

# 遊戲程式設計

## 小王子之馬戲團跳火圈

指導教授：江清水

組員：

資四B 96156216 潘偉翔

資四B 96156223 賈涵宇

資四B 96156224 林玉清

資四B 96156235 葉泊佑

資四B 96156241 李俞興

程式碼：

## Game1.cs

```
14 namespace WindowsGame1
15 {
16
17     public class Game1 : Microsoft.Xna.Framework.Game
18     {
19
20         static int objs_nums = 16;
21
22         GraphicsDeviceManager graphics;
23         SpriteBatch spriteBatch;
24         Texture2D BG_Texture;
25         Texture2D[] mySpriteTexture = new Texture2D[8];
26         Texture2D hp;
27         int hp_counter = 3;
28
29         bool superstate = false;
30
31         int supertime = 0;
32
33         int r = 0;
34
35         SpriteFont Font1; // 字型物件
36
37         int lv = 1;
38
39         int distance=0;
40
41         int Offset_X = 0; // X 軸 偏移值
42
43         ClassSprite mySprite;
44
45         Rectangle[,] srcRect = new Rectangle[4, 4];
46         int pcW = 30; // 主角的寬
47         int pcH = 68; // 主角的高
48         int Dir = 2; // 走路的方向
49         int Seq = 0; // 走路的第幾個動作
50         bool superdown = false; // 主角 停止走路
51
52
53         int jump = 0; //跳躍
54         bool jumpping = false;
55         int p_now_h = 0;
56         static int p_start_h;
57         int G = 4;
58         int SG = 6;
59         bool win = false;
60         bool gameover = false;
61
62
63         ClassSprite[] objs = new ClassSprite[objs_nums];
64         bool[] objs_go = new bool[objs_nums];
65         int ball_index = 1;
66         int horse_index = 5;
67         int fire_index = 10;
68         int go_index = 0;
69
70
71         int time_counter = 0;
72
73         KeyboardState oldState;
74         double StepDuration = 0;
75
```

```

76 public Game1()
77 {
78     graphics = new GraphicsDeviceManager(this);
79     Content.RootDirectory = "Content";
80     this.IsMouseVisible = true;
81     this.Window.AllowUserResizing = true;
82     this.Window.Title = "小王子之馬戲團跳火圈(↑↓)";
83
84     for (int i = 0; i < 4; i++)
85         for (int j = 0; j < 4; j++)
86             {
87                 srcRect[i, j] = new Rectangle(j * pcW, i * pcH, pcW, pcH);
88             }
89 }
90
91
92 protected override void Initialize()
93 {
94
95     p_now_h = this.Window.ClientBounds.Height - pcH;
96     p_start_h = this.Window.ClientBounds.Height - pcH - jump;
97
98     base.Initialize();
99 }
100
101
102 protected override void LoadContent()
103 {
104
105     spriteBatch = new SpriteBatch(GraphicsDevice);
106     BG_Texture = Content.Load<Texture2D>("stage");
107
108
109
110     for (int i = 0; i < 8; i++)
111     {
112         mySpriteTexture[i] = Content.Load<Texture2D>("Prince_" + i);
113     }
114
115
116
117     mySprite = new ClassSprite(mySpriteTexture[0],
118                               new Vector2(this.Window.ClientBounds.Width / 2 - pcW / 2, this.Window.ClientBounds.Height - pcH),
119                               new Vector2(graphics.GraphicsDevice.Viewport.Width, graphics.GraphicsDevice.Viewport.Height)
120                               , 0);
121
122
123
124     Texture2D texture = Content.Load<Texture2D>("tire");
125     Texture2D texture2 = Content.Load<Texture2D>("fire");
126
127     hp = Content.Load<Texture2D>("head");
128
129     Texture2D texture3 = Content.Load<Texture2D>("horse");
130
131
132
133
134     for (int i = 0; i < objs_nums; i++)
135     {
136         if (i == 0)
137         {
138             objs[i] = new ClassSprite(hp,
139                                     new Vector2(-48, Window.ClientBounds.Height - 48),
140                                     new Vector2(graphics.GraphicsDevice.Viewport.Width, graphics.GraphicsDevice.Viewport.Height)
141                                     , 3);
142             objs[i].velocity = new Vector2(-4, 0); // 設位移速度
143         }
144         else if (i < 5)
145         {
146             //球
147             objs[i] = new ClassSprite(texture,
148                                     new Vector2(-64, Window.ClientBounds.Height - 64),
149                                     new Vector2(graphics.GraphicsDevice.Viewport.Width, graphics.GraphicsDevice.Viewport.Height)
150                                     , 1);
151             objs[i].velocity = new Vector2(-4, 0); // 設位移速度
152         }
153         else if (i < 10)
154         {
155             //馬
156             objs[i] = new ClassSprite(texture3,
157                                     new Vector2(-120, Window.ClientBounds.Height - 90),
158                                     new Vector2(graphics.GraphicsDevice.Viewport.Width, graphics.GraphicsDevice.Viewport.Height)
159                                     , 1);
160             objs[i].velocity = new Vector2(-4, 0); // 設位移速度
161         }
162         else if (i < objs_nums)
163         {
164             //火
165             objs[i] = new ClassSprite(texture2,
166                                     new Vector2(-20, Window.ClientBounds.Height - 192),
167                                     new Vector2(graphics.GraphicsDevice.Viewport.Width, graphics.GraphicsDevice.Viewport.Height)
168                                     , 2);
169             objs[i].velocity = new Vector2(-4, 0); // 設位移速度
170         }
171     }
172
173     Font1 = Content.Load<SpriteFont>("Courier New");
174 }
175
176
177 protected override void UnloadContent()
178 {
179 }
180
181

