웹 프로그래밍(project 2) 보고서

캔버스와 자바 스크립트 활용하여 간단한 게임 만들기

**게임명**: Ppeokkeum Ppeokkeum(뻐끔뻐끔)

**조원**: 60191651 류채현, 60191674 이상은, 60191681 임성제

1. **게임 시나리오**
   1. 미로에서 ‘뻐끔이’가 먹이를 모두 먹어야 승리하는 게임이다.
   2. ‘뻐끔이’가 상어에 닿을 때마다 목숨(라이프)이 하나 깎이고 제자리로 돌아간다. 남은 먹이의 수는 리셋되지 않으며, 캐릭터의 위치만 제자리로 돌아간다.
   3. 상어는 라운드를 거듭할수록 빨라지고, 캐릭터는 상어를 피해 먹이를 모두 먹어야 한다.
   4. 라이프가 모두 소모되면 게임이 종료된다.
   5. 게임을 진행할 때마다 라운드별로 통과시간, 남은 라이프 수에 따라 추가 점수가 주어진다. (각 라운드가 끝나면 라이프는 다시 리셋된다.)
   6. ‘뻐끔이’를 방해하는 상어를 피해 먹이를 다 먹은 자는 이 게임에서 승리하게 된다.
2. **구현 내용**
   1. **Main\_Page(Main\_page.html)**

* 게임 시작 전 게임설명, 게임 룰 등을 확인하고 배경음악을 끄고, 킬 수 있으며 게임 시작 버튼이 있는 페이지이다.
* Description, Game Rule, Sound On, Sound Off, Game Start는 버튼이고, onclick을 활용해 description은 따로 만들어둔 description 페이지로, game rule은 gamerule 페이지로 연결된다. 이때 onclick = “window.open(‘desciption.html’)”을 사용했다.
* 스타일은 css를 사용했고, background-image를 두가지를 사용하여 x방향으로 repeat를 하여위쪽과 아래쪽에 게임에 등장하는 캐릭터들을 배치한다.
* 배경음악 관련 속성은 bgsound와 embed 태그 중 bgsound는 화면에 아무것도 출력되지 않아 사용자가 음악을 끄기 힘들고 internet explorer에서만 작동하는 단점이 있어 embed 태그를 사용했다. Embed 태그가 크롬에서 작동이 되기는 하지만 크롬의 버전이 바뀌어서 미디어파일을 크롬에서 재생 시키려면 크롬 플러그인을 설치해야 하고, 크롬에서 embed 관련 속성을 사용하기 위해서도 크롬 플러그인을 설치해야 한다. 크롬 플러그인 설치는 굉장히 까다롭고 오래 걸리기 때문에 배경음악과 관련된 속성을 정확하게 파악하기 위해서는 internet explorer에서 확인하기를 권장한다. 이때 explorer로 열면 인코딩 문제가 발생하며 한글이 제대로 출력되지 않는(깨지는) 모습을 확인할 수 있다. 따라서 head 에서 meta charset=”UTF-8”을 추가해주어야 한다. Sound on/ off, 배경음악 자동 재생도 internet explorer에서 작동시키면 잘 작동한다.
* Sound on은 button 속성에서 onclick 시 bgm.mute = 0으로 해주고, bgm.mute=1을 해주니 배경음악이 자동으로 꺼지는 것을 확인할 수 있다.
  1. **Description page(description.html)**
* 게임에 대한 설명을 보여주는 페이지로, 게임에 대한 정보를 얻을 수 있다.
* 게임 설명은 순서가 있는 <ol>태그를 사용했고, 게임하러 가기는 button 태그를 사용했다.
* 하지만 이전처럼 button 태그를 사용하여 main\_page.html를 연결했는데도 페이지로 이동하지 않는 것을 알 수 있다.
* button태그 <form> 태그를 사용하는데 <form>태그 안에서 button은 기본적으로 submit이 default이다. 하지만 우리는 전송이 아닌 자바스크립트로 실행할 것이기 때문에 type=”button”이라는 속성을 반드시 추가해 주어야 한다.
  1. **Game rule page(gamerule.html)**
* 게임 규칙을 알려주는 페이지로, 게임을 진행하는데 필요한 규칙들을 얻을 수 있다.
* 게임 규칙도 description과 마찬가지로 순서가 있는 <ol>태그를 사용했고, 게임하러 가기도 button 태그를 사용했다.
* description과 마찬가지로 button이 <form>태그 안에 있기 때문에 자바 스크립트로 실행하기 위해서는 type=”button”이라는 속성을 반드시 추가해 주어야 한다.
  1. **Game page(Main.html)**
     1. background
* 그래픽 관련 요소(background\_dark, background\_light, character1, character2(뻐끔이), food, shark1, shark2(상어), wall 은 포토샵을 활용하여 수작업 했고, 게임 화면은 전체적으로 canvas와 javascript를 사용한다.
* 게임 화면에서 배경화면은 background\_dark 를 background\_image로 받았고 크기를 지정해주었다.
* 배경화면에서 상어와 뻐끔이가 이동할 때 필요한 벽들은 일일이 직접 계산하고 해보면서 좌표는 wall이라는 배열에 담아두고 for문을 돌려 원하는 위치에 벽을 배치시켰다. 이때 벽은Context.rect를 통해 직사각형으로 만들었고 만들어진 사각형 모양을 패턴을 채웠다. 또한 벽을 배치할 때 context.rect(1080-wall[i][0]-wall[i][2],wall[i][1]…)를 써서 좌우대칭이 되게 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 1. food 그리기
* food는 background의 벽과 마찬가지로, food를 배치할 위치를 계산하여 food라는 배열에 위치 좌표를 담아두고 for문을 돌려 그렸다.
* 이때 food는 패턴을 만들지 않아도 되고, 만들어진 food를 그리기만 하면 되기 때문에 context.drawImage을 사용한다.
* 주의해야 할 점은 상어가 먹이 위로 지나가면 먹이가 지워지면 안되는데 지워지는 문제가 발생하기 때문에 이를 방지하기 위해서 음식을 eaten이라는 배열에 담아두고 상어가 지나갈 때마다 그림을 다시 그려주어야 한다.

