

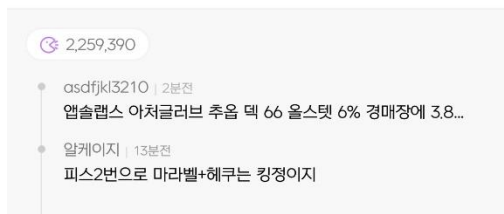
## 1. 개요

### 요구조건 1단계


구현사항		구현여부
Canvas 그리기	2D 컨텍스트 정의	○
	직선, 원, (사)다각형	○
	베지어곡선, 텍스트, 이미지	○
JavaScript이벤트핸들러	화면 전환지원	○
	입력방식설명	○
객체지향프로그래밍	하나 이상의 객체 정의	○
	각 객체는 하나의 UI요소대응	△
	해당 객체 최소 2개이상 생성	○




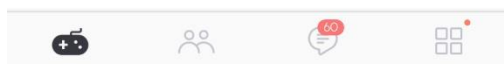
내가 즐겨하는 게임  
**메이플스토리**  
3주 연속 접속! 다음주에 또 만나요~



내가 플레이한 게임

 메이플스토리 2021.04.06 21:06

 서든어택 2021.03.11 17:05



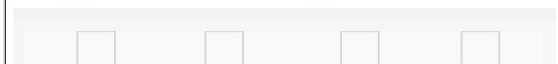
내가 즐겨하는 게임  
**메이플스토리**  
3주 연속 접속! 다음주에 또 만나요~

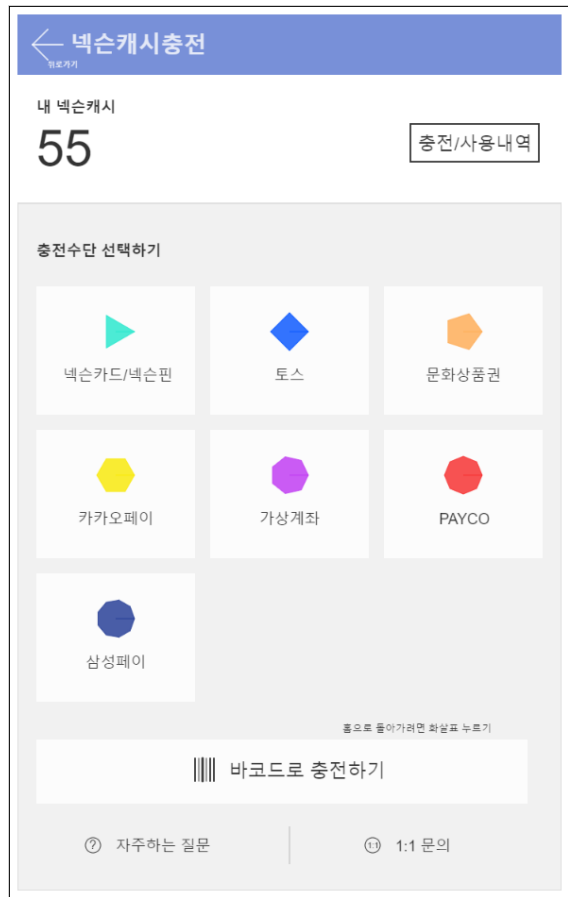
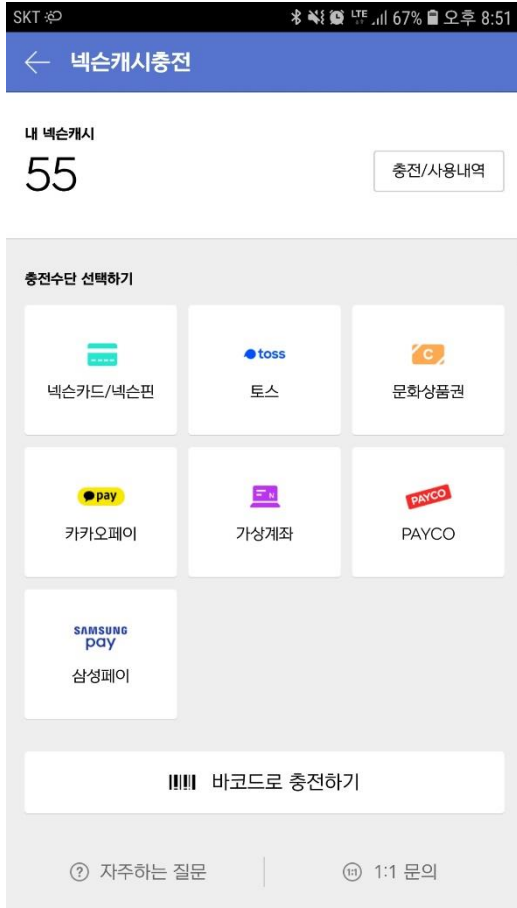