```

```

181
182 protected override void Update(GameTime gameTime)
183 {
184     if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
185         this.Exit();
186
187     KeyboardState newState; // 宣告一個 KeyboardState 結構的變數
188     newState = Keyboard.GetState(); // 得到目前鍵盤全部按鍵的狀況
189     if (newState.IsKeyDown(Keys.Escape)) this.Exit(); // 判斷Esc鍵是否已經被按下
190     else if (newState.IsKeyDown(Keys.P)) this.Initialize(); // 密技:按P障礙物消失
191     else if (newState.IsKeyDown(Keys.S))
192     {
193         if (gameover)
194             restart();
195     }
196     else if (newState.IsKeyDown(Keys.Up) /*&& Dir != 1*/)
197     {
198         Dir = 2;
199         Seq = 0;
200
201         if (p_start_h == p_now_h)
202         {
203             superdown = false;
204             jumping = true;
205             jump += G;
206         }
207     }
208     else if (newState.IsKeyDown(Keys.Down) /*&& Dir != 0*/)
209     {
210         Dir = 2;
211         Seq = 0;
212
213
214         if (p_start_h != p_now_h && p_start_h > p_now_h)
215         {
216             superdown = true;
217             jumping = false;
218             jump -= SG;
219         }
220     }
221
222     |
223
224     StepDuration += gameTime.ElapsedGameTime.TotalMilliseconds;
225     if (StepDuration > 300)
226     {
227         StepDuration = 0;
228         Seq++; // 下一張
229         Seq = Seq % 4; // 每一方向只有四張
230         mySprite.texture = mySpriteTexture[Seq];
231     }
232

```

```

233
234     oldState = newState;
235
236
237     if (p_start_h < p_now_h)
238     {
239         p_now_h = p_start_h;
240         jump = 0;
241     }
242
243
244     if (p_now_h < Window.ClientBounds.Height / 2)
245     {
246         jumpping = false;
247         if (newState.IsKeyDown(Keys.Down))
248             superdown = true;
249     }
250
251
252     if (jumpping == false && p_start_h != p_now_h)
253     {
254         if (superdown == false)
255         {
256             jump -= G;
257         }
258         else
259         {
260             jump -= SG;
261         }
262     }
263
264     if (jumpping == true)
265     {
266         jump += G;
267     }
268
269     if (gameover)
270     {
271         Offset_X = 0;
272     }
273     else
274     {
275         Offset_X -= 4;
276
277         distance = -Offset_X / 400;
278
279         if (distance % 10 == 0 && Offset_X % 400 == 0)
280             lv++;
281         if (lv == 4)
282         {
283             win = true;
284             Offset_X = 0;
285         }
286     }
287
288     int rr = 0;
289
290
291     Random rnd = new Random();
292     rr = rnd.Next(90, 550);
293

