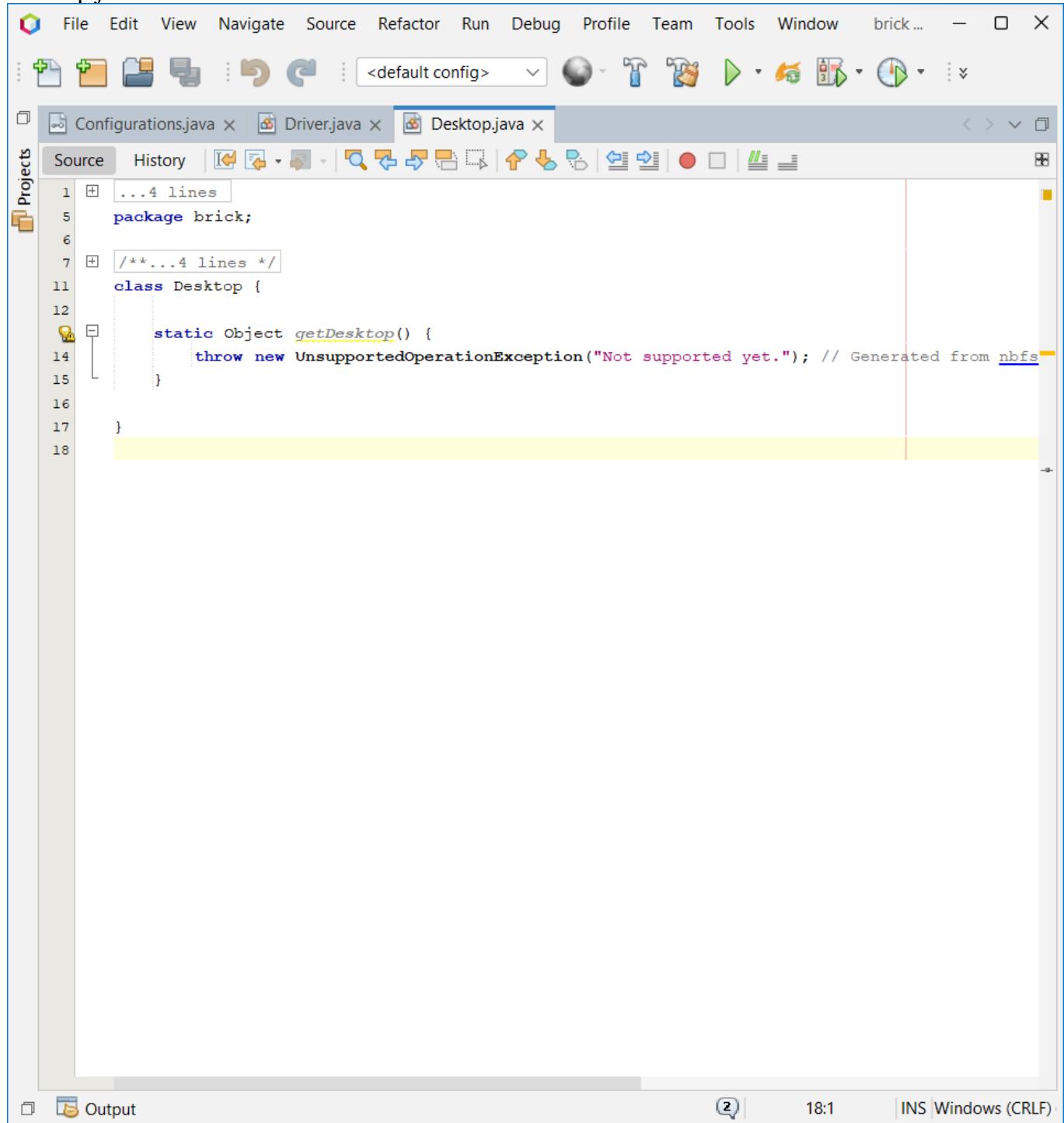
```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window brick ... - X
Configurations.java X Driver.java X Desktop.java X GameBoard.java X
Source History
365     //have new racket appear under ball
366     double temp = ball.x;
367     racket = new Racket(racketType);
368     racket.x = temp;
369     //reset itemDrop condition
370     itemDrop = false;
371 }
372 for (int i = 0; i < Configurations.N_OF_BRICKS; i++) {
373     if ((ball.getRect()).intersects(bricks[i].getRect())) {
374         int ballLeft = (int) ball.getRect().getMinX();
375         int ballHeight = (int) ball.getRect().getHeight();
376         int ballWidth = (int) ball.getRect().getWidth();
377         int ballTop = (int) ball.getRect().getMinY();
378         var pointRight = new Point(ballLeft + ballWidth + 1, ballTop);
379         var pointLeft = new Point(ballLeft - 1, ballTop);
380         var pointTop = new Point(ballLeft, ballTop - 1);
381         var pointBottom = new Point(ballLeft, ballTop + ballHeight + 1);
382         if (!bricks[i].isDestroyed()) {
383             if (bricks[i].getRect().contains(pointRight)) {
384                 ball.setXDir(-speed);
385             } else if (bricks[i].getRect().contains(pointLeft)) {
386                 ball.setXDir(speed);
387             }
388             if (bricks[i].getRect().contains(pointTop)) {
389                 ball.setYDir(speed);
390             } else if (bricks[i].getRect().contains(pointBottom)) {
391                 ball.setYDir(-speed);
392             }
393             //if it should drop an item
394             if(bricks[i].hasItem()) {
395                 itemDrop = true;
396                 drop = new item(bricks[i].x, bricks[i].y);
397             }
398             bricks[i].doDamage();
399         }
400     }
401 }
402 }
```

