



โครงการ

Mini Project

จัดทำโดย

6504062630022 นางสาวกิตติมา แพงไพร

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 ที่มาและความสำคัญของโครงการ

โครงการนี้จัดขึ้นเพื่อประยุกต์การเรียนรู้ในรายวิชา Object Oriented Programming ในการสร้างชิ้นงานที่มีความสำคัญและน่าสนใจ โดยผู้จัดทำได้ใช้ความชื่นชอบส่วนตัวที่เกี่ยวข้องกับแสงและสีของเมืองในตอนกลางคืนเพื่อสร้างความเข้าใจในบทเรียนอย่างลึกซึ้งมากยิ่งขึ้น

ผู้จัดทำได้นำเสนอเกมที่มีตัวละครที่สะท้อนตัวเอง ซึ่งเป็นทักษะที่สำคัญในการพัฒนาทักษะด้านเทคนิคและการแก้ปัญหา นอกจากนี้ โครงการยังเป็นทางเลือกที่ดีในการฝึกฝนการทำงาน การวางแผน การจัดการเวลา และการดำเนินการ โครงการที่เป็นที่น่าพอใจ

ผ่านการพัฒนาโครงการนี้ ผู้จัดทำได้มีโอกาสประยุกต์การเรียนรู้ในการสร้างเกมและการใช้งานหลักการที่ได้เรียนรู้ในวิชา Object Oriented Programming ซึ่งเป็นพื้นฐานที่สำคัญในการพัฒนาซอฟต์แวร์ในปัจจุบัน ทั้งนี้ยังช่วยให้ผู้จัดทำมีความเข้าใจในการนำหลักการเหล่านี้ไปใช้ในการสร้างผลงานที่มีคุณภาพในอนาคตได้อีกด้วย

ประเภทโครงการ

เกม

ประโยชน์

- 1.ฝึกสมาธิ
- 2.ฝึกความอดทนและความพยายาม
- 3.เพื่อความบันเทิง

ขอบเขตของโครงการ

- ตารางการทำงานเดือนกันยายน-พฤษภาคม

ลำดับ	รายการ	5 ก.ย. – 15 ก.ย.	16 ก.ย. – 30 ก.ย.	1 ต.ค. – 31 ต.ค.
1	หารูปจัดตัวละครและกราฟิกต่างๆ			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบและแก้ไขข้อผิดพลาด			

เล็ท เกิร์ล อัป

(Let's Girl Up!)

- รายละเอียดเกม

คุณจะต้องควบคุมตัวละครเด็กผู้หญิง ให้กระโดดผ่านสิ่งกีดขวางต่างๆแล้วเก็บเหรียญ ถ้าหากเด็กผู้หญิงชนสิ่งกีดขวาง พลังชีวิตของเด็กจะลดลง และถ้าชนสิ่งกีดขวางจนกระทั่งพลังชีวิตหมดก็จะเป็นการจบเกม คุณจะได้เห็นความแตกต่างเล็กน้อยของฉากและสิ่งกีดขวางในเกมที่จะทำให้เกมนี้น่าสนใจมากขึ้น

- วิธีการเล่น

เมื่อเริ่มเกม ถ้าเจอสิ่งกีดขวาง กดปุ่ม Space bar หรือปุ่มลูกศรขึ้น เพื่อบังคับให้เด็กผู้หญิงให้กระโดดขึ้นเพื่อหลบสิ่งกีดขวาง เมื่อเจอเหรียญให้ทำการเก็บ โดยการชนกับเหรียญ

