

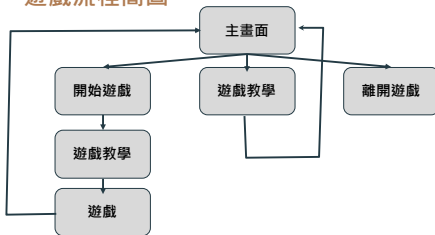
奔跑吧! Master

成圖 final project 第七組
4105053001 黃謙儒
4105053102 唐健樞

分工表

| 黃謙儒: 50% | 唐健樞: 50% |
|-------------|----------|
| 碰撞 | 載入模組、貼圖 |
| 主畫面、教學模式、選單 | 物理引擎 |
| 跨欄、岩漿陷阱 | 加速點 |
| 地圖設計 | 終點動畫 |

遊戲流程簡圖



遊戲說明



這是一款障礙物跑步遊戲。玩家扮演Master進行移動。角色使用WASD移動，P鍵衝刺，空白鍵跳躍。落地前再按一次空白可以進行二段跳。左上角有角色的氣力條，衝刺時消耗，若氣條全空，角色便會因力竭而跌倒。



角色必須一邊跨過障礙物（跨欄、岩漿），一邊收集聖晶石。收集滿30顆方能開啟最後的門，抵達終點。若沒有收集齊30顆就不能進行十連召喚。若沒有收集齊30顆，碰到終點的門就會回到起始點，必須重新跨過障礙物。

遊戲設計: 衝天跑玩法 + FGO外型



+



衝天跑影片: <https://www.youtube.com/watch?v=Y1kNvK29Gc>

OBJ資料結構

以struct紀錄每個物體的三維

座標，以及其bonding box
另有void set_xyz的function

參數填入座標及其長寬高
就能計算其bonding box

角色例外，角色只設定其xyz座標

```

struct obj {
    bool isappear = true;
    float x, y, z;
    float MaxX, MaxY, MaxZ;
    float MinX, MinY, MinZ;
    float length, width, height; //長寬高
};
  
```

Loading model : glm

```

127 GLModel *gla_model;
128
129 gla_model = glmLoadOBJ("res/sgmodel.obj");
130 glInit(gla_model);
131 glScalef(gla_model, 5, 0);
132 glTranslatef(gla_model, 0, 0);
133 glTranslatef(gla_model, 90);
134 sgmodel_id = glInit(gla_model, GL_DEPTH | GL_WATER | GL);
135 glDelete(gla_model);
136

```

obj model 直接用glm的函式庫即可繪製

Loading texture : SOIL

```

127 void *soil;
128
129 soil = soilLoadTexture("res/soil.png", SOIL_LOAD_RGBA, SOIL_TEXTURE_2D);
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
1000

```

先loading 含有alpha值的貼圖(去路圖)。

SOIL_LOAD_RGBA 參數可以以RGBA格式讀取bitmap。

在display function 中 開啟GL_BLEND。
GL_TEXTURE2D

繪出設好alpha值的texture;

```

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
1000

```

▲ 圖片為繪製透明特效用的code

碰撞判定：牆壁-col(角色,牆壁)

角色視為單一點座標 (x, y, z)

我們的碰撞判定分為兩種: 牆壁的判定與 物體(障礙物、加速點)的判定

牆壁的判定：void col (角色, 牆壁, open);

角色的點座標必須在box內，即 $Wall_MaxX \geq x \geq Wall_MinX$ · Y, Z 依此類推
若超出某一方(x/y/z)的範圍，則設角色的 x/y/z 座標為牆壁的 Min/Max



必須待在box內

碰撞判定：物體 - obj_col

物體的判定用於：加速點、障礙物、岩漿、判斷角色是否落入區域內

物體的判定：bool obj_col (角色, 物體);

當角色的點座標在box內，即 $Wall_MaxX \geq x \geq Wall_MinX$ 時，
obj_col function 會回傳True · Y, Z 同樣

根據function回傳值，我們可以知道此角色與物體是否發生碰撞，再根據各情況作出回應



當物體在box內
col() function回傳 True

Keyoperation

#define up 0 #define down 1
int keystates[256] = {up};