The screenshot shows an IDE interface with multiple tabs at the top: Configurations.java, Driver.java, Desktop.java, and GameBoard.java (which is the active tab). The main area displays Java code for the GameBoard class. The code handles ball collision with bricks and racket appearance. It uses Point objects to check collision boundaries and sets ball movement direction based on which brick edge was hit. It also manages item drops from destroyed bricks.



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

## Desktop.java



The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, brick ...
- Toolbar:** Includes icons for file operations like New, Open, Save, and a dropdown for configurations.
- Project Bar:** Shows three files: Configurations.java, Driver.java, and Desktop.java (which is the active tab).
- Source Tab:** Displays the Java code for the Desktop class. The code includes a package declaration for 'brick', a class definition for 'Desktop' with a static method 'getDesktop()' that throws an 'UnsupportedOperationException'. A note at the end of the method indicates it was generated from 'nbfs'.
- Output Tab:** Located at the bottom left, showing the word 'Output'.
- Status Bar:** Located at the bottom right, showing '2' (indicating two open files), '18:1', and 'INS Windows (CRLF)'.

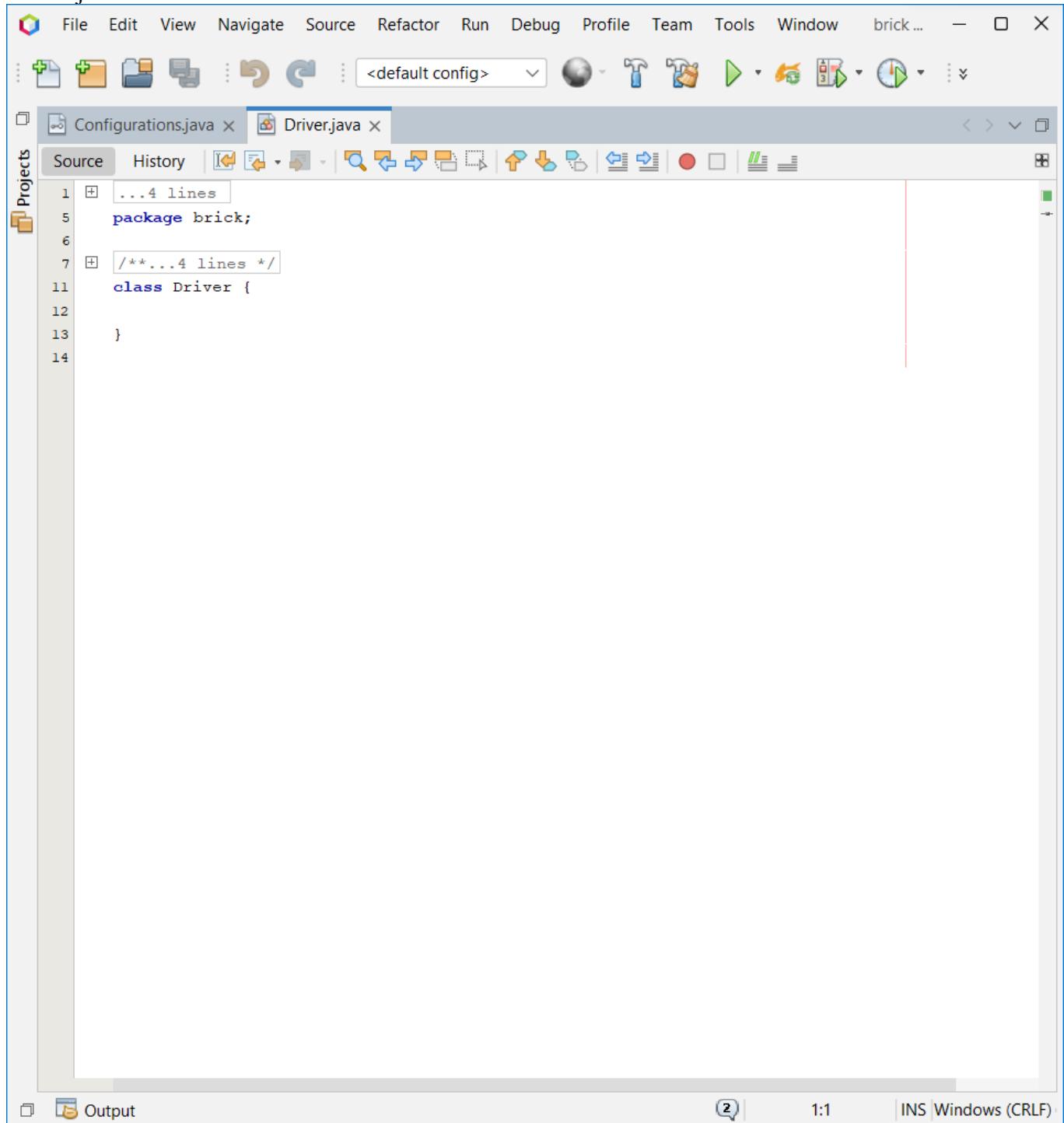
```
1  ...4 lines
5  package brick;
6
7  /**...4 lines */
8  class Desktop {
9
10     static Object getDesktop() {
11         throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs
12     }
13
14 }
15
16
17 }
18
```



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

॥यमार्पयत् युद्धम्॥

## Driver.java



The screenshot shows a Java code editor interface. The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and brick ... . The toolbar contains various icons for file operations like new, open, save, and search. The project navigation bar shows Configurations.java and Driver.java, with Driver.java currently selected. The code editor displays the following Java code:

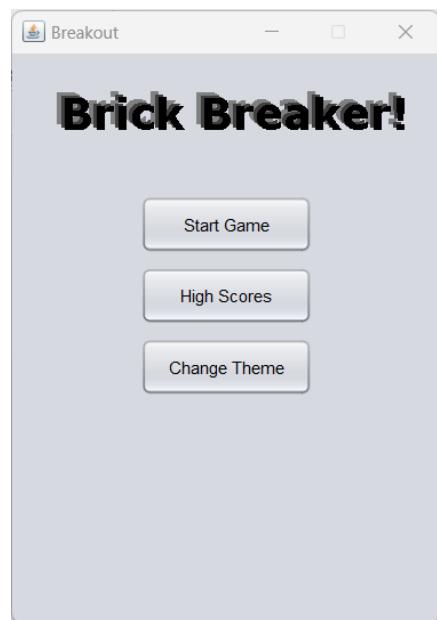
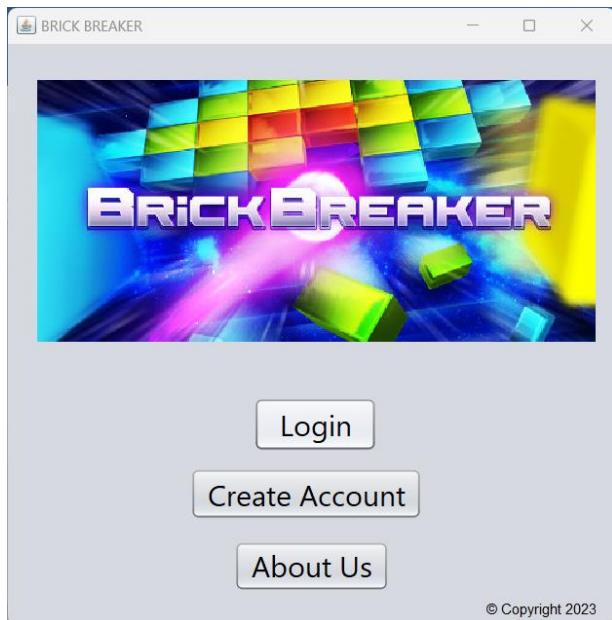
```
1  ...4 lines
5  package brick;
6
7  /**...4 lines */
8  class Driver {
9
10 }
11
12 }
```