- Storyboard

ตัวละคร



เด็กผู้หญิง



แมว



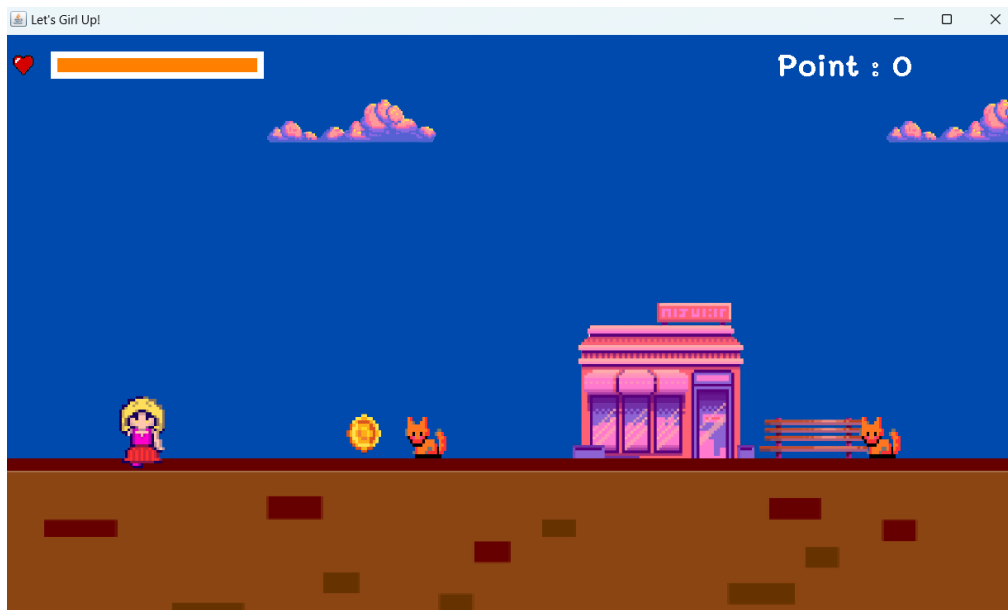
เหรียญ

ฉาก

-เริ่มเกม



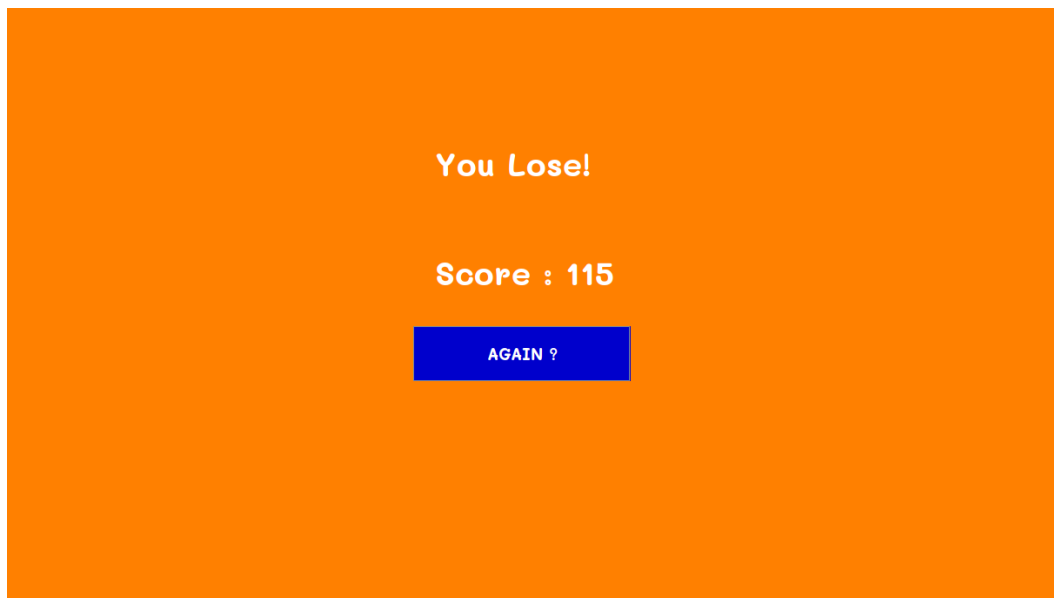
-เมื่อเริ่มเกมจะมีพลังชีวิตมาให้ และเริ่มนับคะแนนเมื่อเก็บเหรียญได้



-เมื่อเจอสิ่งกีดขวางให้กดปุ่ม Space bar หรือปุ่มลูกศรขึ้นเพื่อกระโดดหลบสิ่งกีดขวาง



-เมื่อชนสิ่งกีดขวางจนพลังชีวิตหมดจะเป็นการจบเกม



บทที่ 2 ส่วนการพัฒนา

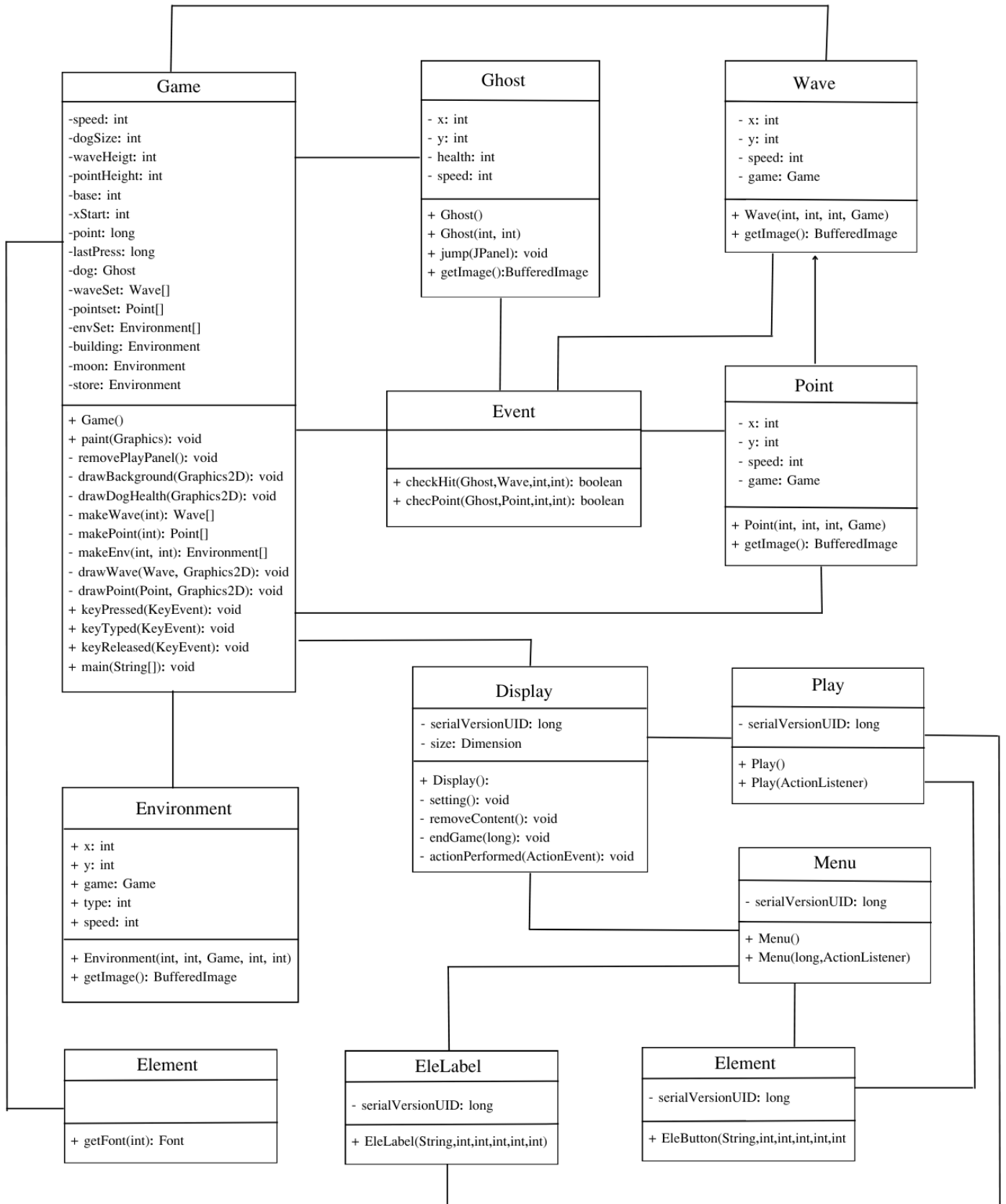
2.1) เนื้อเรื่องย่อ

ผู้เล่นจะได้รับบทเป็นเด็กผู้หญิงที่เดินเล่นในเมืองยามค่ำกึ่ง และต้องเจอเข้ากับแมวส้ม เด็กหญิงจะต้องหลบหนีจากแมวส้มและทำการเก็บเหรียญ จะต้องใช้ทักษะในการกระโดดเพื่อหลบหนีจากแมวและรักษาพลังชีวิตของตัวเองให้มากที่สุด โดยมีเป้าหมายคือเก็บเหรียญเพื่อให้ได้คะแนนมากที่สุดเท่าที่ทำได้ เกมนี้จะมีกราฟิกที่น่ารักเพื่อให้ผู้เล่นได้รับประสบการณ์ที่น่าสนใจ

