



# 자바스크립트 심화

## 04 행맨 게임 만들기

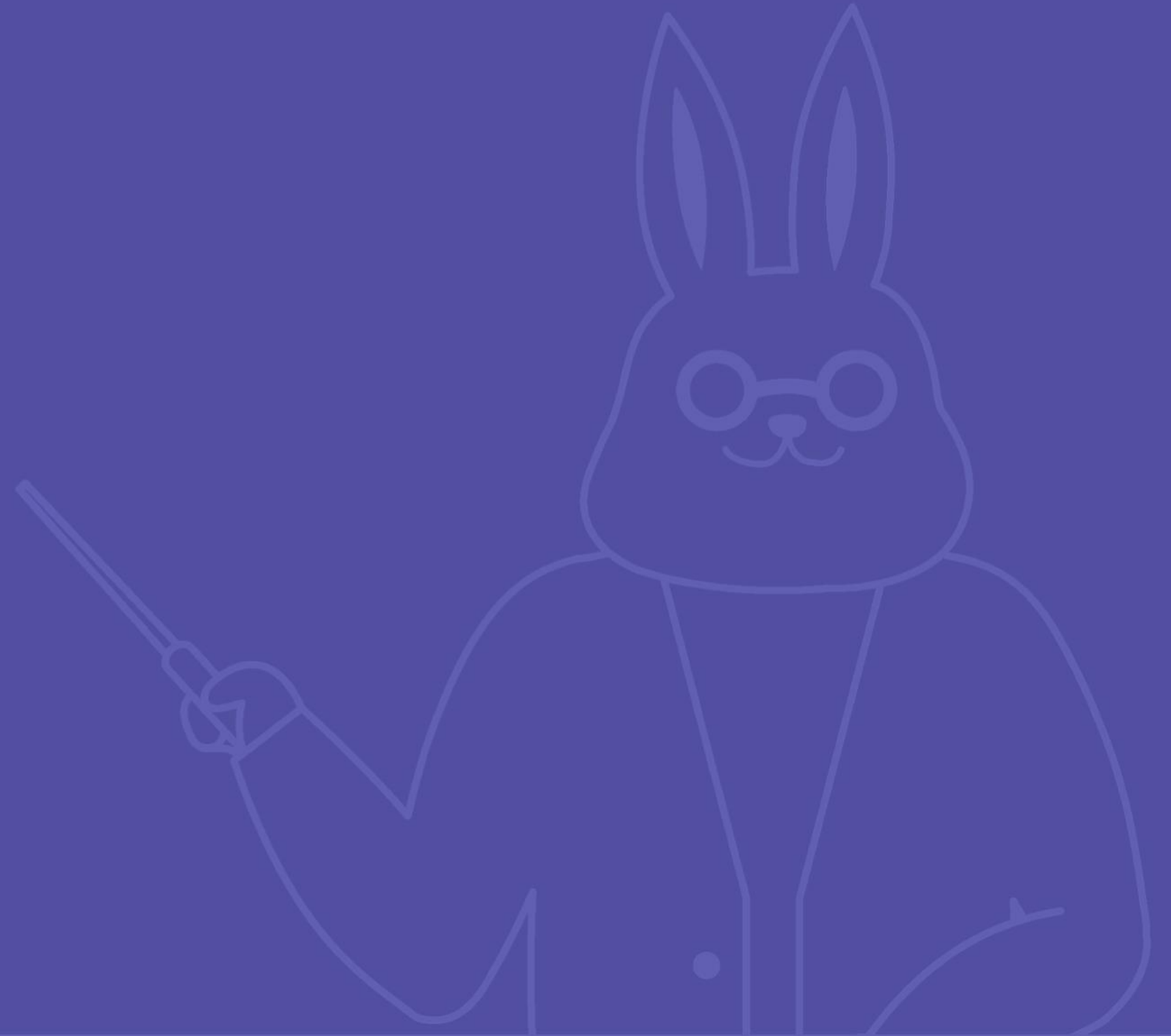


## 목차

- 01. 행맨 게임 소개
- 02. 행맨 게임 코드 구조 - 상태 관리
- 03. 행맨 게임 코드 구조 - 컴포넌트
- 04. 행맨 게임 코드 구조 - 게임 로직

