

15장

진급 프로젝트: 게임북 본뜬 게임 만들기

15장 **진급 프로젝트:게임북 본뜬 게임 만들기**

15.1 게임 규칙 소개

15.2 여러 다른 경로 만들기

15.3 더 많은 선택? 좋지요!

15.4 요약



15 프로젝트 문제

» 조건문과 분기를 사용해 이야기를 만들어 보자.

- 각 장면에서 사용자는 단어를 하나 입력한다. 프로그램은 사용자가 입력한 단어에 따라 다음 진행을 결정한다.
- 사용자가 선택할 수 있는 가능성을 모두 다루도록 프로그램을 만들어야 한다.
- 다만 입력하라고 요구하지 않은 단어를 사용자가 입력한 경우까지 처리할 필요는 없다.

15.1 게임 규칙 소개



15.1 게임 규칙 소개

- » 사용자에게 입력을 받을 때 유의할 점은 사용자가 프로그램이 요청한 규칙을 지키지 않을 수 있다는 것이다.
- » 프로그램은 사용자에게 어떤 입력을 받을지 명확히 밝히고, 요청하지 않은 입력이 들어올 경우에는 프로그램이 중단될 수 있다고 경고해야 한다.
- » 다음과 같이 간단한 print 문이면 충분하다

```
print("당신은 지금 무인도에 있습니다.")
print("구조될 때까지 살아 남도록 노력하세요!")
print("사용할 수 있는 명령은 (LOOK)처럼 괄호 안에 대문자로 표시됩니다.") ---- 게임 진행 방법
print("명령어를 잘못 입력하면 프로그램이 끝납니다. 조심하십시오.") ---- 잘못된 입력 시 프로그램이 종료됨
print("우선 주변을 둘러봅니다(LOOK)..." )
```

코드 15-1
사용자 입력 받기

15.2 여러 다른 경로 만들기



15.2 여러 다른 경로 만들기

» 프로그램의 일반적인 흐름

- 사용자에게 어떤 선택지가 있는지 설명한다.
- 사용자 입력을 받는다.
- 사용자가 첫 번째 선택지를 고르면 그에 따른 메시지를 출력한다. 첫 번째를 선택하면 사용자에게는 새로운 선택지가 생겨난다. 다시 사용자 입력을 받고, 그에 따라 다음 단계를 진행하는 과정을 반복한다.
- 사용자가 첫 번째가 아닌 두 번째 선택지를 고르면 다른 메시지를 출력한다. 이 경우에도 사용자에게는 다른 새로운 선택지가 생겨난다. 다시 사용자 입력을 받고 그에 따라 다음 단계를 진행한다.
- 세 번째나 네 번째 등 다른 선택지가 있고 사용자가 그 중 하나를 고르면 그에 따라 새로운 선택지가 생겨난다. 다시 사용자 입력을 받고, 그 과정을 반복한다.



15.2 여러 다른 경로 만들기

```
do = input(":: ") ---- 사용자 입력 받기
if do == "LOOK": ---- 사용자가 LOOK을 입력했는지 검사하는 조건문
    print("모래 독이 있는 도랑에 갇혔습니다.")
    print("왼쪽(LEFT)이나 오른쪽(RIGHT)으로 엉금엉금 기어갑니다.")

do = input(":: ") ---- LOOK으로 주변을 둘러본 후 다시 사용자 입력 받기
if do == "LEFT": ---- 사용자가 LEFT를 입력했는지 검사하는 조건문
    print("도랑 밖으로 기어 나왔습니다. 배가 있네요!")
    print("살아 남았습니다!")
elif do == "RIGHT": ---- 사용자가 RIGHT를 입력했는지 검사하는 조건문
    print("아! 독을 올라갈 수가 없습니다. 오른쪽 독은 아주 미끄럽습니다.")
    print("멀리 미끄러져 내려가 이상한 동굴로 떨어졌습니다.")
    print("살아남지 못했습니다 :(")
else: ---- 대문자로 표시된 단어만 입력하라고 알려주는 코드 블록
    print("대문자로 표시된 동작만 입력해 주십시오.")
    print("다시 한번 도전해 보십시오!")
```

코드 15-2

선택지가 한 가지 뿐인 모험



15.2 여러 다른 경로 만들기

» 코드 15-2는 사용자 입력을 요청하면서 시작한다.