2.2) วิธีการเล่น

เมื่อเริ่มเกม ถ้าเจอสิ่งกีดขวาง กดปุ่ม Space bar หรือปุ่มลูกศรขึ้น เพื่อบังคับให้เด็กผู้หญิงให้กระโดดขึ้นเพื่อหลบสิ่งกีดขวาง เมื่อเจอเหรียญก็ทำการเก็บเหรียญด้วยการชนเหรียญ

2.3) คลาสไดอะแกรม



- **Game Class (คลาสเกม):** คลาสนี้เป็นคลาสหลักที่ควบคุมการทำงานของเกม มีตัวแปรและเมธอดต่างๆ เช่น การวาดฉากหลัง, การตรวจสอบการชน, การจัดการการกระโดดของตัวละคร และการจัดการกับการกดแป้นพิมพ์
- **Ghost Class (คลาสตัวละคร):** คลาสนี้เป็นคลาสที่จัดการกับตัวละครหลักของเกม มีตัวแปรเกี่ยวกับตำแหน่ง, รูปภาพ, และพลังชีวิตของตัวละคร รวมถึงเมธอดเกี่ยวกับการกระโดดของตัวละคร
- **Environment Class (คลาสสิ่งแวดล้อม):** คลาสนี้เป็นคลาสที่จัดการกับสิ่งแวดล้อมในเกม เช่น อาคาร, ก้อนเมฆ, รถ มีตัวแปรที่เก็บตำแหน่งและประเภทของสิ่งแวดล้อม และเมธอดที่ใช้ในการดึงรูปภาพสิ่งแวดล้อม
- **Wave Class (คลาสสิ่งกีดขวาง):** คลาสนี้เป็นคลาสที่จัดการกับสิ่งกีดขวางที่เคลื่อนที่ในเกม มีตัวแปรเกี่ยวกับตำแหน่งและความเร็วของสิ่งกีดขวาง รวมถึงเมธอดที่ใช้ในการดึงรูปภาพของสิ่งกีดขวาง
- **Point Class (คลาสเหรียญ):** คลาสนี้เป็นคลาสที่จัดการกับเหรียญที่ใช้ในการนับคะแนน มีตัวแปรเกี่ยวกับตำแหน่งและความเร็วของเหรียญ รวมถึงเมธอดที่ใช้ในการดึงรูปภาพของเหรียญ
- **Event Class (คลาสอีเวนต์):** คลาสนี้เป็นคลาสที่จัดการเกี่ยวกับ event handling จัดการกับเหตุการณ์ต่างๆ ที่เกิดขึ้นในเกม
- **Display Class (คลาสแสดงผล):** คลาสนี้เป็นส่วนหนึ่งของการจัดการหน้าต่างแสดงผลของเกม โดยมีการใช้งาน JFrame และส่วนอื่น ๆ ที่เกี่ยวข้อง เพื่อจัดการกับการแสดงผลและการกระทำต่าง ๆ ภายในเกม
- **Play Class (คลาสหน้าจอ Play):** คลาสที่ถูกสร้างขึ้นเพื่อแสดงหน้าจอเล่นเกม คือ หน้าเริ่มต้น โดยมีการเพิ่ม Element เช่น Label และ Button เพื่อให้ผู้เล่นสามารถทำการเริ่มเกมหรือทำกิจกรรมต่าง ๆ ในเกมได้
- **Menu Class (คลาสเมนู):** ถูกสร้างขึ้นเพื่อแสดงหน้าจอเมนูหลังจบเกม โดยมีการแสดงคะแนนที่ผู้เล่นได้ทำได้ในเกม และปุ่มที่ให้ผู้เล่นเริ่มเกมใหม่
- **Element Class:** มีเมธอดที่ชื่อ `getFont` ซึ่งเป็นส่วนหนึ่งในการจัดการกับแบบอักษร (Font) ในเกม โดยมีการนำเข้าคลาส `Font` และ `File` เพื่อใช้ในการสร้างและปรับแต่งแบบอักษรตามที่ต้องการ

- **EleButton Class:** คลาส EleButton สืบทอดมาจากคลาส JButton ซึ่งถูกสร้างขึ้นเพื่อสร้างปุ่มที่ปรับแต่งสีพื้นหลัง, สีตัวอักษร, และแบบอักษรตามที่กำหนด โดยมีการใช้งานคลาส Element ในการรับค่าแบบอักษร
- **EleLabel Class(คลาส):** คลาส EleLabel ที่สืบทอดมาจากคลาส JLabel ซึ่งถูกสร้างขึ้นเพื่อสร้างแท็กข้อความที่สามารถปรับแต่งขนาดและรูปแบบตามที่กำหนด โดยมีการใช้งานคลาส Element ในการรับค่าแบบอักษร

2.4) รูปแบบการพัฒนา Application / Applet

เป็นการพัฒนา Application ที่ใช้ภาษา Java โดยใช้ไลบรารี Swing ในการสร้างหน้าต่างและกราฟิกสำหรับเกม โดยโค้ดเกมประกอบด้วยหลายคลาสที่ทำหน้าที่ต่าง ๆ ดังนี้:

1. **Charactor.Ghost:** คลาสนี้เป็นคลาสที่เกี่ยวข้องกับตัวละครหลักในเกม มีเมธอดต่าง ๆ เช่น jump และ getImage ที่ใช้ในการควบคุมการกระโดดและการดึงรูปภาพของตัวละคร
2. **display.Game:** คลาสนี้เป็นหน้าต่างหลักของเกม มีเมธอดที่ใช้ในการวาดกราฟิกและการจัดการกับการกระทำต่าง ๆ ในเกม เช่นการวาดพื้นหลัง ตัวละคร การกระโดด และควบคุมการกดปุ่ม
3. **display.Display:** คลาสนี้ใช้สำหรับแสดงหน้าต่างของเกม และมีเมธอดที่ใช้ในการกำหนดการตั้งค่าหน้าต่าง เช่น การกำหนดชื่อและขนาดหน้าต่าง การแสดงผลผลการเล่น และการรีเซ็ตเกม

โดยทั้งสามคลาสนี้เป็นส่วนสำคัญในการพัฒนา Application แบบ Desktop ที่ใช้งานบนเครื่องคอมพิวเตอร์ และไม่ได้ใช้ Applet เนื่องจากไม่ได้นำโค้ดมาใช้งานบนเว็บไซต์ หากต้องการนำโค้ดไปใช้งานในรูปแบบ Applet จะต้องปรับแต่งโค้ดเพิ่มเติมเพื่อให้สามารถทำงานได้อย่างถูกต้องในบริบทของเว็บเบราว์เซอร์ได้อย่างเหมาะสม.

2.5) ส่วนของโปรแกรมที่เกี่ยวกับหลักการOOP

- **Constructor**

-มีการเรียกใช้ Constructor ของคลาส Game

```
39
40●   public Game() {
41       this.setBounds(0,0,1000,600);
42       this.addKeyListener(this);
43       this.setLayout(null);
44       this.setFocusable(true);
45
46●       playPanel = new Play(new ActionListener() {
47●           @Override
48               public void actionPerformed(ActionEvent e) {
49                   removePlayPanel();
50                   gameStarted = true;
51               }
52           });
53       this.add(playPanel);
54   }
```

-มีการเรียกใช้ Constructor ของคลาส Ghost

```
18
19●   public Ghost() {
20   }
21
22●   public Ghost(int x,int y) {
23       this.x=x;
24       this.y=y;
25   }
26
```

-มีการเรียกใช้ Constructor ของคลาส Wave

```
17●   public Wave(int x,int y,int speed,JPanel page) {
18       this.x = x;
19       this.y = y;
20       this.speed = speed;
21       this.move(page);
22   }
23
```

-มีการเรียกใช้ Constructor ของคลาส Point

```
13
14●   public Point(int x, int y, int speed, JPanel page) {
15       super(x, y, speed, page);
16       // TODO Auto-generated constructor stub
17   }
```

-มีการเรียกใช้ Constructor ของคลาส Environment

```
20●    public Environment(int x,int y,JPanel page,int eType,int speed) {
21        this.x = x;
22        this.y = y;
23        this.startX = x;
24        this.speed = speed;
25        this.eType = eType;
26        this.move(page);
27    }
```

-มีการเรียกใช้ Constructor ของคลาส Display

```
14
15●    public Display() {
16        this.setting();
17        this.getContentPane().add(new Game()); // เพิ่ม Game ลงใน JFrame
18
19    }
20
```

-มีการเรียกใช้ Constructor ของคลาส Menu

```
15
16●    public Menu() {
17        //----
18    }
19
20●    public Menu(long point,ActionListener main) {
21        try {
22            this.point = point;
23            this.setBackground(new Color(255, 128, 0));
24            this.setBounds(0,0,1000,600);
25            this.setFocusable(true);
26            this.setLayout(null);
27
28            EleLabel status = new EleLabel("You Lose!",30,400,100,200,100);
29            status.setForeground(Color.white);
30
31            EleLabel showPoint = new EleLabel("Score : "+this.point,30,400,200,200,100);
32            showPoint.setForeground(Color.white);
33
34            EleButton restart = new EleButton("AGAIN ?",15,380,300,200,50);
35            restart.addActionListener(main);
36
37            this.add(showPoint);|
38            this.add(status);
39            this.add(restart);
40        } catch (Exception e) {
41            e.printStackTrace();
42        }
43    }
```

-มีการเรียกใช้ Constructor ของคลาส Play

```
15
16 public Play() {
17     //----
18 }
19
20 public Play(ActionListener main) {
21     try {
22         this.setBackground(new Color(255, 128, 0));
23         this.setBounds(0,0,1000,600);
24         this.setFocusable(true);
25         this.setLayout(null);
26
27         EleLabel status = new EleLabel("Let's Girl Up!",30,400,100,200,100);
28         status.setForeground(Color.white);
29
30         EleButton play = new EleButton("PLAY",15,380,300,200,50);
31         play.addActionListener(main);
32
33         this.add(status);
34         this.add(play);
35     } catch (Exception e) {
36         e.printStackTrace();
37     }
38 }
39
40 }
41
```

● Encapsulation

-ประกาศตัวแปร jumpImage และ isJumping ในคลาส Ghost ในส่วนของ private

```
19
20 private BufferedImage jumpImage;
21 private boolean isJumping = false;
22
```

● Composition

- คลาส Game มีความสัมพันธ์แบบ Composition กับคลาส Ghost, Wave, Point และ Environment

```
26
27 private Ghost dog = new Ghost(100,base-50);
28 static Display display;
29
30 // -----Wave Size -----
31 private Wave[] waveSet = makeWave(4);
32 //-----Environment-----
33 private Environment[] envSet = makeEnv(2,Environment.CLOUD);
34 private Environment building = new Environment(xStart-100,base-150,this,Environment.BUILDING,5);
35 private Environment moon = new Environment(xStart+600,base-150,this,Environment.MOON,8);
36 //-----Point Size-----
37 private Point[] pointset = makePoint(4);
38
```

-คลาส Display มีความสัมพันธ์แบบ Composition กับคลาส Game และ Menu

```
14
15 public Display() {
16     this.setting();
17     this.getContentPane().add(new Game()); // เพิ่ม Game ลงใน JFrame
18
19 }
20
```

```
34
35 public void endGame(long point) {
36     removeContent();
37     this.getContentPane().add(new Menu(point, this));
38 }
```

