

11장 이차원 배열 다루기: 틱택토

11.1 이 장에서 만드는 프로그램

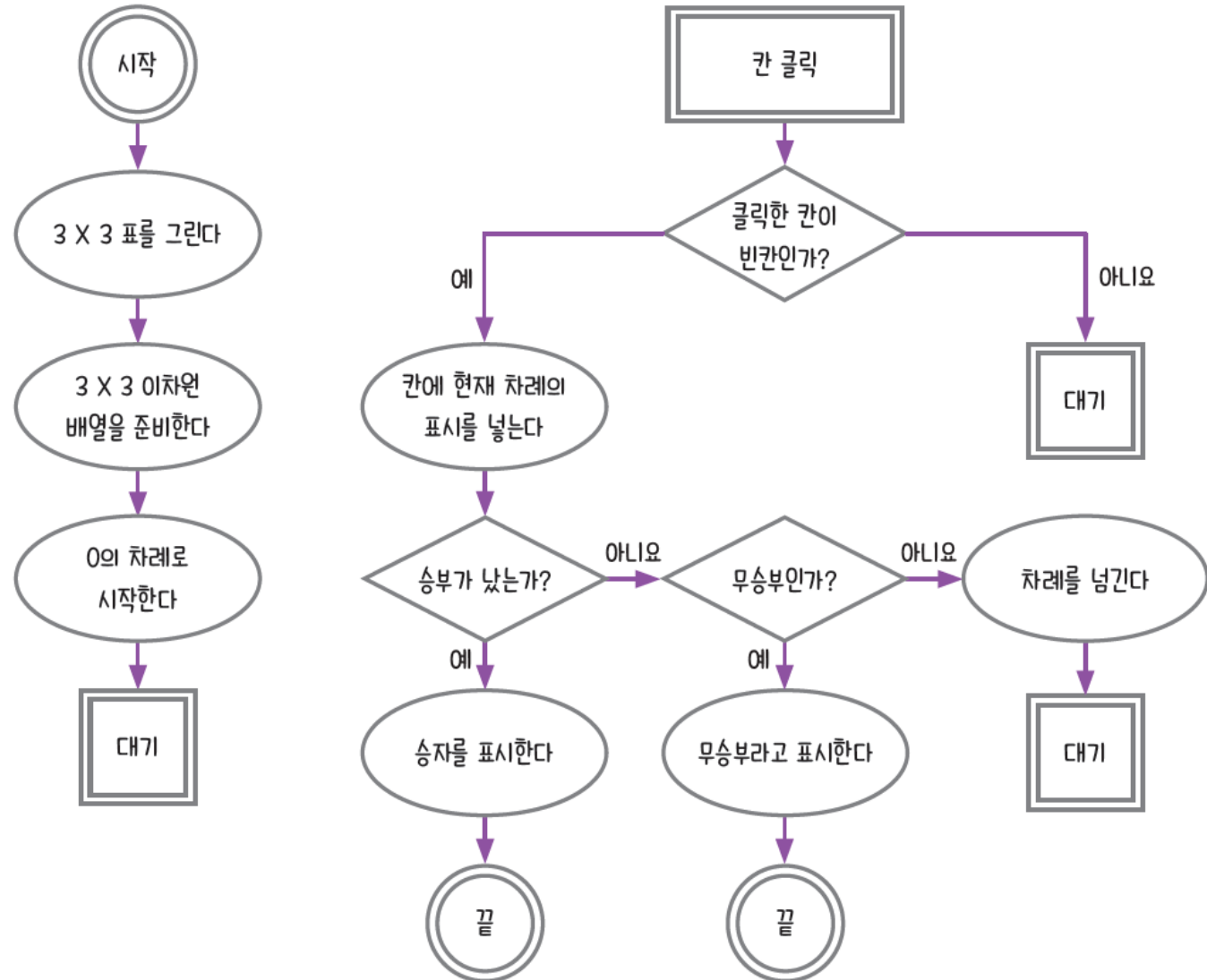
11.2 화면에 3 X 3 표 그리기

11.3 차례 바꾸면서 빈칸에 ○와 X 표시하기

11.4 승패 판단하기

11.1 이 장에서 만드는 프로그램

- 틱택토의 순서도



11.2 화면에 3 X 3 표 그리기

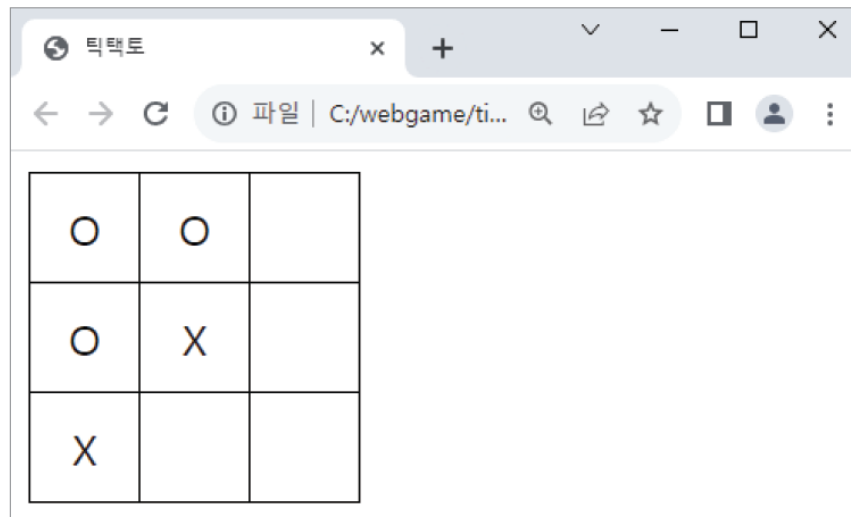
- 틱택토 형태

○		○
○	x	
x		

- 표 형태를 표시할 때는 자바스크립트에서 이차원 배열 사용
- 3줄 3칸으로 이루어진 표를 이차원 배열로 구성
- table은 전체 표를 담당하는 태그, tr은 표의 가로줄, td는 각 칸을 담당

11.3 차례 바꾸면서 빈칸에 O와 X 표시하기

- 화면에서 표의 빈칸을 클릭하면 차례대로 O와 X가 입력되도록 작성
 - 차례(turn)는 O부터 시작
 - 클릭한 칸이 비어 있는지 확인
 - 클릭한 칸에 O나 X를 입력하도록 td 태그에 클릭 이벤트 리스너 붙임
 - callback() 함수 마지막에서 turn이 O면 X로, X면 O로 바꿈