```

```

294 |
295 |     if (time_counter >= rr)
296 |     {
297 |
298 |         {
299 |
300 |             switch (lv)
301 |             {
302 |                 case 1:
303 |
304 |                     //生命加此
305 |                     r = rnd.Next(1, 50);
306 |                     if (r % 26 == 0)
307 |                     {
308 |                         //生命加此
309 |                         objs[0].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 48);
310 |
311 |                     }
312 |                     else
313 |                     {
314 |
315 |
316 |                         ball_index++;
317 |
318 |
319 |                         if (ball_index > 4)
320 |                         {
321 |                             ball_index = 1;
322 |                         }
323 |                         objs[ball_index].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 64);
324 |                         go_index = ball_index;
325 |                     }
326 |                     break;
327 |                 case 2:
328 |                     r = rnd.Next(1, 50);
329 |                     if (r % 26 == 0)
330 |                     {
331 |                         //生命加此
332 |                         objs[0].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 48);
333 |
334 |                     }
335 |                     else
336 |                     {
337 |                         r = rnd.Next(1, 5);
338 |
339 |
340 |                         if (r == 1 || r == 4)
341 |                         {
342 |                             ball_index++;
343 |
344 |                             if (ball_index > 4)
345 |                             {
346 |                                 ball_index = 1;
347 |                             }
348 |
349 |                             objs[ball_index].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 64);
350 |                             go_index = ball_index;

```

```

351         else if (r == 2 || r == 5)
352         {
353             horse_index++;
354             if (horse_index > 9)
355             {
356                 horse_index = 5;
357             }
358             objs[horse_index].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 90);
359             go_index = horse_index;
360         }
361         else
362         {
363         }
364     }
365     break;
366
367
368
369 case 3:
370     r = rnd.Next(1, 50);
371     if (r % 26 == 0)
372     {
373         //生命加此
374         objs[0].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 48);
375     }
376     else
377     {
378         r = rnd.Next(1, 10);
379
380
381         if (r == 1 || r == 6)
382         {
383             ball_index++;
384
385             if (ball_index > 4)
386             {
387                 ball_index = 1;
388             }
389             objs[ball_index].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 64);
390             go_index = ball_index;
391         }
392         else if (r == 2 || r == 5 || r == 9)
393         {
394             horse_index++;
395             if (horse_index > 9)
396             {
397                 horse_index = 5;
398             }
399             objs[horse_index].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height - 90);
400             go_index = horse_index;
401         }
402         else if (r == 3 || r == 7 || r == 8)
403         {
404             fire_index++;
405             if (fire_index > objs_nums)
406             {
407                 fire_index = 10;
408             }
409             objs[fire_index].position = new Vector2(Window.ClientBounds.Width, graphics.GraphicsDevice.Viewport.Height / 2 - 192 / 3);
410             go_index = fire_index;
411         }

```

```

412         else
413         {
414         }
415     }
416     break;
417 default:
418     win = true;
419     break;
420
421 }
422 }
423
424
425     time_counter = 0;
426 }
427
428
429 for (int i = 0; i < objs_nums; i++)
430 {
431     {
432         objs[i].Move(); // 移動精靈圖物件
433
434         if (superstate)
435         {
436             supertime++;
437             if (supertime > 1000)
438             {
439                 supertime = 0;
440                 superstate = false;
441             }
442         }
443     }
444     else
445     {
446         if (mySprite.Collides(objs[i]))
447         {
448             //碰撞動作!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
449             if (objs[i].obj_type == 3)
450             {
451                 hp_counter++;
452                 objs[0].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height);
453             }
454             else
455                 hp_counter--;
456             superstate = true;
457             supertime++;
458             if (hp_counter == 0)
459             {
460                 Offset_X = 0;
461                 gameover = true;
462                 objs[i].position = new Vector2(Window.ClientBounds.Width, Window.ClientBounds.Height);
463             }
464         }
465     }
466 }
467
468 }
469 }
470
471
472     time_counter++;
473     base.Update(gameTime);
474 }