-คลาส Menu มีความสัมพันธ์แบบ Composition โดยการเพิ่ม showPoint, status, restart ลงใน Menu

```
36
37     this.add(showPoint);
38     this.add(status);
39     this.add(restart);
40 } catch (Exception e) {
41     e.printStackTrace();
42 }
```

-คลาส Play มีการใช้งานหลักการ Composition โดยการสร้างอ็อบเจกต์ EleLabel และ EleButton แล้วนำมาใช้ในคลาส Play

```
EleLabel status = new EleLabel("Let's Girl Up!", 30, 400, 100, 200, 100);
status.setForeground(Color.white);

EleButton play = new EleButton("PLAY", 15, 380, 300, 200, 50);
play.addActionListener(main);

this.add(status);
this.add(play);
```

- **Polymorphism** -มีการใช้หลักการ Polymorphism โดยการใช้ this.getContentPane().add() ในการเพิ่มวัตถุของคลาส Game หรือ Menu ลงใน JFrame ของ Display

```
34
35 public void endGame(long point) {
36     removeContent();
37     this.getContentPane().add(new Menu(point, this));
38 }
39
40 @Override
41 public void actionPerformed(ActionEvent e) {
42     if(e.getActionCommand().equals("AGAIN ?")) {
43         removeContent();
44         Game game = new Game();
45         this.getContentPane().add(game);
46         game.requestFocus();
47     }
48 }
```

- **Inheritance**

-คลาส Game สืบทอดลักษณะจาก JPanel

```

15
16 public class Game extends JPanel implements KeyListener{
17
18     private static final long serialVersionUID = 1L;
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
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
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
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
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

-คลาส Point สืบทอดลักษณะจาก Wave

```
11
12 public class Point extends Wave{
13
14     public Point(int x, int y, int speed, JPanel page) {
15         super(x, y, speed, page);
16         // TODO Auto-generated constructor stub
17     }
18     public void move(JPanel page) {
19         this.timeMove = new Timer(speed, new ActionListener() {
```

-คลาส Display สืบทอดลักษณะจาก JFrame

```
9
10 public class Display extends JFrame implements ActionListener{
11
12     private static final long serialVersionUID = 1L;
13     private Dimension size = new Dimension(1000,600);
14
15     public Display() {
16         this.setting();
17         this.getContentPane().add(new Game()); // เพิ่ม Game ลงใน JFrame
18     }
19
20
21     private void setting() {
22         this.setTitle("Let's Girl Up!");
23         this.setSize(size);
24         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25         this.setLocation(280,100);
26         this.setVisible(true);
27     }
}
```

-คลาส Environment ได้ใช้การสืบทอดเนื่องจากการใช้งานเมธอด move และ stop ที่ถูกสืบทอดมาจากคลาสหลัก