11.4 승패 판단하기

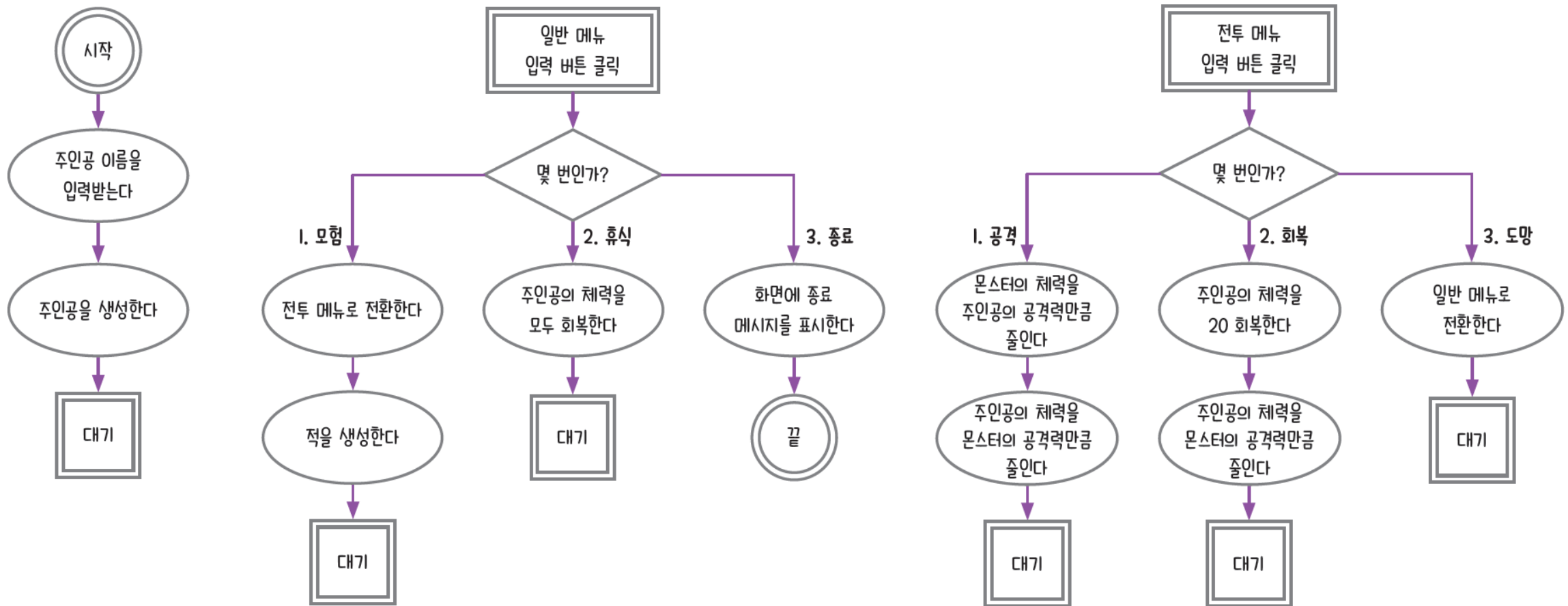
- 가로줄, 세로줄, 대각선 3칸에 같은 모양이 오면 승리
- 승패가 가려지면 누가 승자인지 div 태그를 만들어 표시
- 9칸을 다 채웠는데도 승자가 없다면 '무승부'라는 메시지 표시
 - 승패를 판단하는 checkWinner() 함수 만들기
 - callback() 함수에서는 checkWinner() 함수를 event.target과 함께 호출
 - 승자가 없는 경우에는 무승부인지 판단
 - 이차원 배열에서는 반복문(forEach())을 중첩해서 사용하게 되므로 flat()과 every() 메서드를 사용해 검사

12장 클래스 다루기: 텍스트 롤플레잉 게임

- 12.1 이 장에서 만드는 프로그램
- 12.2 초기 화면 만들고 일반 메뉴 화면으로
전환하기
- 12.3 주인공, 몬스터, 일반 메뉴 구현하기
- 12.4 전투 메뉴 구현하기
- 12.5 클래스로 코드 재구성하기
- 12.6 전투 결과 표시하기