The code consists of a single class definition named Driver. The class has no methods or fields. The code editor interface includes a Projects panel on the left, a Source tab at the top, and an Output panel at the bottom. The status bar at the bottom right shows 2, 1:1, and INS Windows (CRLF).



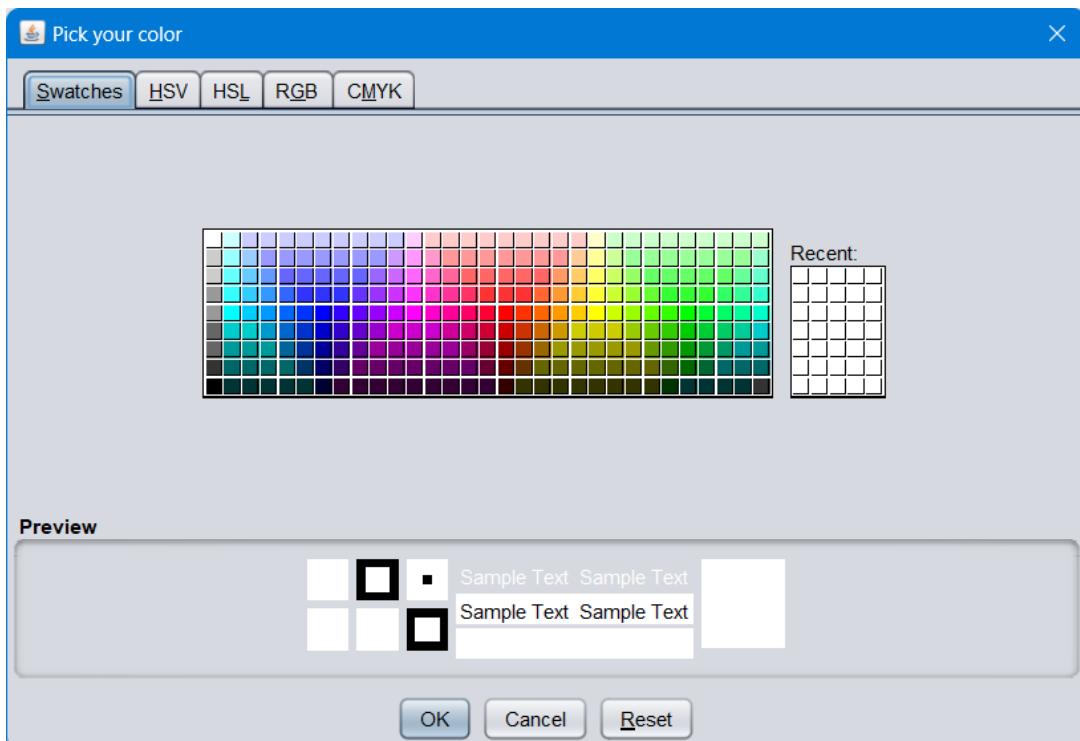
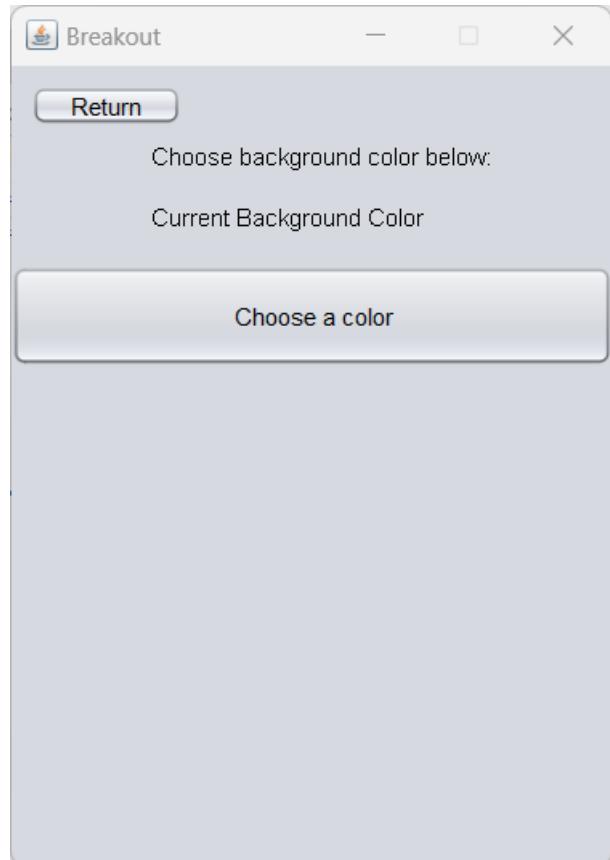
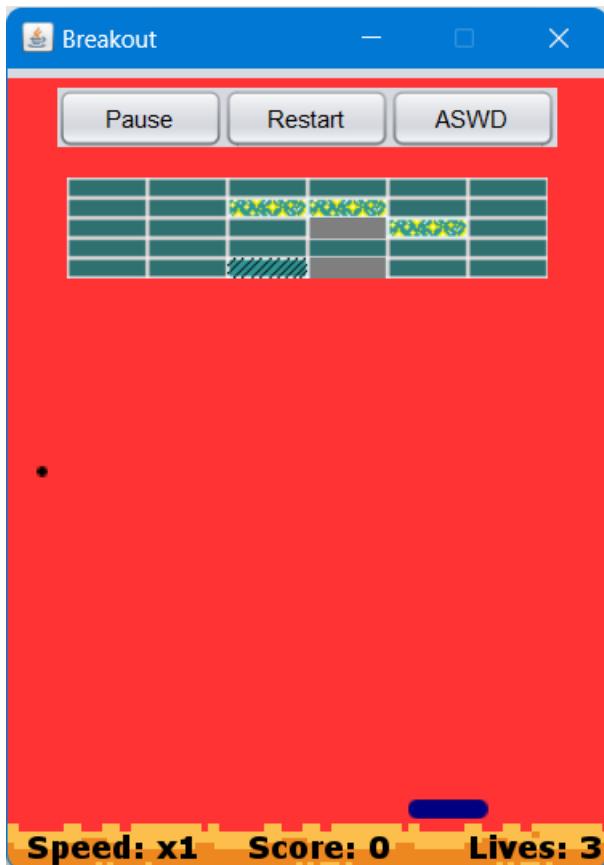
SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