```
28
29     public void move(JPanel page) {
30         this.timeMove = new Timer(10,new ActionListener() {
31             public void actionPerformed(ActionEvent e) {
32                 if(x+400<0) {
33                     x = startX;
34                 }
35                 x -= speed;
36                 page.repaint();
37             }
38         });
39         this.timeMove.start();
40     }
41
42     public void stop() {
43         this.timeMove.stop();
44     }
45 }
```


2.6) หน้าจอ GUI

2.6.1) ส่วนประกอบของ GUI

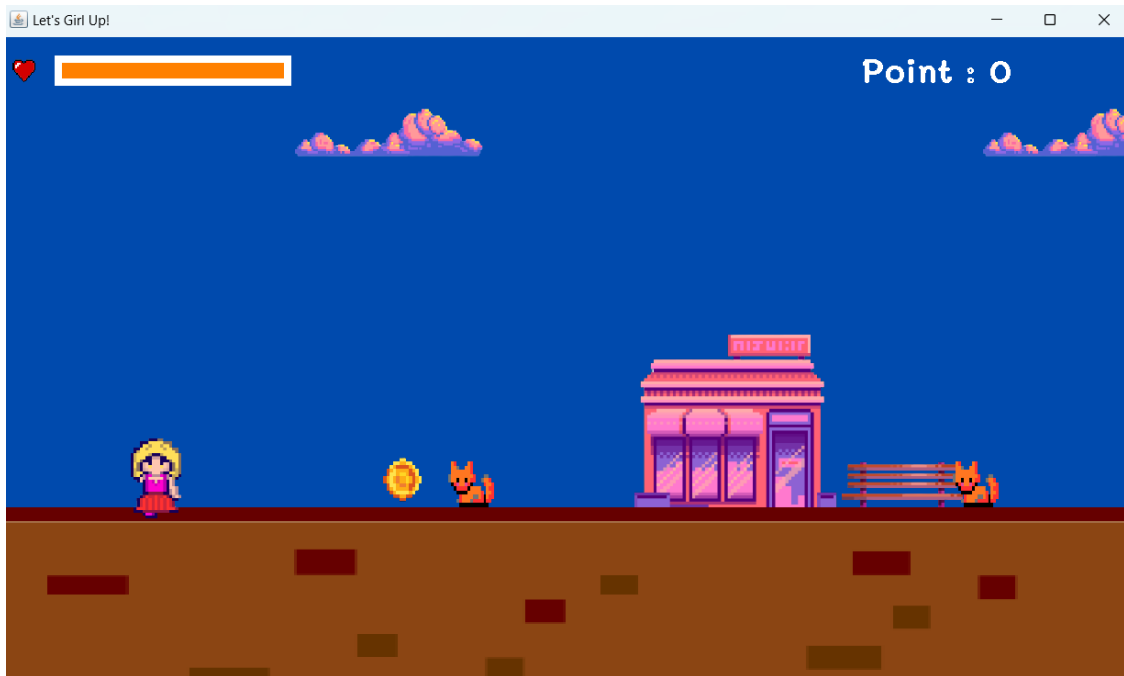
1. JPanel (display.Game): สร้าง JPanel ที่ใช้เป็นพื้นที่การทำงานหลักของ GUI ซึ่งในที่นี้ถูกใช้เพื่อแสดงภาพ, สร้างการแสดงผล, และจัดการกับเหตุการณ์ต่าง ๆ เช่น KeyEvents
2. JFrame (display.Display): สร้าง JFrame ที่เป็นหน้าต่างหลักของแอปพลิเคชัน GUI โดยที่มีการกำหนดขนาด, ชื่อ, และตำแหน่งของหน้าต่าง
3. Graphics2D: สำหรับการวาดภาพและกราฟิกต่าง ๆ บน JPanel
4. KeyListener: ใช้ในการรับข้อมูลเกี่ยวกับการกดปุ่มบนแป้นพิมพ์เพื่อกระทำตามที่กำหนด
5. Charactor Package: มีคลาส Ghost และ Environment ที่ใช้ในการสร้างวัตถุต่าง ๆ ในเกม
6. Element Package: ประกอบด้วยเอกสารและองค์ประกอบที่ใช้ในการสร้างข้อความและภาพ

-หน้าจอ Play ก่อนเริ่มเกม จะมีปุ่ม PLAY เพื่อให้ผู้เล่นกดเริ่มเล่น

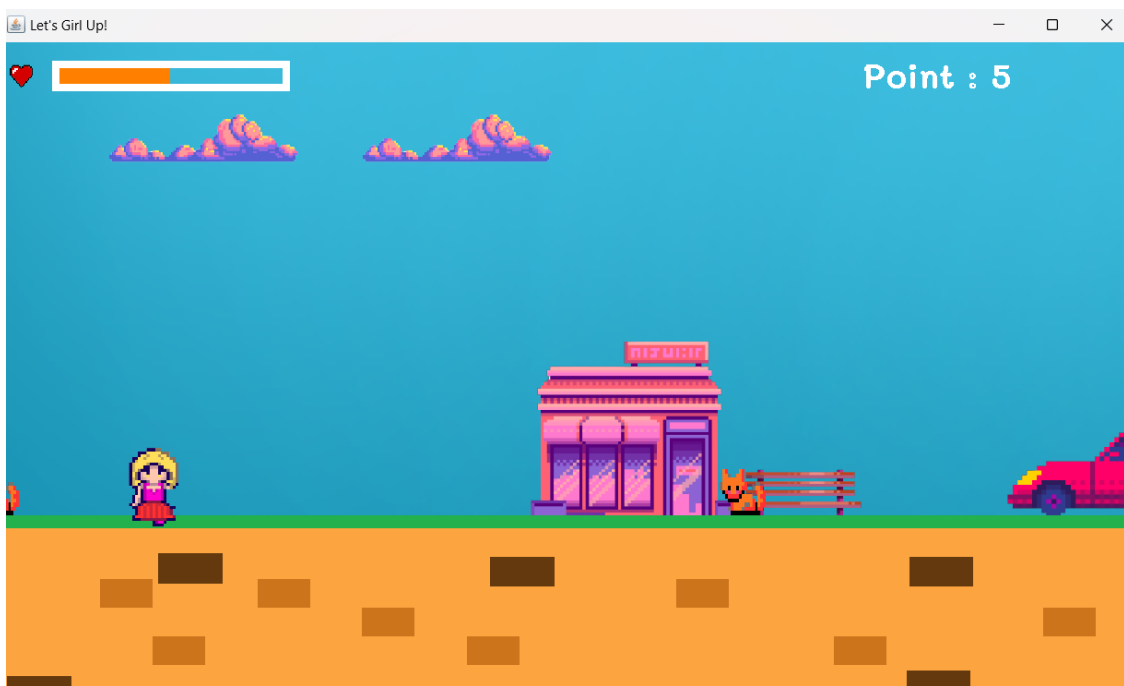


-เริ่มเกม มีหลอดพลังชีวิตที่มุมซ้ายบน มีการนับคะแนนที่มุมขวาบน

มีตัวละครเด็กผู้หญิง สิ่งกีดขวาง(แมว) เหยี่ยว และสิ่งแฉะลื่น คือ ก้อนเมฆ อาคาร และรถ



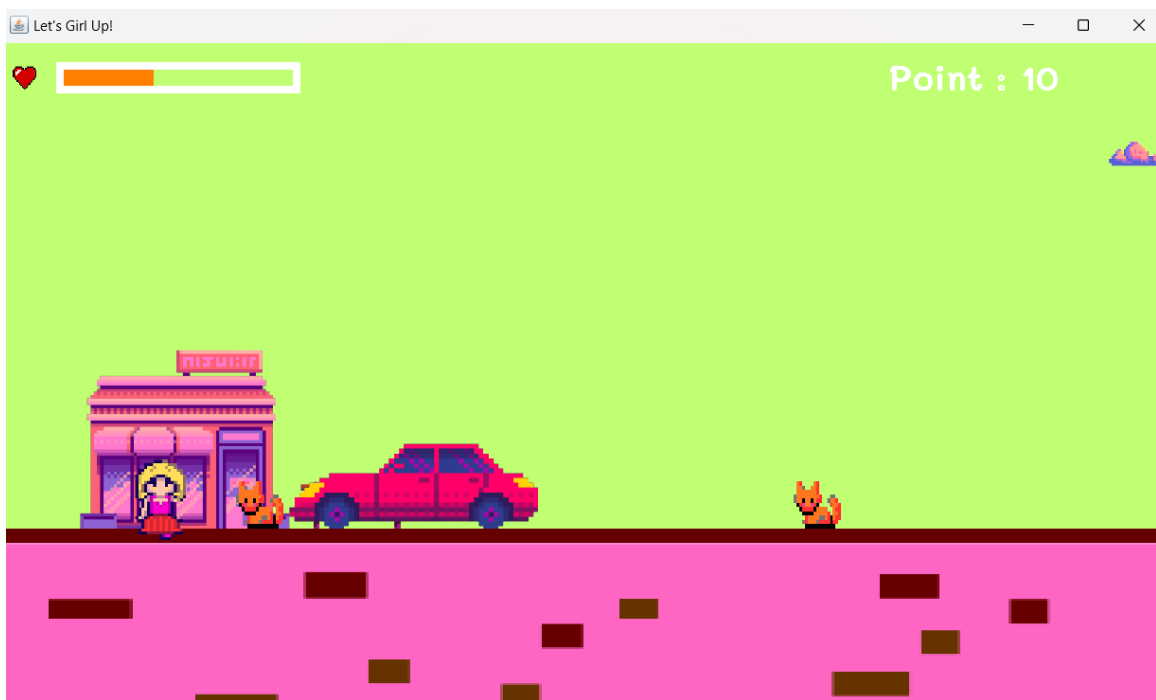
-เมื่อเล่นได้คะแนนมากกว่าเท่ากับ 4 ขึ้นไปจะทำการเปลี่ยนด่าน



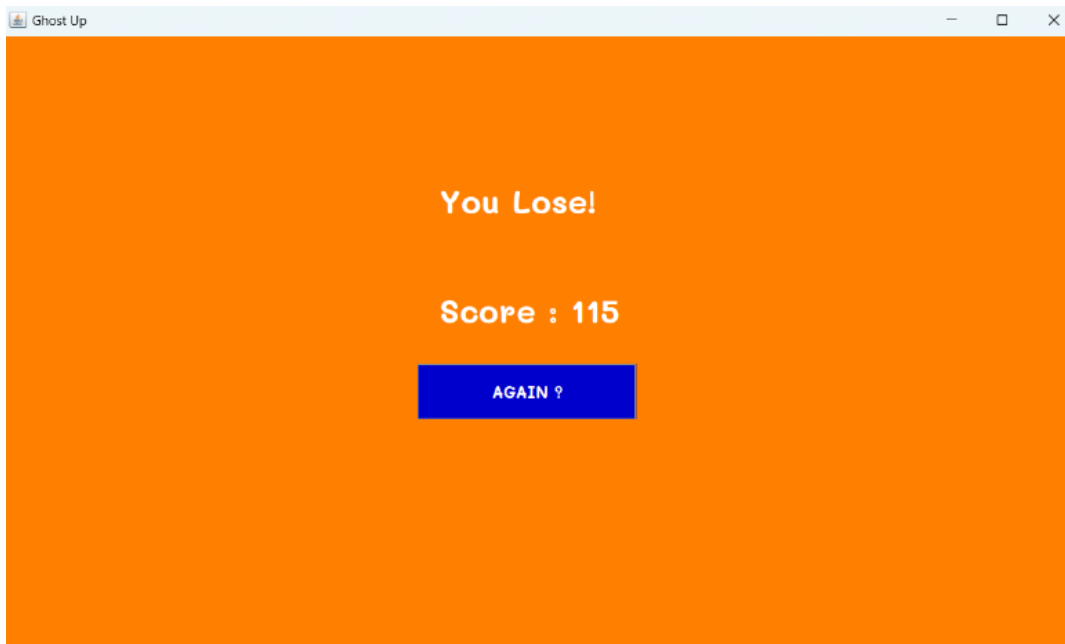
-เมื่อเล่นได้คะแนนมากกว่าเท่ากับ 7 ขึ้นไปจะทำการเปลี่ยนด่าน



-เมื่อเล่นได้คะแนนมากกว่าเท่ากับ 10 ขึ้นไปจะทำการเปลี่ยนด่าน



-จบเกมมีเมนูแสดงคะแนนที่เล่นได้ และมีปุ่ม AGAIN ? ให้กดเล่นอีกครั้ง



2.7) Event handling ที่มีหน้าจอ

1. checkHit(Ghost dog, Wave wave, int dogSize, int waveHeight):

คือเมธอดที่ใช้ในการตรวจสอบว่าวัตถุทั้งสองมีการชนกันหรือไม่ โดยในที่นี้เป็นการตรวจสอบว่าวัตถุ dog (ตัวละคร) และ wave (เวฟกริดขวาง) มีพื้นที่ทับซ้อนกันหรือไม่ โดยเงื่อนไขที่ใช้คือ ถ้าตำแหน่ง x และ y ของ dog และ wave อยู่ในขอบเขตที่กำหนดไว้ และหากเงื่อนไขเหล่านี้เป็นจริงจะมีการส่งคืนค่า true ซึ่งแสดงว่ามีการชนเกิดขึ้น หากไม่เป็นจริงจะส่งคืนค่า false