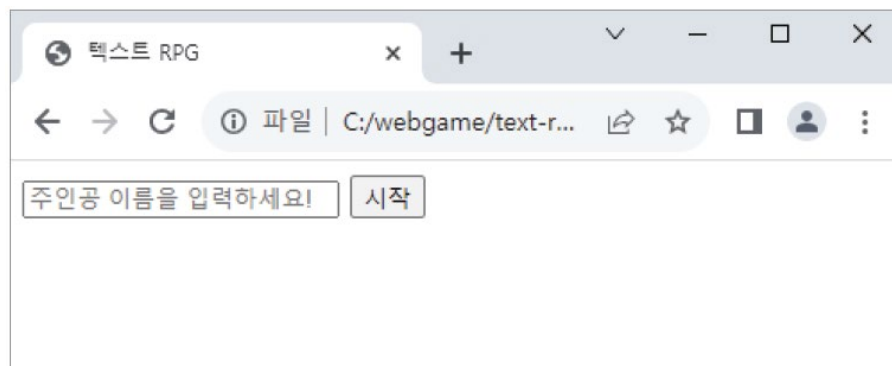
12.1 이 장에서 만드는 프로그램

- 텍스트 RPG 순서도

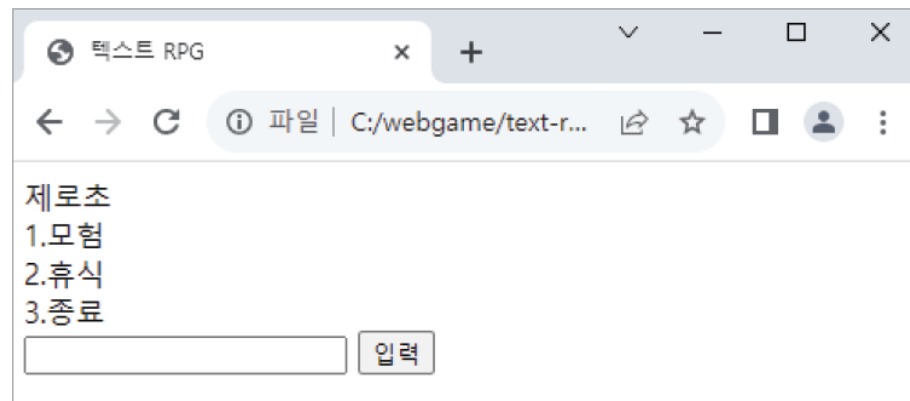


12.2 초기 화면 만들고 일반 메뉴 화면으로 전환하기

- 초기 화면: 사용자에게 이름을 입력받아 주인공을 생성



- 주인공 이름을 입력하고 나면 일반 메뉴 화면으로 전환



12.3 주인공, 몬스터, 일반 메뉴 구현하기

- 주인공 정보 작성
 - 객체 리터럴로 만들기
 - lev는 레벨, maxHp는 최대 체력, hp는 현재 체력, xp는 경험치, att는 공격력
 - hp가 0이 되면 게임 오버
 - xp가 일정 수준에 도달하면 레벨 업하고 체력과 공격력이 올라감
- 주인공 정보 표시

제로초 1Lev HP: 100/100 XP: 0/15 ATT: 10

1.모험

2.휴식

3.종료

입력

- 몬스터 정보 작성
 - 객체 리터럴로 만들어 monsterList에 저장

12.3 주인공, 몬스터, 일반 메뉴 구현하기

- 일반 메뉴에서 1번 모험을 선택하면 화면이 전투 메뉴로 바뀜
- 상대할 몬스터는 monsterList에서 무작위로 가져옴
- 몬스터 정보 표시