텍스트이(가) 표시된 사진

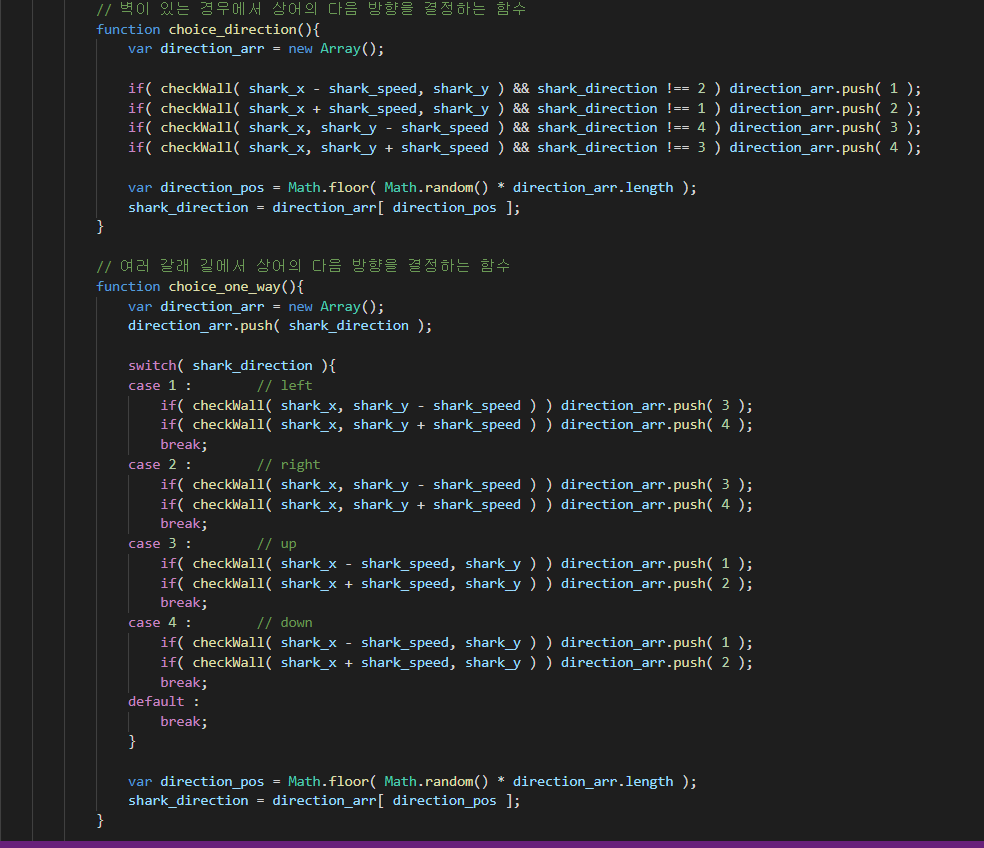
자동 생성된 설명

* + 1. 캐릭터 애니메이션
* 캐릭터가 지나가면서 뻐끔거리는 것을 표현하기 위해 gif 파일을 사용하려 했으나 제대로 표현되지 않아서 character\_animate 함수를 구현하였다.
* Character\_animate에서 character.src=char\_image[char\_index]; char\_index의 초기 값은 0이고, 0.5초마다 char\_index 값이 변경되도록 하여 캐릭터의 이미지가 0.5초마다 바뀐다. 0.5초마다 변경되는 함수는 setInterval(character\_animate, 500)을 사용했고 char\_index는 0 또는 1이다.
  + 1. 키보드 감지(keyhandler())
* 키보드 감지는 window.event를 사용하였고, keycode를 활용하여 key 값이 37이면 왼쪽 버튼, 38이면 위 버튼, 39면 오른쪽 버튼, 40이면 아래쪽 버튼을 감지할 수 있도록 하였다.
* keyhandler()라는 함수는 만들어 이벤트가 발생하면 그림을 그리는 draw 함수를 호출하여 주인공 캐릭터의 그림을 그린다.
  + 1. draw
* 만든 draw() 함수에서는 이벤트가 발생하면 그림을 그리는데, 기존의 그림을 지우고 새로 그림을 그리게 된다.
* 기존의 그림을 지울 때는 context.clearRect(x, y, width, height)을 사용했고, event가 발생했을 때 받아왔던 keycode를 사용해 그림을 다시 그려주었다. 캐릭터를 다시 그릴 때는 이동을 해서 다시 그리는 것이기 때문에 ‘y좌표 = 기존 y좌표 – 캐릭터 속도(얼마나 이동)’ 로직을 사용해 그림을 다시 그린다.
* draw()에서 캐릭터가 이동할 때마다 먹이를 먹었는지 확인을 해야 하기 때문에 checkEaten() 함수를 불러주어야 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 1. 벽에 닿았는지 체크(checkWall())