```
1 package event;
2
3 import Character.*;
4
5 public class Event {
6     public static boolean checkHit(Ghost dog, Wave wave, int dogSize, int waveHeight) {
7         if (dog.x + dogSize > wave.x && dog.x < wave.x) {
8             if (dog.y + dogSize >= wave.y - waveHeight) {
9                 return true;
10            }
11        }
12        return false;
13    }
14
15    public static void gameStop(Wave[] wave, Environment[] env) {
16
17    }
18
19 }
```

2. checPoint(Ghost dog, Point point, int dogSize, int pointHeight):

คือเมธอดที่ใช้ในการตรวจสอบว่าวัตถุทั้งสองมีการชนกันหรือไม่ โดยในที่นี้เป็นการตรวจสอบว่าวัตถุ dog (ตัวละคร) และ point(เหรียญ) มีพื้นที่ทับซ้อนกันหรือไม่ โดยเงื่อนไขที่ใช้คือ ถ้าตำแหน่ง x และ y ของ dog และ point อยู่ในขอบเขตที่กำหนดไว้ และหากเงื่อนไขเหล่านี้เป็นจริงจะมีการส่งคืนค่า true ซึ่งแสดงว่ามีการชนเกิดขึ้น หากไม่เป็นจริงจะส่งคืนค่า false

```
public static boolean checPoint(Ghost dog, Point point, int dogSize, int pointHeight) {
    if (dog.x + dogSize > point.x && dog.x < point.x) {
        if (dog.y + dogSize >= point.y - pointHeight) {
            return true;
        }
    }
    return false;
}
```

3. คลาส Game ใช้งาน Event Handling ผ่านการใช้ KeyListener

เป็นการจัดการกับการกระทำที่เกิดขึ้นจากการกดปุ่มบนแป้นพิมพ์ เช่น การกระโดดของตัวละครหลักของเกม dog จะถูกจัดการผ่านเมธอด keyPressed, keyTyped, และ keyReleased ที่ได้รับการโอเวอร์ไรด์จาก KeyListener ดังนั้น ส่วนนี้จะถือเป็นการจัดการเหตุการณ์ (Event Handling) ของการกระทำในเกม หากผู้ใช้กดปุ่มบนแป้นพิมพ์ เกมจะตอบสนองตามกับการกระทำนั้นๆ ที่ได้ถูกกำหนดไว้