제로초 1Lev HP: 100/100 XP: 0/15 ATT: 10

1.공격

2.회복

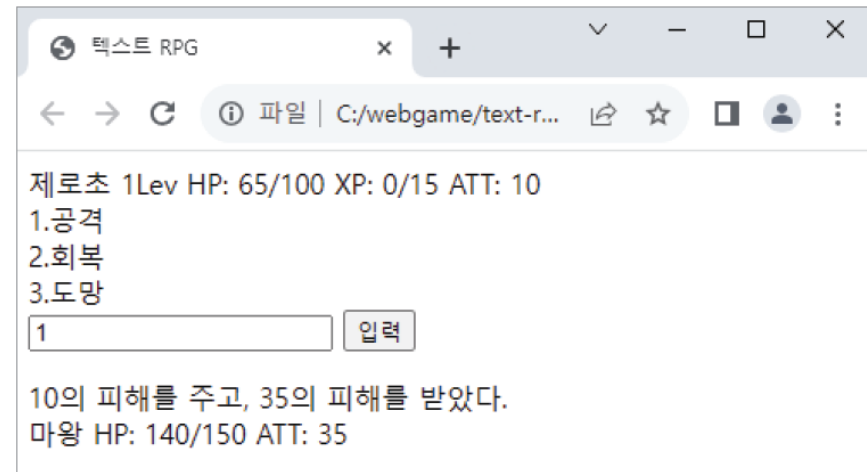
3.도망

입력

마왕 HP: 150/150 ATT: 35

12.4 전투 메뉴 구현하기

- 몬스터와 공격을 주고받거나 회복 또는 도망가기 구현
 - 주인공 객체와 몬스터 객체 간의 상호 작용은 어떤 객체가 다른 객체에 특정한 기능을 수행할 때 일어남
 - 기능은 함수(메서드) 구현
 - attack() 메서드: hero 객체에 몬스터를 공격하는 기능
 - heal() 메서드: 체력을 회복하는 기능
- 서로 공격을 주고받은 결과 화면 표시



12.5 클래스로 코드 재구성하기

12.5.1 클래스 생성하기

- Game 클래스
 - 게임 자체를 클래스로 만들기
 - 주인공 이름을 입력받을 때 `new Game(name)`으로 게임 클래스를 호출해 게임 객체 생성
 - `new Game`을 하면 Game 클래스의 constructor 메서드가 실행되고 `monster`, `hero`, `monsterList` 속성을 만듦
- Unit 클래스
 - 주인공과 몬스터에 공통 속성과 메서드를 한 클래스에 모으기
 - 공통된 속성(`name`, `maxHp`, `hp`, `xp`, `att`)
 - 공통된 메서드(`attack()`)

12.5 클래스로 코드 재구성하기

- Hero 클래스
 - 주인공을 클래스로 만들기
 - `super()` 함수를 통해 부모 클래스인 Unit 클래스에 있는 속성을 상속받아 사용
 - 부모 클래스에 존재하지 않는 `lev` 속성과 `heal()` 메서드는 `super()` 함수 아래 따로 작성
- Monster 몬스터
 - 몬스터를 클래스로 만들기
 - Unit 클래스에 있는 속성과 메서드를 그대로 사용
 - 부모 클래스 내용을 그대로 사용할 때는 자식 클래스에서 코드 생략

12.5 클래스로 코드 재구성하기

12.5.2 클래스에 메서드 추가하기

- Game 클래스 안에 기본 메서드 작성
- start() 메서드: 게임시작
- changeScreen() 메서드: 화면 전환
- onGameMenuInput() 메서드: 일반 메뉴 담당
- onBattleMenuInput() 메서드: 전투 메뉴 담당
- 객체의 메서드는 this를 통해 서로 호출 가능

12.5 클래스로 코드 재구성하기

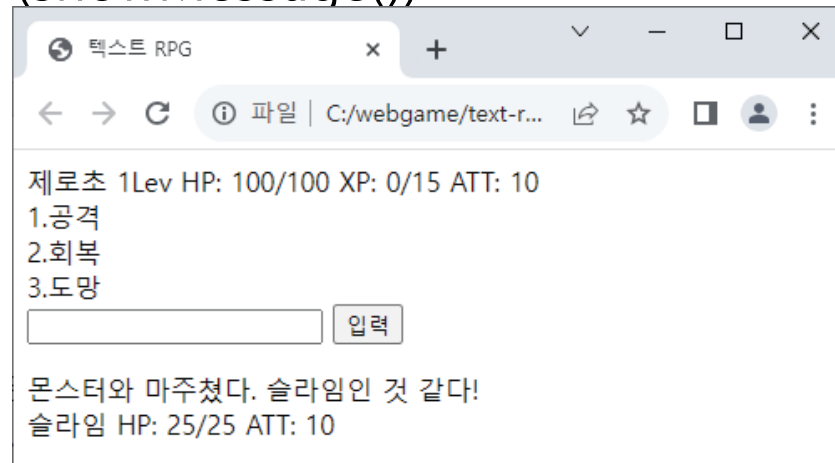
12.5.3 클래스 간 연동하기

- 클래스 간의 상호 작용
- start() 메서드에서 주인공 생성
 - constructor() 메서드에서 전달받은 주인공의 이름(name)을 start() 메서드를 거쳐 Hero 클래스에 전달
 - 주인공이 Game 클래스의 hero 속성에 등록됨
- updateHeroStat() 메서드
 - 주인공 생성 시 주인공의 체력, 공격력, 경험치가 화면에 표시
 - 주인공이 죽으면 주인공 정보를 지움

