

# HTML 웹 애플리케이션 실습

## 목차

- 변경점
- 수정 실습 1
- 수정 실습 2
- 수정 실습 3
- 최종 코드

# 변경점

## - 아이템 추가

벽돌을 깨면 일정 확률로 아이템이 드랍됨.

아이템 종류 : 공 개수 증가(+2), 공 개수 증가(\*2), 패들 길이 증가  
패들에 아이템이 접촉하면 효과 발동.

## - 레벨, 목숨 개념 삭제 + 벽돌 개수 조정

스테이지가 넘어가면서 무한대로 진행되던 게임에서 클리어가 가능한 게임으로 변경.  
기존 벽돌 개수로는 플레이 타임이 너무 짧기에 벽돌 개수를 늘려 플레이 타임 연장.

## - 벽돌 배치 변경

기존의 벽돌 배치 : 직사각형 모양

변경된 벽돌 배치 : 현재 강의에 사용되는 챗 GPT의 로고를 본뜬 모양

# 수정 실습 1

실습 프롬프트 : 벽돌을 깼을 때 20퍼센트 확률로 아이템이 나오게 해줘  
아이템 종류는 공 개수 증가(+2), 공 개수 증가(\*2), 패들 길이 증가(지속시간 5초)  
아이템은 위에서 천천히 내려오다가 패들에 닿으면 얻어지도록 공 개수 증가는 공의 현재 위치 기준에서 공이 증가하게 만들어줘

결과 : 만들어진 html파일이 매우 크기 때문에 확인 질문에 대한 답을 요구함.

공 복제 방식

A. dx/dy 그대로 복제

B. 랜덤 약간 추가

패들 증가

A. 110 → 160px 사이즈 OK?

B. 너가 원하는 특정 크기 있음?

코드 제공 방식

A. 전체 HTML 완성본 다시 만들어줘

B. 필요한 수정 부분만 덧붙여줘

답변 제시 :

1. 공 복제 방식

A : 속도는 그대로 방향은 진행 방향에서 약간 조정해서 복제

2. 패들 증가

A : OK

3. 코드 제공 방식

A : 전체 완성본 다시 만들어줘

## 수정 실습 2

실습 프롬프트 : 레벨과 목숨 개념은 삭제하고 벽돌을 더 작고 촘촘하게 해서 개수를 늘려줘

결과 : 레벨이 삭제됨. 공이 모두 떨어지면 게임이 끝나는 방식으로 변경됨. 벽돌의 크기가 기존 64 \* 20 크기에서 40 \* 16 크기로 작아졌고, 벽돌 배치는 14줄 \* 16열 개로 변경됐지만 벽돌 개수가 늘어나면서 패들과 벽돌 사이의 공간이 작아지는 문제 발생

추가 요구사항 : 패들이랑 벽돌 사이를 좀더 늘려줘

결과 : 벽돌과 패들 사이의 간격에 대한 답 요구

패들과 벽돌 사이 간격을 120으로 할까?

아니면 150, 200처럼 더 넓게 할까?

원하는 숫자 말해주면 그 값으로 완전체 다시 만들어줄게!

이후 값을 200으로 설정했지만 차이점이 보이지 않았고, 마우스로 패들을 움직이는 기능이 삭제되는 문제가 발생함. 게임 화면의 크기로 인한 문제로 판단되어 게임 화면 크기 조정을 요구함.

추가 요구사항 : 마우스로 움직이는 기능이 사라졌는데? 그리고 벽돌 이랑 패들 사이 거리가 안늘어났어 게임 화면을 세로로 더 길게 만들어줘

결과 : 마우스로 움직이는 기능이 복구되었고, 벽돌과 패들 사이 거리가 원했던 만큼 늘어남.

## 수정 실습 3

실습 프롬프트 : 벽돌을 상단으로 몰아놓고 크기를 줄인다음 개수를 늘려줘 그리고 벽돌은 gpt로고 모양으로 배치해줘

결과 : 상단으로 깔끔하게 배치되었지만, 원하는 모양이 나오지는 않음.

추가 요구사항 : gpt로고가 이거 아니냐? 이모양으로 만들어줘 필요하면 벽돌 크기랑 개수 조정해도 됨



결과 : 모양은 비슷하게 나왔지만 해상도가 떨어짐.

추가 요구사항 : 벽돌 개수 더 늘려서 로고를 더 뚜렷하게 나타내줘

결과 : png 파일의 경로와 이미지를 분석하여 더 뚜렷하게 나타냄. 주의점도 설명해줌.

좋아, 이번엔 **\*\*진짜 GPT 로고 이미지(gpt 꽃 모양)\*\***를 그대로 샘플링해서 벽돌을 훨씬 촘촘하게 찍어내는 방식으로 바꿔봤어.

△ 준비물 딱 하나

이 HTML 파일과 같은 폴더에 gpt\_logo.png 이름으로 방금 그 GPT 로고 이미지를 저장해둬.