내가 플레이한 게임

 메이플스토리 2021.04.06 21:06

 서든어택 2021.03.11 17:05





## <소개>

- 앱 특징
  - 다채로운 색감
  - 간편하면서도 여러 기능으로 이동가능
- 앱 선정한 이유
  - 다양한 모양의 요소가 많이 들어감.
  - 직관적이어서 기능이용이 편함.
- 전반적인 모사 방안
  - 우선 canvas요소들을 전부 그려놓고 메인화면, 충전화면, OTP입력화면 3개화면 사이를 이동할 수 있도록함.

## <Canvas 그리기>

- 전체적인 HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>2017204021</title>

  <style>
    body{
      background: #dddddd;
    }
    #canvas {
      margin: 10px;
      padding: 10px;
      background: #ffffff;
      border: thin inset #000000;
    }
  </style>
</head>
<body>
  <canvas id="canvas" width="720" height="1160">
    Canvas not supported
  </canvas>

  <script src="main.js" >
  </script>
</body>
</html>
```

- 텍스트 그리기

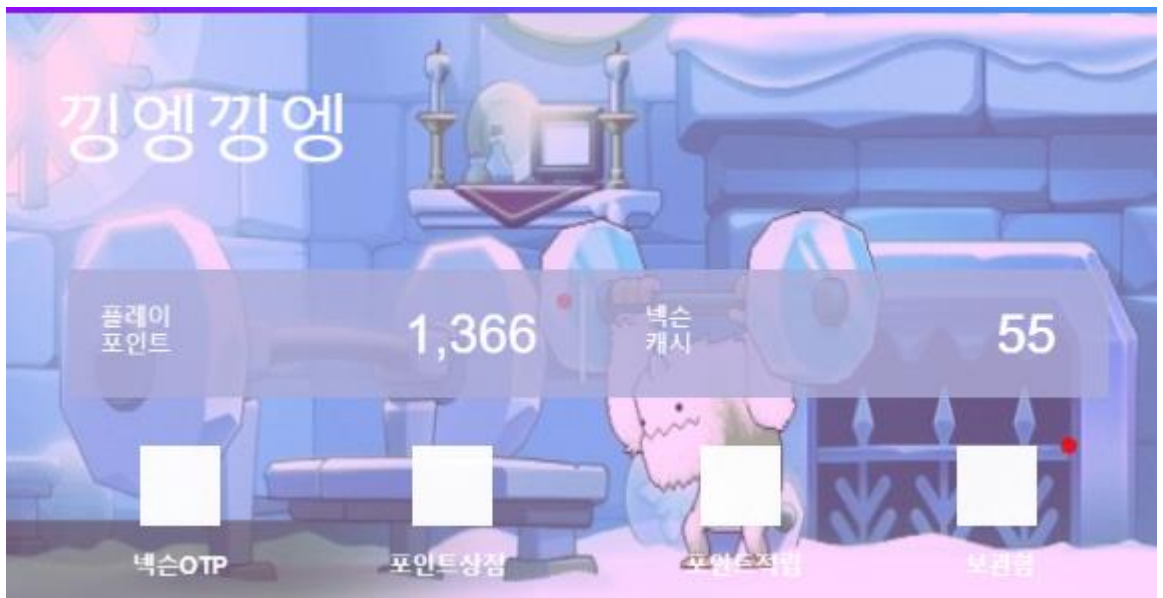
```
var nexontitle = document.getElementById("canvas"),
    nexontitle = canvas.getContext("2d");
nexontitle.font = "bold 30px Helvetica";
nexontitle.fillStyle = "white";
nexontitle.fillText("NEXON PLAY", 260, 50);
```



– font를 이용한 글꼴설정, fillstyle을 이용한 색설정 filltext를 이용한 좌표에 텍스트입력

- 이미지 그리기