```



```

475
476 protected override void Draw(GameTime gameTime)
477 {
478     graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
479
480
481     int W = graphics.GraphicsDevice.Viewport.Width;
482     int H = graphics.GraphicsDevice.Viewport.Height;
483
484     spriteBatch.Begin(SpriteBlendMode.AlphaBlend);
485
486     Rectangle DesRect = new Rectangle();
487
488     if (gameover)
489     {
490         for (int i = -1; i <= W / BG_Texture.Width + 1; i++) // i 是 X 方向 要貼幾次
491         {
492             for (int j = -1; j <= H / BG_Texture.Height + 1; j++) // j 是 Y 方向 要貼幾次
493             {
494                 Vector2 position = new Vector2(
495                     i * BG_Texture.Width + (Offset_X % BG_Texture.Width),
496                     j * BG_Texture.Height); // 算出要 貼上的位置
497
498                 spriteBatch.Draw(BG_Texture, position, Color.White);
499             }
500         }
501
502         // 字串位置
503         Vector2 FontPos2 = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2,
504             0);
505
506         Vector2 FontOrigin2 = Font1.MeasureString("") / 2; // 字串中心
507
508         spriteBatch.DrawString(Font1, // 字型
509             "Lv:" + lv.ToString(),
510             FontPos2, // 位置
511             Color.Red, // 字的顏色
512             0, // 旋轉角度
513             FontOrigin2, // 字串中心點
514             3.0f, // 縮放倍數
515             SpriteEffects.None, // 旋轉校果
516             0); // 圖層深度 0.0 ~ 1.0 (後)
517
518
519         // 字串位置
520         Vector2 FontPos = new Vector2(6 * graphics.GraphicsDevice.Viewport.Width / 7,
521             0);
522
523         Vector2 FontOrigin = Font1.MeasureString("") / 2; // 字串中心
524
525         spriteBatch.DrawString(Font1, // 字型
526             distance.ToString() + "M",
527             FontPos, // 位置
528             Color.White, // 字的顏色
529             0, // 旋轉角度
530             FontOrigin, // 字串中心點
531             3.0f, // 縮放倍數
532             SpriteEffects.None, // 旋轉校果
533             0); // 圖層深度 0.0 ~ 1.0 (後)
534

```

```

535 // 字串位置
536 Vector2 Fontover = new Vector2(1 * graphics.GraphicsDevice.Viewport.Width / 5,
537 graphics.GraphicsDevice.Viewport.Height / 4);
538
539 Vector2 FontOrigin_over = Font1.MeasureString("") / 2; // 字串中心
540
541 spriteBatch.DrawString(Font1, // 字型
542 "GameOver",
543 Fontover, // 位置
544 Color.Black, // 字的顏色
545 0, // 旋轉角度
546 FontOrigin_over, // 字串中心點
547 6.Of, // 縮放倍數
548 SpriteEffects.None, // 旋轉校果
549 0); // 圖層深度 0.0 ~ 1.0 (後)
550
551 Vector2 Fontover2 = new Vector2(1 * graphics.GraphicsDevice.Viewport.Width / 3,
552 2 * graphics.GraphicsDevice.Viewport.Height / 4);
553 Vector2 FontOrigin_over2 = Font1.MeasureString("") / 2; // 字串中心
554 spriteBatch.DrawString(Font1, // 字型
555 "Press 'S' to continuous.....",
556 Fontover2, // 位置
557 Color.White, // 字的顏色
558 0, // 旋轉角度
559 FontOrigin_over2, // 字串中心點
560 1.Of, // 縮放倍數
561 SpriteEffects.None, // 旋轉校果
562 0); // 圖層深度 0.0 ~ 1.0 (後)
563
564 }
565 else if (win)
566 {
567     if (lv == 4)
568     {
569         for (int i = -1; i <= W / BG_Texture.Width + 1; i++) // i 是 X 方向 要貼幾次
570         {
571             for (int j = -1; j <= H / BG_Texture.Height + 1; j++) // j 是 Y 方向 要貼幾次
572             {
573                 Vector2 position = new Vector2(
574                     i * BG_Texture.Width + (Offset.X % BG_Texture.Width),
575                     j * BG_Texture.Height); // 算出要 貼上的位置
576
577                 spriteBatch.Draw(BG_Texture, position, Color.White);
578             }
579         }
580
581         mySprite.position = new Vector2(this.Window.ClientBounds.Width / 2 - pcW / 2,
582             this.Window.ClientBounds.Height - pcH - jump);
583
584         DesRect = new Rectangle(
585             this.Window.ClientBounds.Width / 2 - pcW / 2,
586             this.Window.ClientBounds.Height - pcH - jump,
587             pcW, pcH);
588
589         p_now_h = this.Window.ClientBounds.Height - pcH - jump;
590
591         spriteBatch.Draw(objs[0].texture, objs[0].position, Color.White);
592
593