* 그림에 벽을 세웠기 때문에 상어나 캐릭터가 벽을 통과하면 안되고 벽에 닿으면 못지나가게 하기 위해 벽에 닿았는지 체크하는 함수가 필요하다.
* checkwall안에 있는 left, top, right, bottom은 벽의 좌상단, 우상단, 좌하단, 우하단을 나타낸다. X,y는 캐릭터의 x좌표, y좌표를 나타내는데 캐릭터의 x좌표가 right보다 작고, 벽두께를 포함하여 x좌표가 left보다 크고 y좌표가 bottom보다 작고 y좌표가 벽두께를 포함하여top 보다 크다면 벽에 위치에 있다는 것이므로 false를 리턴해준다. 반대쪽도 마찬가지로 계산하여 벽의 위치를 계산하여 벽에 위치에 있으면 false를 리턴해준다.
  + 1. 먹이를 먹었는지 체크(checkEaten(x,y))
* 먹이를 먹었는지 체크하여 먹이의 그림을 다시 그려야 하기 때문에 먹이를 먹었는지 체크하는 함수가 필요하다.
* 로직은 벽에 닿았는지 체크하는 함수의 로직과 동일하다
* 먹이를 모두 먹었을 경우 cleared()라는 함수를 호출한다.
  + 1. 상어를 그리고 이동시키는 함수(shark\_draw())
* 상어는 자동으로 움직일 수 있게 하였으며 0.5초마다 이동한다.
* 우선 벽이 있는지 체크하고 벽이 없으면 먹이를 다시 그리고 상어도 다시 그린다.
* 벽이 있는 경우 상어는 갈 수 있는 방향을 checkwall이라는 함수에서 탐색한다. checkwall에서 true가 나오면 그 방향으로 나갈 수 있는 것을 의미하고 우선 갈 수 있는 방향을 모두 배열에 넣어둔다. 이때 왔던 길을 되돌아가지 못하도록 왔던 길의 방향은 배열에 넣지 않는다.
* 저장된 배열(방향은 1,2,3,4로 표시한다)에서 랜덤으로 방향을 하나 찾아서 상어는 그 방향으로 이동하게 된다.



* + 1. 캐릭터와 상어의 충돌을 감지하는 함수(crash\_cha\_shark())
* 캐릭터와 상어의 위치를 계산하여 각각의 x좌표, y좌표를 비교하여 동일하면 true를 리턴해주고 game over라는 메시지를 띄운다.
* 충돌이 감지되면 life의 수를 줄이고 life가 0이면 restart() 함수를 호출하고 상어와 캐릭터를 다시 원래위치로 보낸다.