```
var background1 = document.getElementById("canvas"),
    context = canvas.getContext("2d"),
    image = new Image();
context.globalAlpha = 0.8;
image.src = "메티큐티.jpg";
image.onload = function(e) {
    context.drawImage(image, 0, 80, 720, 370);
};
```



– 비슷한 이미지를 img.src, onload로 불러온뒤 context.globalAlpha를 이용해 투명도 적용, drawImage를 이용해 좌표에 설정 너비높이로 입력

- 직선 그리기

```
var centerline = document.getElementById("canvas"),
    context = canvas.getContext("2d");
context.beginPath();
context.moveTo(45, 690);
context.lineWidth = 30;
context.lineCap = "round";
context.lineTo(180, 690);
context.strokeStyle = "black";
context.stroke();
context.closePath();
```



- lineWidth를 통해 굵기 설정, lineCap을 이용한 round 설정
- moveTo를 이용한 시작점 설정,.lineTo를 이용한 경로 설정

- 사각형 그리기

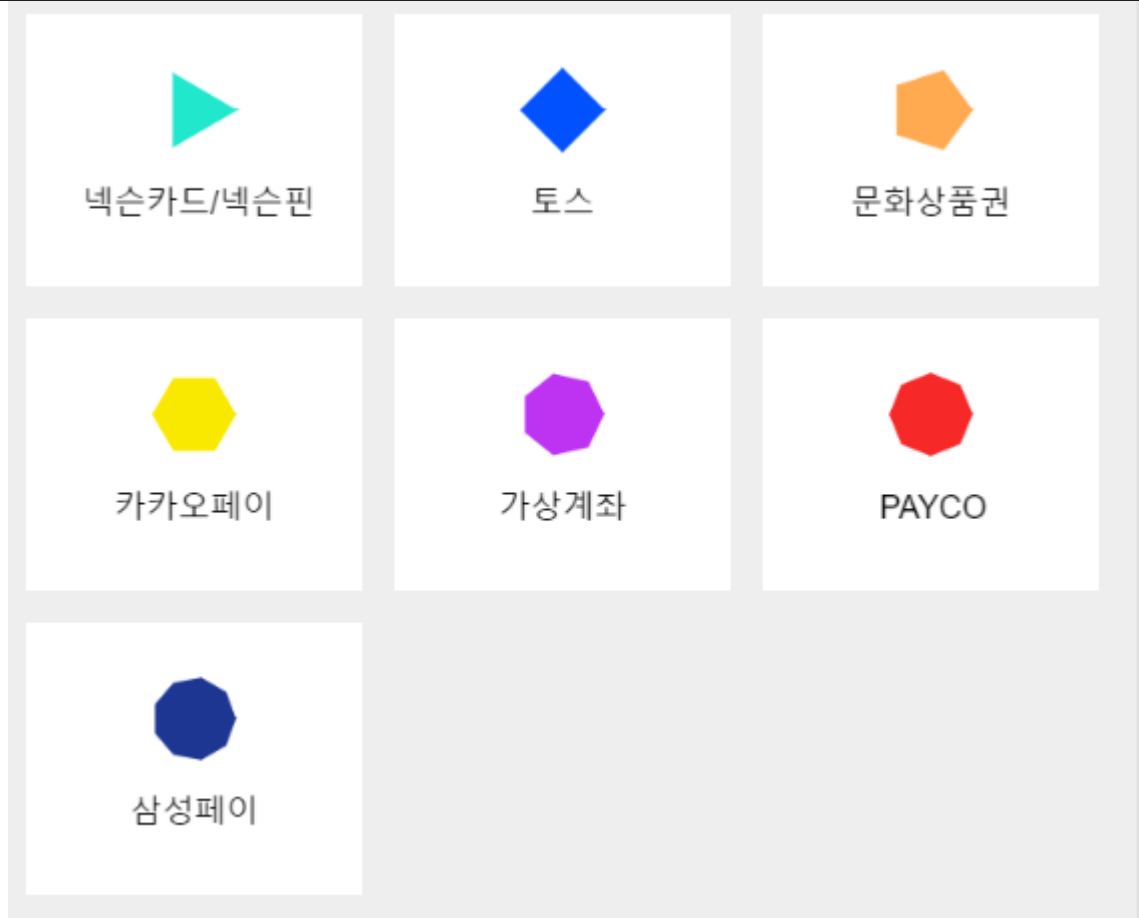
```
var titlebackground = document.getElementById("canvas"),
    context = canvas.getContext("2d");
var lingrad = context.createLinearGradient(0, 0, 720, 80);
lingrad.addColorStop(0, "#9106F9");
lingrad.addColorStop(0.9, "#4493F8");
context.fillStyle = lingrad;
context.fillRect(0, 0, 720, 80);
```



- fillRect를 이용한 색칠된 사각형 그리기 (테두리만 그릴 땐 strokeRect 이용)
- 추가적으로 lingrad를 이용한 그라데이션 설정.

- 다각형 그리기