**Output:**

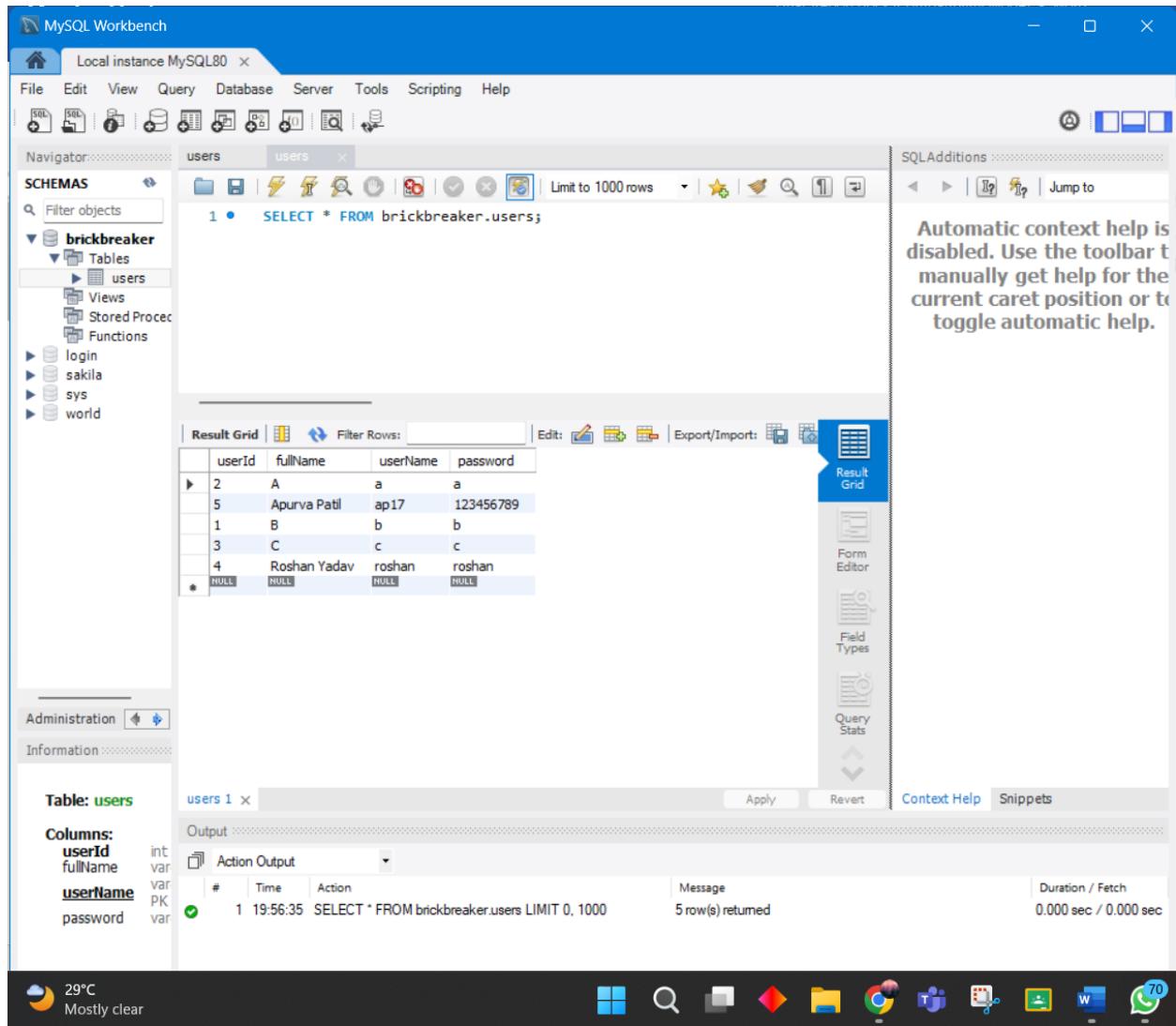
A screenshot of a registration form titled "Brick Breaker". It includes fields for "Full Name", "Username", "Password", and "Confirm Password", each with an associated input box. A "Create" button is located at the bottom right. A "Return" button is in the top left corner. The background is blue.A screenshot of a login form titled "Breakout". It features a cartoon illustration of a sad-looking monkey covering its face with its hands. Below the illustration are fields for "UserName" and "Password", each with an associated input box. A "Login" button is at the bottom right. A "Return" button is in the top left corner. The background is blue.



# SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)



## Database:



The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with the 'brickbreaker' database selected, containing the 'users' table. The main area shows the results of the query `SELECT * FROM brickbreaker.users;`. The results are displayed in a grid with columns: userId, fullName, userName, and password. The data is as follows:

	userId	fullName	userName	password
1	2	A	a	a
2	5	Apurva Patil	ap17	123456789
3	1	B	b	b
4	3	C	c	c
5	4	Roshan Yadav	roshan	roshan
	NULL	NULL	NULL	NULL

The bottom status bar shows the weather as 29°C Mostly clear and a system tray with various icons.

## References

- [\(PDF\) Effects of digital game-based learning on students' self-efficacy, motivation, anxiety, and achievements in learning mathematics \(researchgate.net\)](#)
- [Brick Breaker Game - Brick Breaker Game Abstract The project that is going to be implemented will consist of a game in which the player smashes a wall | Course Hero](#)
- [Brick Breaker Game using Tkinter \(Python Project\) - Studytonight](#)
- [Research on of 3D game design and development technology | IEEE Conference Publication | IEEE Xplore](#)