```

```

594 // 字串位置
595 Vector2 Fontwin = new Vector2(1 * graphics.GraphicsDevice.Viewport.Width / 4,
596 graphics.GraphicsDevice.Viewport.Height / 2);
597
598 Vector2 FontOrigin_win = Font1.MeasureString("") / 2; // 字串中心
599
600 spriteBatch.DrawString(Font1, // 字型
601 "Winner",
602 Fontwin, // 位置
603 Color.Black, // 字的顏色
604 0, // 旋轉角度
605 FontOrigin_win, // 字串中心點
606 6.Of, // 縮放倍數
607 SpriteEffects.None, // 旋轉效果
608 0); // 圖層深度 0.0 ~ 1.0 (後)
609
610 Vector2 FontPos2 = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2,
611 0);
612
613 }
614
615
616 }
617 else
618 {
619     for (int i = -1; i <= W / BG_Texture.Width + 1; i++) // i 是 X 方向 要貼幾次
620     {
621         for (int j = -1; j <= H / BG_Texture.Height + 1; j++) // j 是 Y 方向 要貼幾次
622         {
623             Vector2 position = new Vector2(
624                 i * BG_Texture.Width + (Offset_X % BG_Texture.Width),
625                 j * BG_Texture.Height); // 算出要 貼上的位置
626
627             spriteBatch.Draw(BG_Texture, position, Color.White);
628         }
629     }
630
631
632
633     mySprite.position = new Vector2(this.Window.ClientBounds.Width / 2 - pcW / 2,
634 this.Window.ClientBounds.Height - pcH - jump);
635
636
637
638     DesRect = new Rectangle(
639         this.Window.ClientBounds.Width / 2 - pcW / 2,
640         this.Window.ClientBounds.Height - pcH - jump,
641         pcW, pcH);
642
643     p_now_h = this.Window.ClientBounds.Height - pcH - jump;
644
645
646     spriteBatch.Draw(objs[0].texture, objs[0].position, Color.White);
647