```
var submenu3icon = document.getElementById("canvas"),
    submenu3icon = canvas.getContext("2d");
submenu3icon.beginPath();
submenu3icon.moveTo(590,420);
submenu3icon.lineWidth = 2;
for(var i=0; i<=5; i++){
    submenu3icon.lineTo(590 + 25*Math.cos((Math.PI*2)/5*i), 420 + 25*Math.sin((Math.PI*2)/5*i));
}
submenu3icon.strokeStyle = "#FFAA50";
submenu3icon.stroke();
submenu3icon.fillStyle = "#FFAA50";
submenu3icon.fill();
submenu3icon.closePath();
```



- for문과 수학적 원의 성질을 이용한 n각형 그리기
- for(var i=0; i<=n; i++){
 

Submenuicon.lineTo(가운데x좌표 + 반지름\*Math.cos((Math.PI\*2)/n\*i), 가운데y좌표 + 반지름\*Math.sin((Math.PI\*2)/n\*i));
- 이후에 객체지향 프로그래밍 부분에서 사용

- 베지어곡선 그리기

```
function drwaBezierCurve() {
    context.strokeStyle = "white";
    context.fillStyle = "white";
    context.beginPath();
    context.moveTo(endPoints[0].x, endPoints[0].y);
    context.bezierCurveTo(controlPoints[0].x, controlPoints[0].y,
        controlPoints[1].x, controlPoints[1].y,
        endPoints[1].x, endPoints[1].y);
    context.stroke();
}
```

```
var titleicon1_1 = document.getElementById("canvas"),
    context = canvas.getContext("2d"),
    endPoints = [ { x: 60, y: 20}, { x: 20, y: 40}],
    controlPoints = [ { x: 60, y: 80}, { x: 30, y: 10}];

drwaBezierCurve();
```



- drawBezierCurve()로 베지어곡선 함수를 정의하고 연결점과 커브기준점을 만들어넣음.

- 원 그리기