(파일명만 맞으면 됨)

아래 코드 전체를 gpt-breakout.html 정도로 저장해서 실행하면 돼.

결과 : html 파일을 직접 여는 경우 로고를 불러올 수 없는 문제 발생.

추가 요구사항 : 같은 파일에 넣어도 로고가 안불러와지는데?

결과 : base64로 넣는 방식과 파일을 불러오는 방식 선택지 제시

✓ A) "응, base64로 넣어서 HTML 하나로 만들어줘"

→ 로고 무조건 뜨고, 파일 필요 없음

→ 바로 완성본 HTML 만들어서 줄게

✓ B) "아니야, 파일로 불러오는 방식을 쓰고 싶어"

→ 그러면 정확하게 로고 png를 불러오는 폴더 구조 방식으로 다시 안내해줄게

추가 요구사항 : A

결과 : 코드에서 Base64 문자열이 너무 길어서 생략됨. 생략되지 않은 html 파일을 다운받을것을 권장.

# 최종 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8"/>
  <title>GPT 로고 벽돌깨기 (report.html)</title>
  <style>
    *margin: 0; padding: 0; box-sizing: border-box; }
    body{
      min-height: 100vh;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      background: #0b0e1a;
      color: #fff;
      font-family: system-ui, -apple-system, sans-serif;
    }
    h1margin-bottom: 10px; }
    .game-wrapper{
      position: relative;
      overflow: hidden;
      border-radius: 12px;
      box-shadow: 0 0 20px rgba(0,0,0,0.7);
    }
    canvas{
      display: block;
      background: #050818;
    }
    .center-text{
      position: absolute;
      inset: 0;
      display: flex;
      align-items: center;
      justify-content: center;
      pointer-events: none;
      color: white;
      font-size: 18px;
      text-shadow: 0 0 5px black;
      text-align: center;
      padding: 0 10px;
    }
  </style>
</head>
<body>
<h1>GPT 로고 벽돌깨기</h1>
<div class="game-wrapper">
  <canvas id="gameCanvas"width="800"height="900"></canvas>
  <div class="center-text" id="centerText">GPT 로고 불러오는 중...</div>
</div>
<script>
const canvas=document.getElementById('gameCanvas');
const ctx=canvas.getContext('2d');
const centerText=document.getElementById('centerText');
let isRunning=false;
```