01

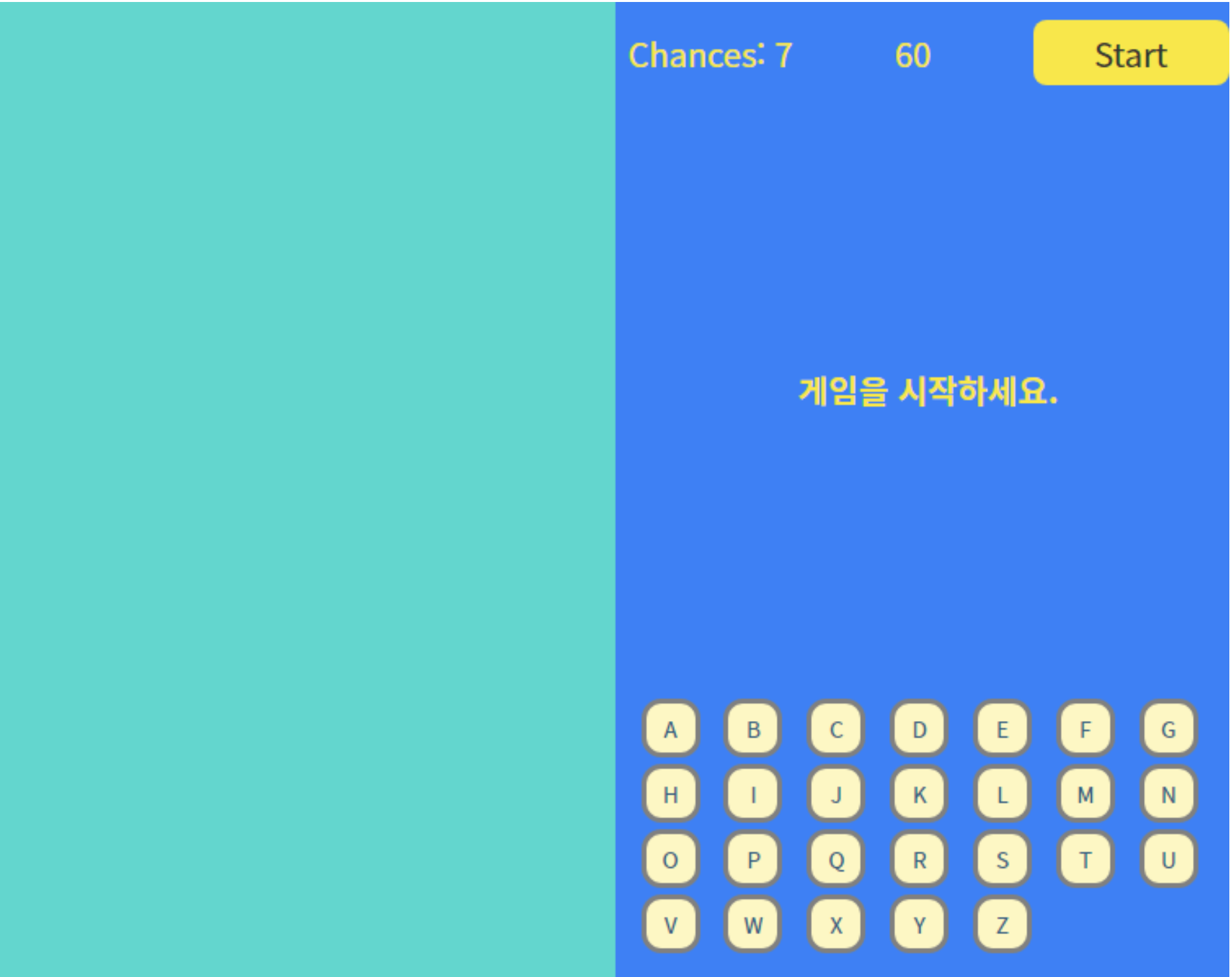
# 행맨 게임 소개



## ✓ 행맨 게임 소개

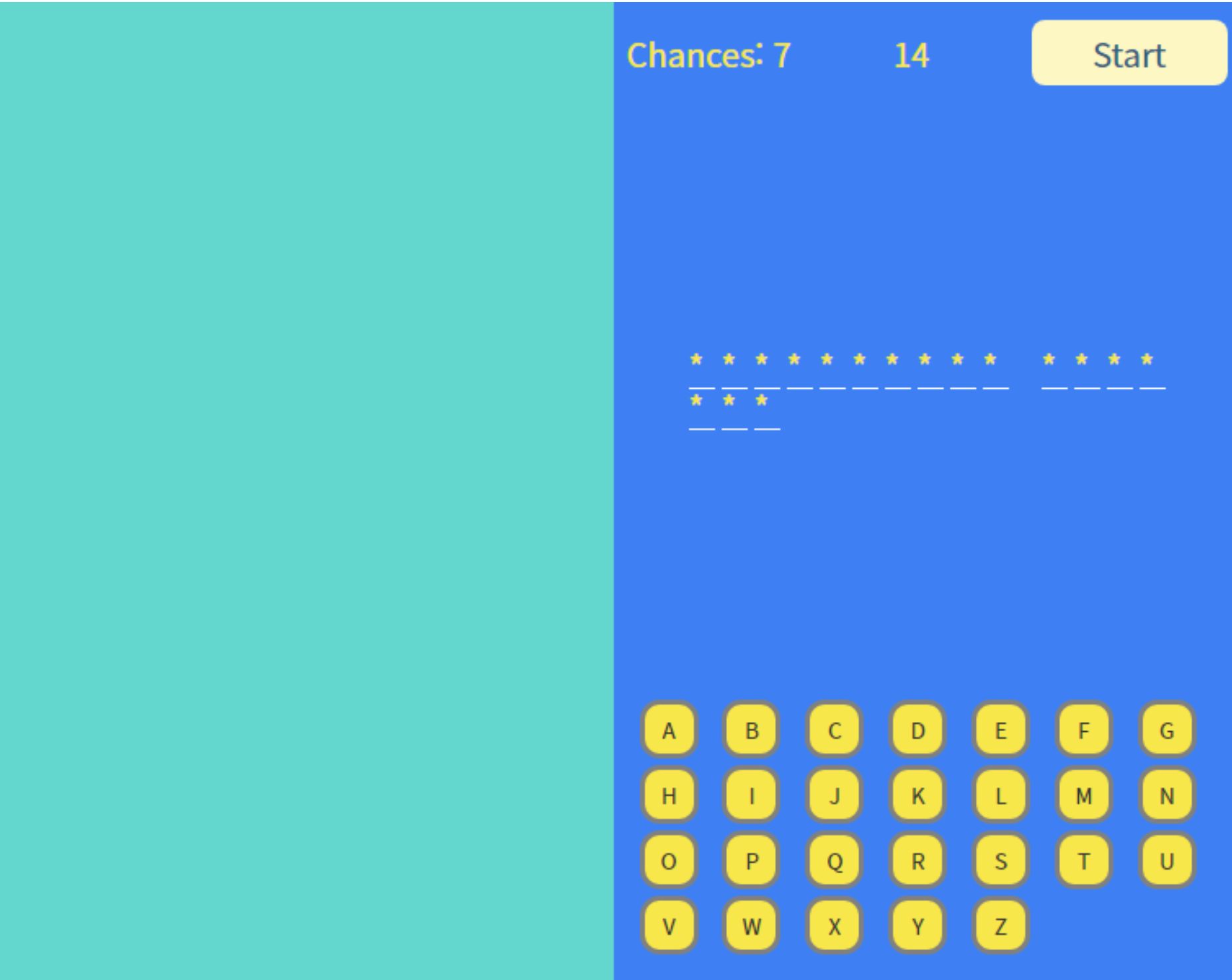
- 알파벳을 하나씩 선택하여 주어진 기회 안에 단어를 맞추는 게임.
- 초기에는 단어의 길이와 공백 정도의 정보만 주어진다.
- 알파벳을 하나씩 선택할 때마다, 그 알파벳이 들어 있는 위치가 드러난다.
- 단어에 들어있지 않은 알파벳을 선택하면 점차 행맨이 완성된다.
- 총 7번의 기회, 혹은 60초 안에 단어를 맞추지 못하면 패배한다.