- 그 후 사용자가 게임의 규칙을 제대로 이해하고 LOOK이라는 키워드를 입력했는지 조건문을 통해 검사한다.
- 사용자가 다른 단어를 입력했다면 사용자에게 화면에 표시된 명령어만 입력하라고 알려준다.
- 사용자가 LOOK을 입력했다면 프로그램은 사용자에게 LEFT나 RIGHT 중 하나를 입력하라고 요청한다.
- 그리고 사용자가 LEFT나 RIGHT 중 어떤 것을 입력 했느냐에 따라 다른 메시지를 출력한다.

15.3 더 많은 선택? 좋지요!



15.3 더 많은 선택? 좋지요!

» 이런 게임북 형태의 모험 게임에는 선택이 더 많아야 한다.

- 조건문을 많이 내포시키면 다양한 하위 경로를 만들 수 있다. 원하는 만큼 게임을 쉽게도 어렵게도 만들 수 있다.

» 그림 15-1은 이런 게임에서 가능한 코드 구조의 한 가지 예다.

- 각 선택은 사용자가 입력한 단어로 표시된다. 사용자는 선택한 단어에 따라 그 경로의 새로운 상황을 볼 수 있고, 이 과정을 최종 결과에 도달할 때까지 반복한다.

» 이 상황을 그림 15-1 코드에 이어서 작성해 보았다. 오직 한 경로만 생존으로 연결된다.

- 사용자는 LEFT, CRAB, YES, TREE, NO를 차례로 입력해야만 살 수 있다.
- 다른 선택은 모두 죽음을 의미 하는 메시지로 끝나버린다.

15.3 더 많은 선택? 좋지요!