```
138
139 • @Override
140 public void keyPressed(KeyEvent e) {
141     if (System.currentTimeMillis() - lastPress > 600) {
142         if (e.getKeyCode() == 32 || e.getKeyCode() == 38) {
143             dog.jump(this);
144             lastPress = System.currentTimeMillis();
145         }
146     }
147 }
148
149 • @Override
150 public void keyTyped(KeyEvent e) {
151     //---
152 }
153
154 • @Override
155 public void keyReleased(KeyEvent e) {
156     //---
157 }
158
159 • public static void main(String[] arg) {
160     display = new Display();
161 }
162 }
```

2.8) อธิบายอัลกอริทึมที่สำคัญในโปรแกรม

เกมนี้เป็นเกมที่ทำให้ผู้เล่นควบคุมตัวละครซึ่งเป็นเด็กสาว โดยต้องกระโดดหลบสิ่งกีดขวางซึ่งคือแมวส้ม และมีการนับคะแนน มีอัลกอริทึมที่สำคัญได้ดังนี้

1. **การวาดหน้าจอ:** โดยใช้ Java Swing เพื่อสร้าง GUI ของเกม ทำให้ผู้ใช้เห็นตัวละคร สิ่งกีดขวาง และองค์ประกอบอื่น ๆ บนหน้าจอ
2. **การจัดการกับเหตุการณ์การกดแป้นพิมพ์:** โดยใช้ KeyListener เพื่อตรวจจับการกดปุ่มบนแป้นพิมพ์ ซึ่งในที่นี้ถูกใช้ในการทำให้ตัวละครกระโดดขึ้นและลง
3. **การคำนวณและเปลี่ยนแปลงคะแนน:** โดยเพิ่มคะแนนขึ้นทุกๆ ครั้งที่มีการชนกันของตัวละครกับเหรียญ และลดพลังชีวิตเมื่อตัวละครชนกับสิ่งกีดขวาง
4. **การจัดการการกระทำของสิ่งกีดขวางและการกระโดดหลบ:** โดยมีการตรวจสอบการชนของตัวละครกับสิ่งกีดขวาง และหยุดเกมเมื่อตัวละครไม่มีพลังชีวิตเหลือ

โดยทั้งหมดนี้สร้างประสบการณ์เล่นเกมที่น่าสนใจและท้าทาย และโค้ดที่พัฒนาขึ้นมีการนำหลักการต่าง ๆ ของการเขียนโปรแกรมแบบ OOP มาประยุกต์ใช้ รวมถึงการใช้งาน Swing GUI การจัดการกับการกระทำจากผู้ใช้ การคำนวณคะแนน สถานะของตัวละครและสิ่งกีดขวางในเกม

บทที่ 3 สรุป

3.1) ปัญหาที่พบระหว่างการพัฒนา

1. ปัญหาในการจัดการเหตุการณ์ (Event Handling) ไม่ถูกต้อง:

เกิดข้อผิดพลาดในการจัดการเหตุการณ์ต่าง ๆ ที่ส่งผลกระทบต่อพฤติกรรมของเกม เช่น การกระโดดของ ตัวละครหรือการตอบสนองต่อปุ่มกดที่ไม่ถูกต้องหรือช้าลง

2. ปัญหาการทำงานของภาพและการเคลื่อนไหว:

บางครั้งภาพอาจไม่แสดงผลได้อย่างถูกต้อง หรือการเคลื่อนไหวของวัตถุอาจไม่เหมือนกับที่คาดไว้ ซึ่งอาจเกิดจากปัญหาในการโหลดและแสดงภาพหรือการควบคุมการเคลื่อนไหวของวัตถุในเกม

3. ปัญหาเกี่ยวกับการทดสอบและตรวจสอบความถูกต้องของโปรแกรม:

การทดสอบไม่เพียงแค่ช่วยค้นพบข้อผิดพลาดและปัญหาต่าง ๆ แต่ยังช่วยให้ทราบถึงความเหมาะสมและประสิทธิภาพของโปรแกรม

3.2) จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

1. การใช้งานตัวกระโดดแบบเรียลไทม์:

โปรแกรมนี้อาศัยการปรับปรุงอัลกอริทึมการกระโดดที่ทำให้ตัวละครกระโดดได้โดยทันทีตามการกดปุ่ม โดยไม่มีความล่าช้าใด ๆ ซึ่งช่วยให้ประสบการณ์การเล่นเกมเป็นไปอย่างราบรื่นและน่าสนใจมากยิ่งขึ้น

2. ระบบการเลือกรูปแบบพื้นหลังแบบไดนามิก:

โปรแกรมนี้อาศัยระบบการเปลี่ยนแปลงรูปแบบพื้นหลังของเกมที่ขึ้นอยู่กับผลการเล่นของผู้เล่น โดยเมื่อผู้เล่นทำคะแนนมากขึ้น รูปแบบพื้นหลังจะเปลี่ยนไปตามไปด้วย ซึ่งทำให้ผู้เล่นรู้สึกถึงความท้าทายและความเปลี่ยนแปลงในการเล่น

3. กราฟิกและภาพที่น่าทึ่ง : โปรแกรมนี้มีการออกแบบกราฟิกและภาพที่น่าทึ่งที่ช่วยให้ผู้เล่นได้รับประสบการณ์การเล่นที่น่าตื่นเต้นและน่าสนใจมากยิ่งขึ้น ทำให้ติดใจและติดตามเกมไปอย่างต่อเนื่องโดยรวมแล้ว โปรแกรมนี้มีความเป็นเอกลักษณ์ที่ชัดเจนที่ช่วยให้ผู้เล่นได้รับประสบการณ์การเล่นเกมที่ไม่มีใครและน่าทึ่งอย่างแท้จริง

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย สำหรับน้องๆรุ่นต่อไป

อยากให้อาจารย์สอนช้าลงหน่อย ให้มีเวลาพักบ้าง