```
var questionicon1 = document.getElementById("canvas"),
    context = canvas.getContext("2d");
context.lineWidth = 1;
context.beginPath();
context.arc(100, 1100, 10, 0, Math.PI*2);
context.closePath();
context.stroke();
```



- arc함수를 이용한 원그리기
- (중심x좌표, 중심y좌표, 반지름, 시작각도, 끝각도)
- 원 안의 "?"는 텍스트입력을 이용한 입력

## <JavaScript 객체 지향 프로그래밍>

```
var drawingdagak = function(x, y, r, n, fillcolor, strokecolor){
    this.x = x;
    this.y = y;
    this.r = r;
    this.n = n;
    this.fillcolor = fillcolor;
    this.strokecolor = strokecolor;
}

drawingdagak.prototype.Drawing=function(){
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext("2d");
    context.beginPath();
    context.moveTo(this.x,this.y);
    context.lineWidth = 1;
    for(var i=0; i<=this.n; i++){
        context.lineTo(this.x + this.r*Math.cos((Math.PI*2)/this.n*i),
            this.y + this.r*Math.sin((Math.PI*2)/this.n*i));
    }
    context.strokeStyle = this.strokecolor;
    context.stroke();
    context.fillStyle = this.fillcolor;
    context.fill();
    context.closePath();
};

var drawingsubbox = function(x, y, w, h, fillcolor){
    this.x = x;
    this.y = y;
    this.w = w;
    this.h = h;
    this.fillcolor = fillcolor;
}

drawingsubbox.prototype.Drawing=function(){
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext("2d");
    context.fillStyle = this.fillcolor;
    context.fillRect(this.x, this.y, this.w, this.h)
}
```

- 다각형을 그려주는 drawingdagak객체와 사각형을그려주는 drawingsubbox 객체 생성.



- 생성자를 통해 필요한변수를 생성자 함수에서 첫번째 생성자 함수에서 정의하였음. (drawingdagak : 중심이되는 x,y좌표. 크기를 정해주는 반지름r, n각형, 색칠되는 색 fillcolor, 테두리색 strokecolor

drawingsubbox : 기준점 x, y좌표, 너비와 높이 w,h ,색칠되는 색 fillcolor

- 각각 그림을 그려주는 공용메소드를 프로토타입을 이용해 만들어줌.

```
var submenu1icon = new drawingdagak(130, 420, 25, 3, "#21E7CC", "#21E7CC");
submenu1icon.Drawing();

var submenu2icon = new drawingdagak(360, 420, 25, 4, "#0151FF", "#0151FF");
submenu2icon.Drawing();

var submenu3icon = new drawingdagak(590, 420, 25, 5, "#FFAA50", "#FFAA50");
submenu3icon.Drawing();

var submenu4icon = new drawingdagak(130, 610, 25, 6, "#F9E900", "#F9E900");
submenu4icon.Drawing();

var submenu5icon = new drawingdagak(360, 610, 25, 7, "#BE33F2", "#BE33F2");
submenu5icon.Drawing();

var submenu6icon = new drawingdagak(590, 610, 25, 8, "#F62828", "#F62828");
submenu6icon.Drawing();

var submenu7icon = new drawingdagak(130, 800, 25, 9, "#1D3692", "#1D3692");
submenu7icon.Drawing();
```

- drawingdagak 객체를 이용해 삼각형부터 9각형까지 다각형 생성

```
var subbox1 = new drawingsubbox(25, 360, 210, 170, "white");
subbox1.Drawing();

var subbox2 = new drawingsubbox(255, 360, 210, 170, "white");
subbox2.Drawing();

var subbox3 = new drawingsubbox(485, 360, 210, 170, "white");
subbox3.Drawing();

var subbox4 = new drawingsubbox(25, 550, 210, 170, "white");
subbox4.Drawing();