```
let ballStucktrue;
const ORIGINAL_PADDLE_WIDTH110;
const ENLARGED_PADDLE_WIDTH160;
const PADDLE_HEIGHT14;
const PADDLE_MARGIN_BOTTOM40;
let paddlewidthORIGINAL_PADDLE_WIDTH;
let paddleX= (canvas.widthpaddleWidth) / 2;
let rightPressedfalse;
let leftPressedfalse;
let isPaddleEnlargedfalse;
let paddleEnlargeEndTime0;
const BALL_RADIUS8;
let balls= [];
const GRID_COLS70;
const GRID_ROWS70;
const BRICK_WIDTH10;
const BRICK_HEIGHT6;
const BRICK_PADDING2;
const BRICK_OFFSET_TOP40;
let BRICK_OFFSET_LEFT0;
let brickMasknull;
let bricks= [];
let bricksRemaining0;
const ITEM_DROP_PROB0.2;
const ITEM_SIZE18;
const ITEM_FALL_SPEED2;
const ITEM_PLUS2'PLUS2';
const ITEM_MULT2'MULT2';
const ITEM_PADDLE'PADDLE';
let items= [];
// Base64 로고 데이터
const LOGO_BASE64='iVBORw0KGgoAAAANSUhEUGAAANUAAACUCAIAAAACLJvsdAAAQAE1EQVR4AeydC2wVxRrHUAHiI4A0QbBBACmFDBggyBtrMCQCQVS
DysAgWbA1FoBXKq1AebaGFQnmWAgawyBWpUKQICQ+VAKWAsSAQfASKIILGaJfBUTG9v9uJyzKzZ8+2Pe12dreZe/zmm2++2Z35n+81veX0Eu/H
2wH7duD0Gt6PtwP27YCHP/v23lu5Rg0Pfx4K7NwBD3927r63toc/DwN27oDb8Wfn3ntre/GfhwF7d8Czf/buv9tX9/DndgTY+/4e/uzdf7ev7u
HP7Q1w9/09/Nm7/3avbv6Hv7sPgF3r+/hz93nb/fbe/iz+wTcvb6HP3efv91v7+HP7hNw9/oe/tX9/na/vd34s/v9vfXt3QEPf/buv9tX9/Dn
dgTY+/4e/uzdf7ev7uHP7Q1w9/09/Jnt/99///3zzz9/8803J06c0HLkYKHSHwi6Fy5cuHbmtlkb8zCDnj4M9gkgAXIsrOz09PTFY1a1JycPH
369ISEhPHjx0+aNAkiJSVl9uzZS5YsyczM3L59+5dffglSDRQZFzrldwsPfbQjA1GVlZS1YsCAxMXH06NFxcXHz5s175513du/evW/fvoCKgqNH
j0K8//77MGfPnv3GG29MmTIFIikpCSBiLG9T53UC7YCHv3926Pvvv09LS5s5c+a4cePAHID79ddf/xkz/c/XX38NHIEgc2fMmIHVxHyazvAGb+
2Ah7//7uW276233sLagSSLsPv/tNv/d/r06VwrVgHBadOm4b5vH/R6xjvgdvYRRkRHR4MYkGe8Q2XkYg5B4dixY8H0H3/8UcbZrhN3Nf6wUoAP
uOB8g3vyvRIn48blz53oRofnGuhd/pAuxsbHEeSYbVKdOnZiYmJycHPB0+fL14uLioqKiwsLCvLw8UDt06FCTufhx4kjcetDBbbKockMuxR/gI2
8lmfV3YCS2YI5MIj09fcaAAW3b3tn3ooYfq1q1bv379Jk2adOnSJSoqavPmzTdv3gSdnTp18qeHNJnaDwk17eOPP8Ypk+XMKf3RaJdutpF3+ts4
B/Op1lPAI1AzfMeXX34ZI5ErKQhMDAX0zLvuuugt05ufnA1Z/KASC/yn96d27N8rJckiWaUB84MCBrVq1CG0NbdGiRxx8PGvtvXI30K7DH7XiFS
tWGFq+1q1a6MNGzZg5Mp69oD180HDDQA2XXda5yPNlwFm3a9cuLCwMu4jLdklB2134w59u3LjRMNV99tlnsYvACDSUr2ELn3nmGaqA5ZvumwXy
sIsYxbVr10L7ma7+dBf+9uzZg/GTjx03uGvXLSi7ecgiHxi08A49uFeLU0zEyF1A4aBBg9DJd8ZEUVUhf+GPDGDr1q0crXBmpLGrV68WmNa74I
04jbIz4R1Xc9YnBpQkSECnsw2hW/BHOEW1T/a8eExykbvvvjsGmQBdII88EHcRj1GFtBzyE4A01Yt5t8faDjwzUNGDCqXy4Stem2Ood2Cv/Pn
z+Nh5W0jREw9ReYH5GBNN23aBIDA4kw+Kb+AjpJLxITE0EScPc1aDjwU1NTkSEA9aeH6ZGRkazoT0Bdvlvwd+nSJdn4UQRp3759WQ+PtIDyIX
kG0R55q7/p4Ancg0/JkydTLOzVq1fz5s3JrL61vgYNBz6jyKAQeWYzKsQXT5gwgdtCw1F1mc7HH5kBNjc3N1c4JEPyWC+gIPBNuoR6JAQLFy6k
biejWzTiHQcZDNuYMMOoVFtJa5BBEnlMMZdn07RpBFc1fGFAv8XxAOFY/HH3Tz2FWho+jrZu3TrhtDjsRo0aCUyTLiBesmQJRohPEzGiuzVr1q
CCuG4VGRNJehQ5ZjEXj0xcKATQX1y5cqXMV5fjQPxhpbjmmjp1KrdcxpJABcshp73dunXD61g50QKvZcuWJSQkYH50nz7tbwrWdP/+/YSDWDKL
mg1VMTc8P3wyuKEvpn6UnZ1tOFFpKpWyrq230iM0zUtGnTgJ1JNYS0WrIx80EZHQRnJtq4Gt6yZcvixYsJ5Qb9T1KRTwxhy5YtuQPkOQU9fJ
HWr18ftEBQ0F71XYfgjWpJSGJjYwEKTiPgkNS9e3fqzgeAJtR7/fXXk50TDc2nd1JLly49fvz4c889FXTkaWohSMYBoBwL8jy8IwIoAa7BH1iJ
jo7G5pn4R/1pPfbYY7g5PUDPU2wbN24cvpskwyTDJf8tLCwEoyCvTHmMfi1zmgsZpFj+qhDa8pDmc5UYdQL+iIcmTZq0e/du6zseGhQKyTSU52