用keyboard () 與 keypad () 改變array的
值用以判斷該按鍵是否為「按下」的狀態

在每次display的一開頭，呼叫
keyoperation ()，檢查是否按鍵按下，
並更新角色的位置資料。

在角色倒下、落入岩漿時
role_down>0, drop_down>0
兩變數等於0時，按下的按鍵才有作用

```

void keypad(unsigned char key, int x, int y)
{
    keystates[key] = up;
}

void keyboard(unsigned char key, int x, int y)
{
    keystates[key] = down;
}

void keyoperation(void)
{
    if (mode == 3) { //遊戲模式
        if (role_down == 0 && drop_down == 0) { //沒有倒下狀態
            if (keystates['W'] == down) || (keystates['w'] == down)
            {
                charactermoving = true;
                rolefacing = 0;
                role.x = move;
                glTranslatef(0, 0, 0);
            }
        }
    }
}

```

Keyoperation

移動完畢，呼叫偵測碰撞的function
(分別為 col 與 obj_col)

用一個一個判斷角色是否有和該
物體產生碰撞，再各別執行相對應
的動作。

障礙、岩漿、加速點、終點線、
聖晶石皆是呼叫obj_col

```

col(role, Wall, open);
for (int i = 0; i < HurdleNumber; i++) {
    if (Hurdle_down[i] == 0 && obj_col(role, Hurdle[i])) {
        if (i == 0) { role_down = role_time; }
        Hurdle_down[i] = Hurdle_time;
        PlaySound(TEXT("res/sound/drown.wav"), NULL, SND_ASYNC);
    }
}

for (int i = 0; i < acceleratenum; i++) {
    if (obj_col(role, accelerat[i]))
    {
        if (CHARACTER_BURST == 0)
        {
            CHARACTER_BURST = 1;
            burstcount = 0;
        }
    }
}

```

物理引擎：跳躍

CHARACTER_ACTION紀錄使用者目前的跳躍是單段跳，還是按下第二次的二段跳
套用 $v=v_0+gt$ 改變角色位置
設定一次跳躍為20幀，
兩次最高40幀，結束後剛好
回到地面上

```
void characterJump()
{
    if (CHARACTER_ACTION == 1 || CHARACTER_ACTION == 2)
    {
        role.y += jumpH - gravity*jumpCnt;
        jumpCnt++;
        if (jumpCnt == 20 * CHARACTER_ACTION || role_down)
        {
            CHARACTER_ACTION = 0;
            role.y = 0.0;
        }
    }
}
```

TimerFunction

glutTimerFunc(30, TimerFunction, 1)
使TimerFunction每30毫秒就會執行一次
當角色跌倒，role_down = 50
利用TimerFunction()
當role_down > 0時，每次執行減1，
就能控制跌倒時間為15秒
跌落岩漿同樣，跌落時關閉人物與牆壁的碰撞
讓角色等速沒入岩漿之中，
要結束前讓角色回到地面，開啟牆壁碰撞

最終結束的動畫也是藉由調動TimerFunction，
來完成動畫的效果

```
void TimerFunction(int value)
{
    if (mode == 3 || mode == 35) {
        if (role_down > 0) {
            role_down--;
        }
        if (drop_down > 1) {
            open = 0;
            drop_down--;
            DROP = DROP + 0.4;
        }
        if (drop_down == 1) {
            drop_down--;
            role.x = (Drop[drop_index]).MaxZ + 10;
            DROP = 0;
            rolefaceangle = 0.0;
            role.y = 0; //回到地面
            role_down = Role_Time; //到地面
            open = 1; //關閉
        }
    }
}
```

資源：音效

- 選單 — 
- 確認 — 
- 人物跌倒 — 
- 掉進岩漿 — 

來源:<https://zh-tw.soundeffect-lab.info/sound/battle/battle2.html>

資源：貼圖

- <https://www.pinterest.com/chenkelvin520/hand-paint-texture/> 牆壁們
- <https://forums.sufficientvelocity.com/threads/fate-layer-zero-terminus-archive.43346/> 召喚陣
- <https://zh.pngtree.com/30%E9%96%B3%E5%85%B9> 特效
- <https://fate-go.fandom.com/zh/wiki/%E8%B1%96%E6%99%B6%E7%9F%B3?variant=zh-tw> 石頭