```

```

647
648 for (int i = 0; i < objs_nums; i++)
649 {
650
651     if (i <= fire_index)
652     {
653         spriteBatch.Draw(objs[i].texture, objs[i].position, Color.White);
654     }
655 }
656
657 spriteBatch.Draw(mySprite.texture,
658                 DesRect,
659                 Color.White);
660
661 for (int k = 0; k < hp_counter; k++)
662 {
663     Vector2 pos = new Vector2(k * 50, 0);
664     spriteBatch.Draw(hp, pos, Color.White);
665 }
666
667 // 字串位置
668 Vector2 FontPos2 = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2,
669                                0);
670
671 Vector2 FontOrigin2 = Font1.MeasureString("") / 2; // 字串中心
672
673 spriteBatch.DrawString(Font1, // 字型
674                        "Lv:" + lv.ToString(),
675                        FontPos2, // 位置
676                        Color.Red, // 字的顏色
677                        0, // 旋轉角度
678                        FontOrigin2, // 字串中心點
679                        3.Of, // 縮放倍數
680                        SpriteEffects.None, // 旋轉校果
681                        0); // 圖層深度 0.0 ~ 1.0 (後)
682
683 // 字串位置
684 Vector2 FontPos = new Vector2(6 * graphics.GraphicsDevice.Viewport.Width / 7,
685                                0);
686
687 Vector2 FontOrigin = Font1.MeasureString("") / 2; // 字串中心
688
689 spriteBatch.DrawString(Font1, // 字型
690                        distance.ToString() + "M",
691                        FontPos, // 位置
692                        Color.White, // 字的顏色
693                        0, // 旋轉角度
694                        FontOrigin, // 字串中心點
695                        3.Of, // 縮放倍數
696                        SpriteEffects.None, // 旋轉校果
697                        0); // 圖層深度 0.0 ~ 1.0 (後)
698 }
699
700
701 spriteBatch.End();
702
703 base.Draw(gameTime);
704 }
705
706 public void restart()
707 {
708     hp_counter = 3;
709     superstate = false;
710     supertime = 0;
711     r = 0;
712     lv = 1;
713     distance = 0;
714     Offset_X = 0;
715     pcW = 30; // 主角的寬
716     pcH = 68; // 主角的高
717     Dir = 2;
718     Seq = 0; // 走路的第幾個動作
719     superdown = false; // 主角 停止走路
720     jump = 0; // 跳躍
721     jumping = false;
722     G = 4;
723     SG = 6;
724     win = false;
725     gameover = false;
726     p_now_h = this.Window.ClientBounds.Height - pcH;
727     p_start_h = this.Window.ClientBounds.Height - pcH - jump;
728     base.Initialize();
729 }
730 }
731 }
732 }
733 }