iJ/T11e+QBuIMHD3JrgomqJ0Rpz4ZPJ93Wuj7i2LFj03fu9NFKfyqPP1INU56CggJ/xwBWZPsREhJiKE/WTBpBjGU46mPi+6hm48EtIg+dpC8t
WRRo06YNXxnfKjJ9E1kK8jzhZ599FjDMEGZVw67a+OM4STX8HQPb+9mzZwnOOC2LW3/lyhV/2tDALUVRUdHlKSmtIo8p5EMdOnQgg8GaEhsMGt
IkLCyMWzuGrDcq1aTegvyZM2ewggJTua7C+CPmmz9/viFcsHmUQvbu3UsWef369YqfCkYRKJOLAACL2gBZ3759cZ3ATj+FB+a+mP0CD3fnCYe
FQRQe/LkSYGpXNdu/JV3w4i+KfMau13MHv6RsKm8um/Nw3FThMvJyeGqFyjfGvBPKThTL4yPjwdkjiEpThxDmJaWxvUMU/zr+2cE3P0l+qfz73
9++0EHpv/bU/K/SuKPiGrPnj0kp/KWEwti9qz7R1mDxgF8b7755nnvvTdgwAB/yYomDAGMMGkg9amnnuIaDU7ARrG6Z8+e27Zt09DIHdH5SkuC
ACWnixcvCky1ukri7/Dhw4bXUARJ6enpwTqAhg0bPvHEEyZ1Gv1C2KEDBw5QkaGBQv0Q9piEWs/R07hRgkKKONzvGUL9kEB37txZ4Fy9epV1Ba
```



ZaXfXwx+nm5UsYzsgsbTbm00zeBwUxU3/9FVADoAE6hr9/igvGHi9cuBA/CyFkXjX12PJ+/fpRvryRqXLPrfD3RtG1TfRf63L1z3333HYS6TT3R  
EflxTsKOY2MmT54cFLcraDbpAhTgsnbtq25du1Iv1EsSq/F94Mo4J5WlbdU2FGsgkpKSYOrF9DRJ0vb71Vde4XqNd5Tice3atfXy0Ez57bfffIN  
RtiuePD0NiK283VZjGjRvL/MrJABGM1ogRI4JhhFVIIknMsYgEjtpXAmLw4MEwyTwQEKZotXIqMAoQQCq9xBoFombNmsAawk1NMfwrBq9YSUI4  
AAAGM2M1RRAm1q8LlKjQARFCPa13vRLMMIAQIwQehliByRACiJUGhfrZt2iQTVCI42Yh/PutAcDRKuEPf0dpA6cJnEJERATICYFZGV2gQNERWF  
DVAYLCeUAjMxwFRWwUoPAmDfJ3JUAI1lm2oEfrEmamHT2wYIFLAqTqJ3qh3BSUwL/169f/+STT4Tdxwg1a9ZMYFZG1yQDKHCTASwM9f/yyy9c  
CiNmOCowAVOXL12oKlNcxH4LoLoXwFHK4d51zpw5smasKRfZmrcKHr4w/L15+cLu0yduBLPAKNLAKsyARQEH6t/GFJy7B1iCDNFp+SPxmzz/A  
SFOBDF7U/s6NGjaE5MTBQECHkr+92FFYPeVQl/xcXfHoFCfM8D8C00FZ1C6ISEh3PZyxYZn4+z13z/dsmULyYSwFmIIMyUrKwt/LYwadJfJ4eHh  
5NEEHsY1Gvk2pUmTJiDYUKcGTGXWzR2C5FfYVvKu+++X2AGq0uuQ5Ub48TBY3r1arFGHYWfgwYNIpOFoKsfrZgpxiJUluKgeHL9qCGN0wZMo0  
aN+vDDD/G2hjiYe/A9/PDDM18hjJL4u3nzplxrbd++fb169Sppu4ER3havqtdPuIKNptANXAANpheCLkywQJckdCNHIEYegLpOjtfGaoIH8zvL  
495Zr82QxvYDQcMhVZjK4I8Nxf/yqW8PPPAAPJlPqTaw0jpcfxQ1dR6A1P+Ibow8cWecYjph0AhFRDhw7x8F6w3THaSE2448G/61UJMS898  
gJ4I05rjJ43JqBgmmsL+AT74VEGSGsqXmQvpJrG4/CLkiILZswpHEB2KMKXkLp+qdzMxMYchf+NKvZy3Fpy7F093rp1K2w2jameIOQz+wnL1  
999nbPGff/73CQNMHAMpJly+fhfSPekPwV3dunURZgoT25SeBiQC3dVKQDkZMo1cJ0uIp/xx7LYbGhcPvkKoyM/tq/yUG2Uy420FJeaKZEBIY  
+ac5jCRByoxYF7GRzr127YsGG4ZiupCQvh3EtKSnC40EKjGMkqA10hrkr4k2t2586doYhdSdsN30+5556KKK9Vq1bDhg9NNMC6wsLCuAlBe11E  
rRdffGYFY9yWQ0xgeEq1z+pDP64fZfxx31D5eEv6IdHtUjQGRcX50s1rKAQX/zBBx/ISqh0Hz582GJyIzyA7V118Ec8j1MT9gsXxskJzD1Jq1  
KYMg21PZyvfiHSCG6To6gJdxw4EPBdiB2xdvI+ULs+deqUxQ0qtDL4Y00JPwQH9PnnnsfjQ0/igsIqbK8++wRYcJlLvziYGAFduGNQhZ5c1KT  
P0z/1aTmzZvL4gPiLmZLmc+VPEOymuphD/y3+7duwvvgEv68ccfBaa+SxAm+ywqecRM5ietVxIUmswDPdRoUlJ5S5+fTymbtrtYpdpQyYSi9e  
vXY9c1vkDgB4YMG5J/D9mHK1euCMLV6sS/kgIevToIewp0Y+5zWjUqBHeTzHNS42NpbIXBOFSroEs1pVwUo3HNw40KNZSe0HSbGDC/MtYrw  
nOwDtUCLqYwW18auSvjjq9+4cWPZmK1Zs6aoqMjFjNLY/fv3p5iINDQHNmHCBErxV1bWbQsX/E2vPD4XdlFRUXWTCaQFVSGsmxh19qfJx47yPp  
w8edIEtcT51aSrEv7YsGYNKGEREDo2759+azY5S6U0bHw4Cm5aW4mhCoagRd+MeXnZ2N8SDE1GuuApraH18PYaGrV6/yMAJ3YUukfFh0KFD  
OHG9WPWnFmCfMrocoXkgh2WUkacePH4fw17AZnPSYMWPIH7E3ASGwjJNmzCAZUOWP+rArPVu3F16W76F8RS7IiF27+4rhj+1q2rRpZGQkh6RAH  
LrFdCncmzU2yZonJiTkY04YzT4DRKRebXWmyaAFW87BTqzqIF0tmnr4A00YQAE97CnXhJdaVg4Ad9yzZ893332XmjtOkb1aI/zX6GpOCCEu+WHI  