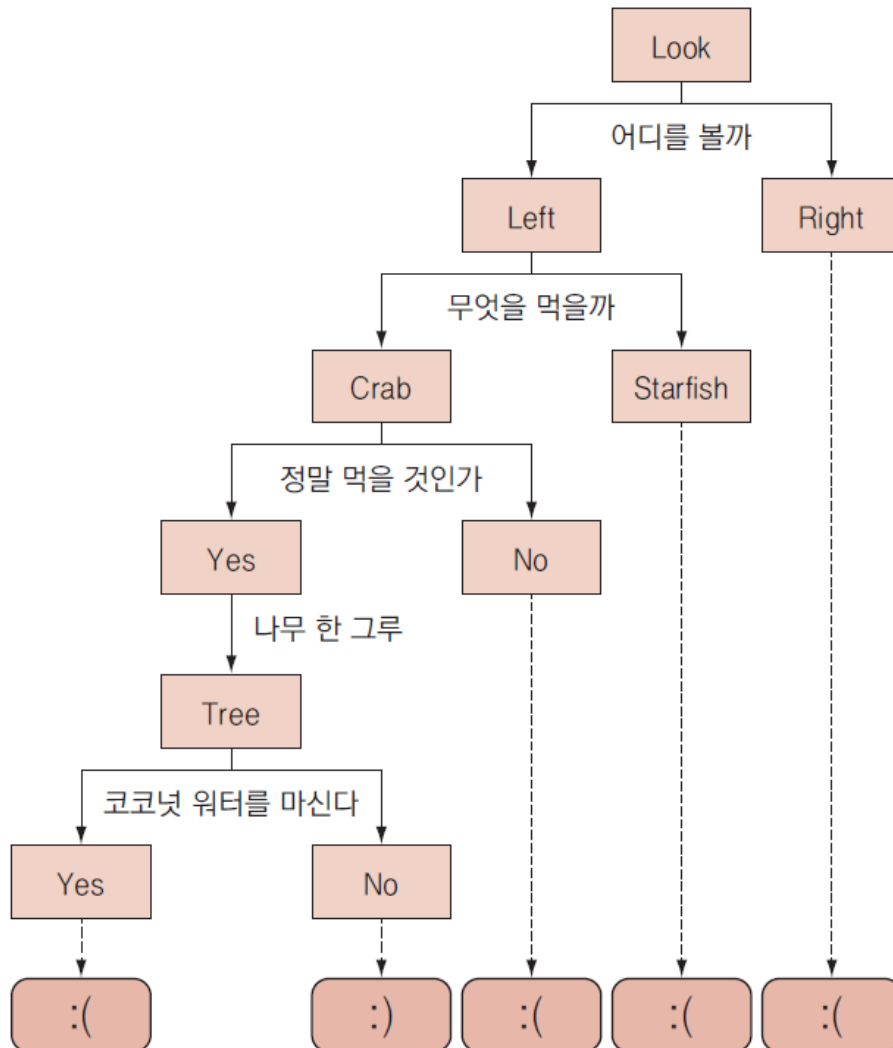


그림 15-1

박스는 사용자의 선택을,
박스 아래에 있는 텍스트는 그 선택에 의해
벌어지는 상황을 표현한다.

실선 화살표는 선택을, 웃거나 우는 얼굴로 이어진
점선 화살표는 프로그램의 끝을 의미한다.

다섯 가지 결과 중에서 단 한가지만 생존으로 연결된다



15.3 더 많은 선택? 좋지요!

```
print("모래 독이 있는 도랑에 갇혔습니다.")
print("왼쪽(LEFT)이나 오른쪽(RIGHT)으로 엉금엉금 기어갑니다.")

do = input(":: ")
if do == "LEFT": ---- ❶ 첫 번째 선택지
    print("모래밭에 불가사리(STARFISH)와 게(CRAB)가 있습니다.")
    print("배가 고픽니다! 어떤 동물을 먹겠습니까?")

    do = input(":: ")
    if do == "STARFISH": ---- ❷ ❶의 if 분기를 위한 첫 번째 선택지
        print("에그머니나! 바로 몸이 아프기 시작합니다.")
        print("살아남지 못했습니다 :(")
    elif do == "CRAB": ---- ❸ ❶의 if 분기를 위한 두 번째 선택지
        print("게를 날로 먹어도 별 문제 없을까요? 예(YES)/아니오(NO).")

    do = input(":: ")
    if do == "YES": ---- ❹ ❸의 if 분기를 위한 첫 번째 선택지
        print("좋습니다. 게를 날로 먹었습니다. 행운을 빕니다.")
```



15.3 더 많은 선택? 좋지요!

```

print("배에 음식이 들어가니 갑자기 나무(TREE)가 눈에 띄니다.")
do = input(":: ")
if do == "TREE": --- ⑤ TREE만 유일한 선택지임

    print("코코넛 나무입니다! 게다가 갑자기 목도 마르네요.")
    print("코코넛 열매에서 코코넛 워터를 마시겠습니까? 예(YES)/아니오(NO).")

    do = input(":: ")
    if do == "YES":
        print("아이구 이런. 코코넛 워터와 게는 잘 어울리지 않네요")
        print("살아남지 못했습니다 :(")
    elif do == "NO":
        print("좋은 생각입니다!")
        print("보십시오! 구조하러 비행기가 왔습니다! 해내셨군요! \o/")

elif do == "NO": --- ⑦ ③의 if 분기를 위한 두 번째 선택지
    print("하지만 먹을 만한 것이 아무 것도 없습니다.")
    print("살아남지 못했습니다 :(")
elif do == "RIGHT": --- 두 번째 선택지
    print("아! 독을 올라갈 수가 없습니다. 오른쪽 독은 아주 미끄럽습니다.")
    print("멀리 미끄러져 내려가 이상한 동굴로 떨어졌습니다.")
    print("살아남지 못했습니다 :(")

```

⑥ ⑤에 대한 두 선택지

코드 15-3
게임복을 본뜬
모험 게임의 한 예

15.4 요약



15.4 요약

- » 조건문은 사용자에게 선택지를 제공한다
- » 조건문을 내포시키면 사용자가 선택한 내용에 따라 다시 다른 선택지를 추가 제공할 수 있다.