12.5 클래스로 코드 재구성하기

12.5.4 일반 메뉴 재구성하기

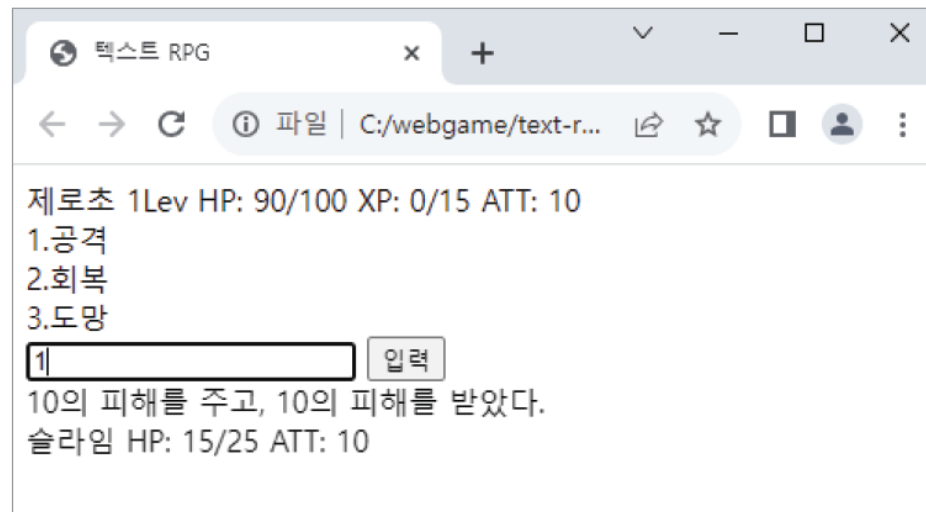
- 일반 메뉴에서 1(모험)을 입력했을 때
 - onGameMenuInput() 메서드 실행
 - 화면이 전투 메뉴로 전환
 - 몬스터 생성(createMonster())
 - 몬스터 정보를 화면에 표시(updateMonsterStat())
 - 몬스터가 나타났다는 메시지를 화면에 표시(showMessage())



12.5 클래스로 코드 재구성하기

12.5.5 전투 메뉴 재구성하기

- 전투 메뉴에서 1(모험)을 입력했을 때
 - 몬스터 공격
 - 공격 결과를 메시지로 표시
 - 주인공과 몬스터 정보 업데이트



12.6 전투 결과 표시하기

- 주인공과 몬스터가 서로 공격하다가 하나가 체력 0이 되면 전투 끝
- 주인공 체력 0이면 게임 오버, 몬스터 체력 0이면 주인공 경험치 얻기
- 게임 오버
 - Game 클래스에 quit() 메서드로 구현
 - 주인공과 몬스터 객체를 null로 만들고, 상태 표시 업데이트
 - removeEventListener() 메서드로 이벤트 모두 제거
 - 초기 화면으로 돌아가서 게임 객체를 null로 만들고 새 게임 준비
- 레벨 업
 - Hero 클래스에 getXp() 메서드로 구현
 - 주인공의 경험치가 주인공 레벨 × 15보다 높으면 레벨 업
 - 레벨 업 하면 주인공 체력 모두 회복
 - 최대 체력과 공격력이 각각 5씩 증가

12.6 전투 결과 표시하기

- 게임 오버되거나 레벨업 했을 때