YJM4pBq+kXr4YxPxpKNGjYIQGHcKMGH11+1wx6Bw0KBBeG2CP0GRPr0sQRkXDPa7zSRp+MvHixYsoMRytODMkKJITUXtDz+++C5xq31USf+xpnz  
59COMghfZQUONChqWff/2FMGJK7oBD7R+RXWFGogI1RieS9qr776a1jQNExhITNRiQ8bFA7279577xvUkTVX9i8ECStWsKsq/vDCJB0krvL7c+S9  
e/fGF1ssR0BCSiG7du3Ky8sTtKGK4y3EfWAAAC15JREFUIAQR3H7QpCasn0UAEf/6x6+YyHt9KEfLO6Aq/ngTQNOvXz8MGLTCKKzUq1dv2bJ11u  
mHupOz5JqHwI2tbsmrJWom/ZGUiJaHMZmZ0YDfYGL1X0YfOmfpM1Cixy4Gfa3KU23g/irwY9S377rb1ZosL8+00P444B8w1Ro4C5RSPCxdq  
NEznZsJKq6n2o1mZzGGvJWNK2yCmU02C9wqglDYdxundunP/PnzAEWxgruGE9NtAZQs0jQgS3b18oL20PvtvQK12vDPGZfbaaRo4C5RSPCxdq  
KdDGL9B6Vh/86c050oAn81a9YUNWU5qzD+qLUGLPhw487Jozjeto3SGNB6fBOCLJ5L4pUrV5KaChHE6wtRzAX/KDQ1etBeoJQoatWrakG/eij  
j+qHgkj/9NPNcrWIBw7hbBBXqWxVCuOPS8BTGwQWBE4+i61fMouwAXG4r71Q4Y0d8cDBgyUGkmi2bBmMwV/ggTEUM4IyOjV69etWrvVaqRfj+  
UrJIOPG2Hu5YTnqebdO6v585k8H1EaBTZBYPHixUuXLqUQKPC1LseGu5w6dSruVfCYmoxAYFSioqlQDLBwqc1oXu4v9GdGGMaPzKIHj3rVu3  
CpqJAfC/ArOadxXGH7VWucDRvHnz8ePHgWMSCADhb/cxgZwWYuNFss0FHvbt29v+P/nINTDSqalpXEH2LJ1S3+LBouP5Sbo5LZXUGj9rWULE2  
3sKow/ufRApt68ePPTm3b2p071zoIKBScJqNao6oyYSiTvWxHj1jRGPKBm603AUx0lv51GgmpqZiUOXfjZVVSZzyMPjCrFmzRphJdEtBSq3g  
j1dQGh88vdD0UOMkiAMmTpz49tvtAxFBuuvigoniK9Tus0OahYNIF9Zv349MjSSX40PQZLBdTLXJ8SIRIpwqqBh/Pbv34/9FtaKiHhQ8W/BKI  
w/+FZJdsFEYMHMD548etJjVZ20gkItjMIOpZc70BpG+oyW/oUvMx5Bg9jA2PLfIo5s1ydfN298HzDtw1o8A7EBLyyvwq39Yxfxh7dh3YYsN800c  
0/Dhw6mwKJqYBAY1dgJryc1JQE74k1MW0xMDEhxCa3GijKJ9r2oqglBPWjQvBKAnHRL4r732G1GHwF51qzb+HnzwQWGXSYZ7Y4A7B15kvkROOm  
rH1D+xoB62kEfGCKHZQp8khvDgwYPcO3NTV2U017c0n45HPJj8FSNDIpdS0fjXUgrjJx2Xs9F86Q+k8PJaql3LTDpBQIjCnUaYCi8/DIEia+  
OycnB9NIkoES/ZA/mhLjnJ17UOVPOEx8wj6+Phw15F1EFmS+M18JjsL4a9cGQYsWLYRd1qtiggbgBgAsKqSoDKdmDI6BvCHDqhI/h4eFM1A/5o8  
1PyWaGDRu2adMmfzJ14pMysbPs+VCCSX7hhReQ3hizdFCawvgDGU21fxKIREFwmobhbsGTBzqV0NiKgMZWDiQ7/99FMcLmsxBU7ARuD44osv  
TpykBQUKnw8H1AAKGPhdu/eUUtilakf8WzykCochFHHfINxoPYLwO/kB/quCQ2kQOGsWbMuX77MWeoLcary8vL509MpaC0mH/JH43AjIyMjxc  
CKf1vQBDL8zRX4XKPduHFj+/bntNt3xsIVFB0IAnRjPKZNM4Yhh1a3V2j/jh07ct7C71NVJ1QXmOZdTMJjevXSBHCUG9n45Kp/f+teGyHx8PMYy  
N+1fJQbWl/oHTHdCX0T4C03z44Wmz1HgDghYEdwYELwkct4NQZvAYTxh+4J1uV952TK8k8Q0Buc+kBvS5AwppAV1YU5MIEYEpMZdhawEY  
jEQmYamk9NDG63A938kdr4C03v7FYEPEqG6B5GegYQ33NQK5C0WmjnG40B3rOgCk15vXx4wH/fHvTRTpyeIjQrNB8vndUHN/EZ127ZmdqH7Xsf  
7XPIKcGkj303iATe11bV3LhxrGtiqAhmC0d++Gug4ktG6u8BKJh/9dVx/kb1fMweUM7NzewV9Xy1aeXxx+5TgMAQQAjt+eeF57ZKYardMnVBHS  
n16twrCfUoyvibi83DJkempK++DN72lwQJLXWuuYeyEPhunXrgLLFZMQhtZvK0gF/nDFnJ6f/aX2y7HhJfGvW1A5uiBvx44d/fv3nzBhgs10  
vgnUtzMyMggrZTLcL6G6S1L8Ezo4yYmLS2NUdQglJmiXhMC/th0Djs2NhzCaBwt6dP4LG50hShrXeZS1QMh+HQU+ptIBYfST51+//TatWuy8c  
NxEOy3Pz589GwnJzsmGhP3jqH4I8Xo0pMurHCanZwc5zLlfxpUWEiXfCB+Vy0aBEJtU1tBcQ0mc2d07esv39awfG0J86EE1RzgN3IKsmNs3se  
zDHNOfgjxscAJX5bIixwEdUUVBQxJLhsWUZgYMMkPQsX44dC0TQIAHoXUBPLMgn5RuNaYUgmpQxTSJFRdNJGyB5VjgHf7xnkyZNsByEGESUwm  
5cXBwYxUtSjdu+ftU2jQQfQpKaNBz4y5YtQ4bsFd5aOfyCyy1btqAT/wj6WaJMDc8LvUowP4cPb1Y1/5yxsK5dXUfhj0EcgAanBG0YcmDc+pg  
C80WmJ1Im0v69wc6ISEBPhhFhVtUANMrh+weSxU7t8/Bev4a1QJ7emnn27cULHARMSu3aqdhj/2EyuITZJdG0P6hs3DAIEGknwN2sqvL2Aa2  
7cWJHfPyWAhEkyZevb0a1bn4dVWPR7LTM0XB