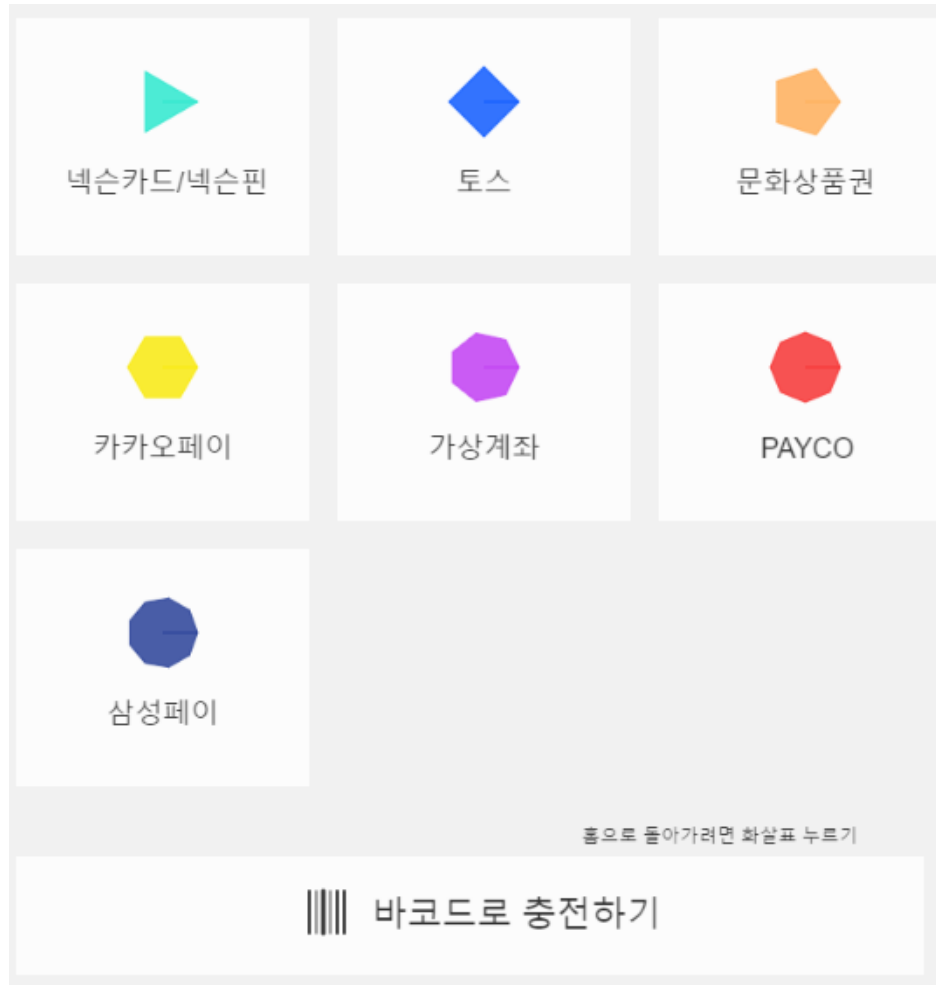
var subbox5 = new drawingsubbox(255, 550, 210, 170, "white");
subbox5.Drawing();

var subbox6 = new drawingsubbox(485, 550, 210, 170, "white");
subbox6.Drawing();

var subbox7 = new drawingsubbox(25, 740, 210, 170, "white");
subbox7.Drawing();

var subbox8 = new drawingsubbox(25, 960, 650, 85, "white");
subbox8.Drawing();
```

- drawingsubbox 객체를 이용한 8개의 하얀색 서브박스 생성



- <최종> 7개의 다양한 다각형과 8개의 사각형바탕이 생성되었다.

## <JavaScript 이벤트 핸들러>

```

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
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
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1
```

- 메인화면(screen1), 캐쉬충전(screen2), OTP(screen3) function으로 묶음.