```

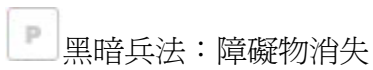
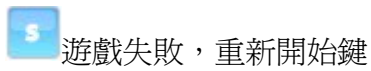
## ClassSprite.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using Microsoft.Xna.Framework;
5 using Microsoft.Xna.Framework.Graphics;
6
7 namespace WindowsGame1
8 {
9     class ClassSprite
10    {
11        public Texture2D texture; // 2D 紋理圖
12        public Vector2 position; // 2D 紋理圖 的位置
13        private Vector2 screenSize; // 視窗寬高
14        public Vector2 velocity = Vector2.Zero; // 2D 紋理圖 的位移速度
15        public int obj_type = 0;
16
17        public ClassSprite(Texture2D texture, Vector2 position, Vector2 screenSize,int type)
18        {
19            this.texture = texture;
20            this.position = position;
21            this.screenSize = screenSize;
22            this.obj_type = type; // 小王子1球拒馬2火圈
23        }
24
25        // 移動
26        public void Move()
27        {
28
29            position += velocity;
30        }
31
32        public bool Collides(ClassSprite other)
33        {
34
35            if (this.obj_type == 0 && other.obj_type == 1)
36            {
37
38                // 檢查是否 碰撞
39                return (this.position.X + this.texture.Width > other.position.X &&
40                    this.position.X < other.position.X + other.texture.Width &&
41                    this.position.Y + this.texture.Height > other.position.Y &&
42                    this.position.Y < other.position.Y + other.texture.Height);
43            }
44            else if (this.obj_type == 0 && other.obj_type == 2)
45            {
46                return (this.position.X + this.texture.Width > other.position.X &&
47                    this.position.X < other.position.X + other.texture.Width &&
48                    ((this.position.Y > other.position.Y && this.position.Y < other.position.Y + 10) ||
49                    (this.position.Y + this.texture.Height < other.position.Y + other.texture.Height &&
50                    this.position.Y + this.texture.Height > other.position.Y + other.texture.Height - 202)
51
52                ));
53            }
54            else if (this.obj_type == 0 && other.obj_type == 3)
55            {
56                return (this.position.X + this.texture.Width > other.position.X &&
57                    this.position.X < other.position.X + other.texture.Width &&
58                    this.position.Y + this.texture.Height > other.position.Y &&
59                    this.position.Y < other.position.Y + other.texture.Height);
60            }
61
62            return false;
63        }
64    }
65 }
66
```

### 遊戲背景介紹：

有一天，小王子被壞人追趕跑進了馬戲團，但馬戲團到處有輪胎、拒馬和火圈等等障礙物...  
請幫助小王子順利越過這些障礙物吧~~!

### 遊戲操作說明：



### 遊戲介面說明：



## 障礙物介紹：

輪胎	拒馬	火圈	補血
			

## 如何開始：

在執行檔上雙擊左鍵，即開始遊戲。

## 遊戲介紹：

《小王子之馬戲團跳火圈》畫面雖然稱不上華麗，卻具有很高的趣味性。在遊戲中你需要操縱小王子，通過三個 關卡的考驗。這些關卡都是我們在馬戲團裡耳熟能詳的躍過障礙物，比如輪胎、拒馬、火圈等等。當然遊戲肯定不會像你真地去跳火圈那麼難，平易近人但卻非常到位的操作為 它贏得了很多玩家青睞。遊戲真正要磨練的是你的手指，通過對上下鍵跳躍的精確掌控，你將很快熟悉動作遊戲的套路。

## 遊戲目標：

闖過所有的關卡。

## 遊戲設計心得：

96156216 潘偉翔

參與這次遊戲設計後，讓我了解到XNA的程式設計主要分為6個區塊Game1()、Initialize()、LoadContent()、UnloadContent()、Update()、Draw()，而比較困難的地方，就在於Update()程式邏輯的寫法以及Draw()匯出圖案。起初，我們常常把要做的動作事情寫在Draw()，發現到我們根本搞錯地方了，原來Draw()只需要管印出圖片的部分，其餘邏輯動作都在Update()。

原本以為設計遊戲很難，在完成這支遊戲後，發現到原來寫一支超復古小遊戲是這麼的容易，實際上只是把課堂上所學到的範例拿來運用，如此而已。不過，離現今市面上出售的遊戲還有好長一段距離！

96156223 賈涵宇

從老師上課示範的程式，了解到每個元件再做什麼，到自己寫程式，創作遊戲程式，沒有想像中的困難，或許是還沒使用到更艱深元件，感覺老師在電腦教室授課的內容對實作比較有幫助。最後體會到每個程式語言都有其目的，以XNA來說，他真的是專門寫遊戲語言，還記得之前修過VB，也曾經寫過小遊戲，可是卻不能表達我想要的，比較沒有那麼流暢，不過VB在操作上比XNA容易上手。

96156224 林玉清

雖然這次不是第一次接觸遊戲設計，但是卻有全新的體驗，之前在V B課時寫的小遊戲，主要還是著重在邏輯運算的方面，這次學了X N A之後，還必須考慮圖形物件的相對配置，才發現在這裡運用到這麼多之前所學過的數學，有時候動作邏輯是對的，但是D r a w出來之後卻是很好笑的動作，撰寫程式的過程中覺得相當有趣。透過上課的範例交互運用，最後完成了屬於自己的遊戲，感覺蠻有成就感的。

96156235 葉泊佑

在這門課學到遊戲的程式架構是如何運行，Update和Draw每秒重複執行好幾次，也顯得電腦運算能力的重要，在以往的程式設計只是按個按鈕，執行結果一下就出來，遊戲因為每秒都draw好幾次，比較能深刻感受到電腦運算能力。另外也學到了如何讓圖產生一連串的動作，在還沒接觸這門課前，一直認為圖片產生動作需要有複雜的數學演算法才寫的出來，學過這門課後，其實用到的數學並不會太難，都只是以前高中的數學。

96156241 李俞興

以前在寫VB或C#時已經了解遊戲的進行規則，就是一堆按鈕按來按去，function換來換去而已，在這門課裡面我讓我更明白了以往吸引我的遊戲的畫面，原來是一堆圖不斷的重新繪圖然後出現，數學的部份讓我知道如何去變換圖片顯示位置。

XNA的程式範例雖然內容很簡單，但操縱起來還是會有以前玩遊戲時的興奮感，特別是開學後的前幾個禮拜，再利用練習題寫出的程式，更加明白遊戲的製作原來可以這麼簡單，只是這堂課沒有教模組的製作。在講解遊戲概念的課堂上，讓我了解遊戲製作的過程以及人員安排等等，不過我發覺的是這概念不只單單只能放在遊戲上，在其他的產品或者計劃，都可以用得上。