```

RUSErkJggg==';
function initPaddle() {
  paddleX= (canvas.widthpaddleWidth) / 2;
}
function createBall() {
  const speed4;
  const angle= (Math.random() * 0.50.3) * Math.PI;
  return{
    x:paddleXpaddleWidth/2,
    y:canvas.heightPADDLE_MARGIN_BOTTOMPADDLE_HEIGHTBALL_RADIUS,
    dx:speedMath.cos(angle),
    dy:-Math.abs(speedMath.sin(angle)),
    radius:BALL_RADIUS
  };
}
function resetBalls() {
  balls= [createBall()];
  ballStucktrue;
}
function createBrickMaskFromImage(img) {
  const off=document.createElement('canvas');
  off.widthGRID_COLS;
  off.heightGRID_ROWS;
  const octx=off.getContext('2d');
  const scaleMath.min(GRID_COLSimg.width, GRID_ROWSimg.height);
  const drawWimg.widthscale;
  const drawHimg.heightscale;
  const dx= (GRID_COLSdrawW) / 2;
  const dy0;
  octx.drawImage(img, dx, dy, drawW, drawH);
  const dataoctx.getImageData(0, 0, GRID_COLS, GRID_ROWS).data;
  brickMask= [];
  for(let y0; y < GRID_ROWS; y++) {
    brickMask[y] = [];
    for(let x0; x < GRID_COLS; x++) {
      const idx= (yGRID_COLSx) * 4;
      const rdata[idx];
      const gdata[idx+1];
      const bdata[idx+2];
      const adata[idx+3];
      const brightness0.299*r0.587*g0.114*b;
      const isBrick > 10 && brightness < 230;
      brickMask[y][x] = isBrick;
    }
  }
  const totalWidthGRID_COLS* (BRICK_WIDTHBRICK_PADDING);
  BRICK_OFFSET_LEFT= (canvas.widthtotalWidth) / 2;
}
function initBricks() {
  bricks= [];
  bricksRemaining0;
  for(let y0; y < GRID_ROWS; y++) {
    bricks[y] = [];
    for(let x0; x < GRID_COLS; x++) {
      if(!brickMask[y][x]) {
        bricks[y][x] = null;
        continue;
      }
    }
  }
  const bxBRICK_OFFSET_LEFTx* (BRICK_WIDTHBRICK_PADDING);

```