```
screen1();
var screen=1;

//클릭이벤트
canvas.addEventListener("click", function(e) {
    if(screen==1){
        console.log(e.clientX);
        if(595<e.clientX && e.clientX<680 && 350 <e.clientY && e.clientY<430){
            screen = 2;
            context.clearRect(0,0,canvas.width,canvas.height);
            screen2();
        };
        if(100<e.clientX && e.clientX<190 && 350 <e.clientY && e.clientY<430){
            screen = 3;
            context.clearRect(0,0,canvas.width,canvas.height);
            screen3();
        };
    }else if(screen == 2 || screen ==3){
        if(30<e.clientX && e.clientX<100 && 20 <e.clientY && e.clientY<100){
            screen = 1;
            context.clearRect(0,0,canvas.width,canvas.height);
            screen1();
        };
    };
});
```

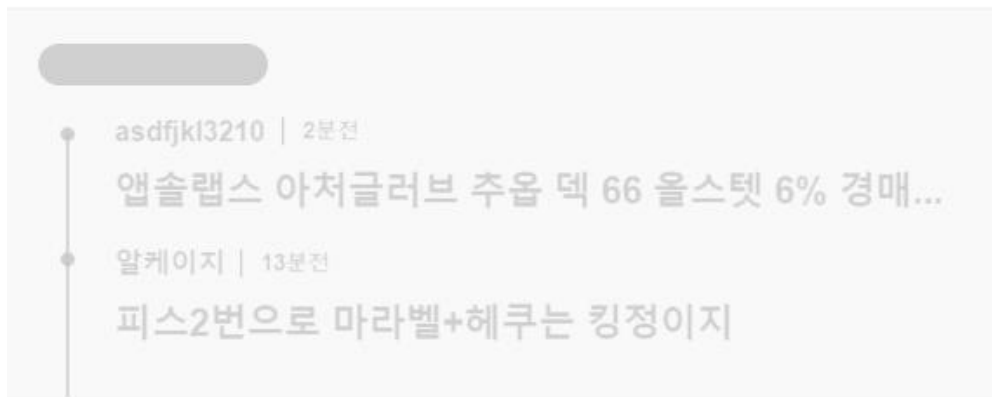
- 시작시에는 `screen1()`; 을 이용해 메인화면이 나오도록함.
- `addEventListener("click", function(e) {});` 이용
  - 일정 범위내를 클릭하면 메인화면에서 OTP 또는 캐쉬충전 화면으로 이동하도록 함. 또한 OTP, 캐쉬충전화면에서 왼쪽 상단 뒤로가기범위를 누르면 메인화면으로 돌아가도록 설정함.



내가 즐겨하는 게임

## 메이플스토리

3주 연속 접속! 다음주에 또 만나요~



내가 플레이한 게임

보관함 아이콘을 누르면 캐쉬충전으로 이동  
넥슨OTP 아이콘을 누르면 OTP입력으로 이동

- 다음과같이 모든 화면 우측하단에 입력방식을 설명해두었음.

## 1. 최종 결과

1단계는 기존 화면 디자인을 따라 구성하였음. 2단계는 객체를 이용하여 디자인을 추가하였으며, 3단계로는 이벤트 핸들러를 통해 화면전환을 구현해 내었음.

## 2. 논의

### 성공한 부분

기존 디자인과 가장 유사한 디자인의 화면이 만들어졌고, 프로토타입을 잘 이해하지 못해 사용이 힘들었으나 사용에 성공하고 응용하면서 객체의 중요성을 느낄 수 있었음. 또한 이벤트 핸들링을 이용한 화면전환도 잘 작동하였음.

## **실패한 부분**

일일이 각 디자인요소들을 하나하나 넣으면서 코드 길어도 길어지고 검증하면서 시간이 오래걸렸었다. 후에 객체 프로토타입을 이용해서 만들어보니 훨씬 간결하고 간편했었다. 기존 코드를 전부 수정하기엔 시간적으로 부족해서 수정하지 못했다. 또 화면전환때 사용하는 영역을 시각적 변화를 주지못했다.

## **자체 평가 및 향후 개선점**

객체와 프로토타입을 이용해 기존 코드를 간결하고 가독성이 높도록 바꾸고 주석을 추가할 예정이다.