```

    const byBRICK_OFFSET_TOPy* (BRICK_HEIGHTBRICK_PADDING);
    bricks[y][x] = { x:bx, y:by, status:1};
    bricksRemaining++;
}
}
}
function resetGame() {
    paddleWidthORIGINAL_PADDLE_WIDTH;
    items= [];
    initPaddle();
    initBricks();
    resetBalls();
    isRunningfalse;
    ballStucktrue;
    centerText.textContent'Space 또는 클릭해서 시작';
    centerText.style.opacity'1';
}
function drawPaddle() {
    ctx.fillStyle'#39c6ff';
    const ycanvas.heightPADDLE_MARGIN_BOTTOMPADDLE_HEIGHT;
    ctx.fillRect(paddleX, y, paddleWidth, PADDLE_HEIGHT);
}
function drawBalls() {
    ctx.fillStyle'ff2d8d';
    for(const b of balls) {
        ctx.beginPath();
        ctx.arc(b.x, b.y, b.radius, 0, Math.PI*2);
        ctx.fill();
    }
}
function drawBricks() {
    for(let r0; r < GRID_ROWS; r++) {
        for(let c0; c < GRID_COLS; c++) {
            const bbricks[r][c];
            if(!b|| b.status=== 0) continue;
            ctx.fillStyle'hsl('+(r+c)*3) + ', 80%, 50%';
            ctx.fillRect(b.x, b.y, BRICK_WIDTH, BRICK_HEIGHT);
        }
    }
}
function drawItems() {
    for(const it of items) {
        if(!it.active) continue;
        let color='fff', text='';
        if(it.type===ITEM_PLUS2) { color='ffee00'; text='+2'; }
        if(it.type===ITEM_MULT2) { color='ff0059'; text='x2'; }
        if(it.type===ITEM_PADDLE){ color='#00d0ff'; text='P'; }
        ctx.fillStylecolor;
        ctx.beginPath();
        ctx.arc(it.x, it.y, ITEM_SIZE/2, 0, Math.PI*2);
        ctx.fill();
        ctx.fillStyle'#000';
        ctx.font'10px system-ui';
        ctx.textAlign'center';
        ctx.textBaseline'middle';
        ctx.fillText(text, it.x, it.y);
    }
}
function maybeDropItem(x,y) {

```

```

    if(Math.random() > ITEM_DROP_PROB) return;
    const types= [ITEM_PLUS2, ITEM_MULT2, ITEM_PADDLE];
    items.push({
      x,y,
      type:types[Math.floor(Math.random()*types.length)],
      active:true,
      vy:ITEM_FALL_SPEED
    });
  }
}

function updateItems() {
  const paddleTop=canvas.height-PADDLE_MARGIN_BOTTOM-PADDLE_HEIGHT;
  for(const it of items) {
    if(!it.active) continue;
    it.y+= it.vy;
    if(
      it.y*ITEM_SIZE/2>= paddleTop &&
      it.x>= paddleX && it.x<= paddleX+paddleWidth
    ) {
      applyItemEffect(it.type);
      it.active=false;
    }

    if(it.y > canvas.height) it.active=false;
  }
  items=items.filter(x=>x.active);
}

function applyItemEffect(type) {
  const baseballs=[0];
  if(!base) return;
  if(type===ITEM_PLUS2) {
    spawnExtraBalls(base, 2);
  } else if(type===ITEM_MULT2) {
    const copy= [...baseballs];
    for(const b of copy) spawnExtraBalls(b,1);
  } else if(type===ITEM_PADDLE) {
    enlargePaddle();
  }
}

function spawnExtraBalls(base, count) {
  const speed=Math.sqrt(base.dx**2+base.dy**2);
  const angle=Math.atan2(base.dy, base.dx);
  const spread=Math.PI/180;
  for(let i=0;i<count;i++){
    const angle=angle+ (i===0? -spread:spread);
    balls.push({
      x:base.x, y:base.y,
      dx:speed*Math.cos(angle),
      dy:speed*Math.sin(angle),
      radius:BALL_RADIUS
    });
  }
}

function enlargePaddle() {
  const centerPaddleX=paddleWidth/2;
  paddleWidth=ENLARGED_PADDLE_WIDTH;
  paddleX=centerPaddleWidth/2;
  paddleEnlargeEndTime=Date.now() + 5000;
  isPaddleEnlarged=true;
}

function updatePaddleEnlarge() {
  if(!isPaddleEnlarged) return;

```

```

    if(Date.now() > paddleEnlargeEndTime) {
        const centerpaddleXpaddleWidth/2;
        paddleWidthORIGINAL_PADDLE_WIDTH;
        paddleXcenterpaddleWidth/2;
        isPaddleEnlargedfalse;
    }
}

function updateBalls() {
    if(ballStuck) {
        const bballs[0];
        b.xpaddleXpaddleWidth/2;
        b.ycanvas.heightPADDLE_MARGIN_BOTTOMPADDLE_HEIGHTBALL_RADIUS;
        return;
    }

    const alive= [];
    for(const ball of balls) {
        if(ball.xball.dx < ball.radius|| ball.xball.dx > canvas.width-ball.radius)
            ball.dx*= -1;
        if(ball.yball.dy < ball.radius)
            ball.dy*= -1;
        handlePaddleCollision(ball);
        handleBrickCollision(ball);
        ball.x+= ball.dx;
        ball.y+= ball.dy;
        if(ball.y < canvas.heightball.radius) alive.push(ball);
    }

    ballsalive;
    if(balls.length=== 0) gameOver();
}

function handlePaddleCollision(ball) {
    const topcanvas.heightPADDLE_MARGIN_BOTTOMPADDLE_HEIGHT;
    if(
        ball.yball.radius>= top &&
        ball.x>= paddleX && ball.x<= paddleX+paddleWidth &&
        ball.dy > 0
    ) {
        const hit= (ball.x- (paddleXpaddleWidth/2)) / (paddleWidth/2);
        const anglehit* (75Math.PI/180);
        const speedMath.sqrt(ball.dx**2ball.dy**2);
        ball.dxspeedMath.sin(angle);
        ball.dy= -Math.abs(speedMath.cos(angle));
    }
}

function handleBrickCollision(ball) {
    for(let r=0;r<bricks.length;r++){
        for(let c=0;c<bricks[r].length;c++){
            const bbricks[r][c];
            if(!b|| b.status===0) continue;
            if(
                ball.x > b.x &&
                ball.x < b.xBRICK_WIDTH &&
                ball.y > b.y &&
                ball.y < b.yBRICK_HEIGHT
            ) {
                ball.dy*= -1;
                b.status0;
                bricksRemaining--;
                maybeDropItem(b.xBRICK_WIDTH/2, b.y);
                if(bricksRemaining=== 0){

```

```

        isRunningfalse;
        centerText.textContent'□ GPT 로고 전체 파괴 성공! Space/클릭으로 다시 시작';
        centerText.style.opacity'1';
    }

    return;
}
}
}
}

function updatePaddle() {
    if(rightPressed) paddleX+= 6;
    if(leftPressed) paddleX-= 6;
    paddleXMath.max(0, Math.min(paddleX, canvas.width-paddleWidth));
}

function gameLoop() {
    ctx.clearRect(0,0,canvas.width,canvas.height);
    drawBricks();
    drawPaddle();
    drawBalls();
    drawItems();
    if(isRunning) {
        updatePaddle();
        updateBalls();
        updateItems();
        updatePaddleEnlarge();
    }

    requestAnimationFrame(gameLoop);
}

document.addEventListener('keydown', e=>{
    if(e.code==='ArrowRight') rightPressedtrue;
    if(e.code==='ArrowLeft') leftPressedtrue;
    if(e.code==='Space') startGame();
});

document.addEventListener('keyup', e=>{
    if(e.code==='ArrowRight') rightPressedfalse;
    if(e.code==='ArrowLeft') leftPressedfalse;
});

document.addEventListener('mousemove', e=>{
    const rectcanvas.getBoundingClientRect();
    const x= (e.clientXrect.left) * (canvas.widthrect.width);
    paddleXxpaddleWidth/2;
    paddleXMath.max(0, Math.min(paddleX, canvas.width-paddleWidth));
});

canvas.addEventListener('click', startGame);
centerText.addEventListener('click', startGame);

function startGame() {
    if(!isRunning && bricksRemaining=== 0) resetGame();
    if(!isRunning && balls.length=== 0) resetGame();
    ballStuckfalse;
    isRunningtrue;
    centerText.style.opacity'0';
}

function gameOver() {
    isRunningfalse;
    balls= [];
    centerText.textContent'□ 게임 오버! Space/클릭으로 재시작';
    centerText.style.opacity'1';
}

function loadLogo() {

```

```
const imgnew Image();
img.src`data:image/png;base64,`LOGO_BASE64;
img.onload= () =>{
  createBrickMaskFromImage(img);
  resetGame();
  gameLoop();
};
img.onerror= () =>{
  centerText.textContent`로고 로딩 오류!`;
};
}
loadLogo();
</script>
</body>
</html>
```

PS. 수정실습에서 GPT와의 대화는 굵은 글씨로 작성