✓ 행맨 게임 소개



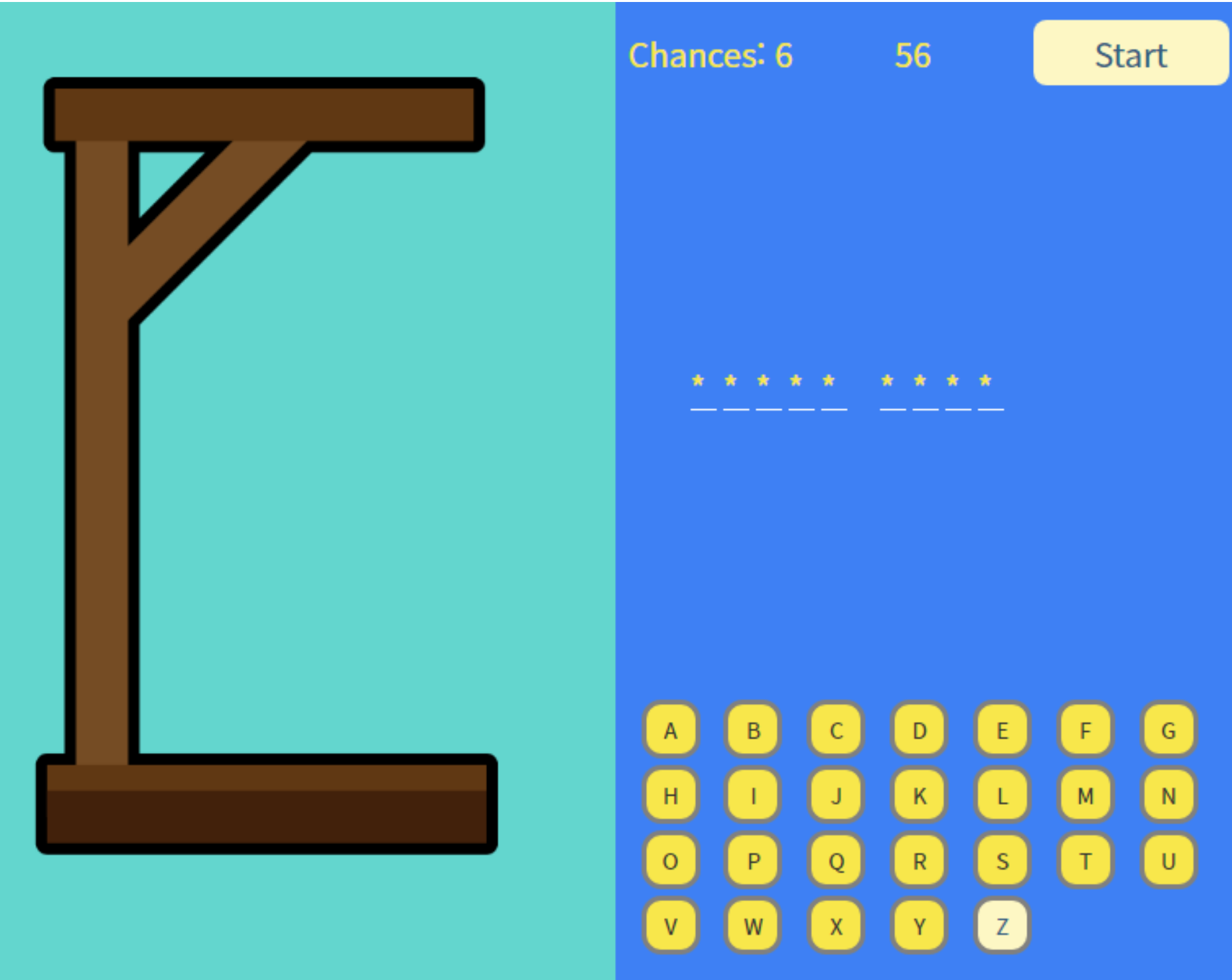
- 초기 시작 화면.
- START를 누르면 게임이 시작된다.

✓ 행맨 게임 소개



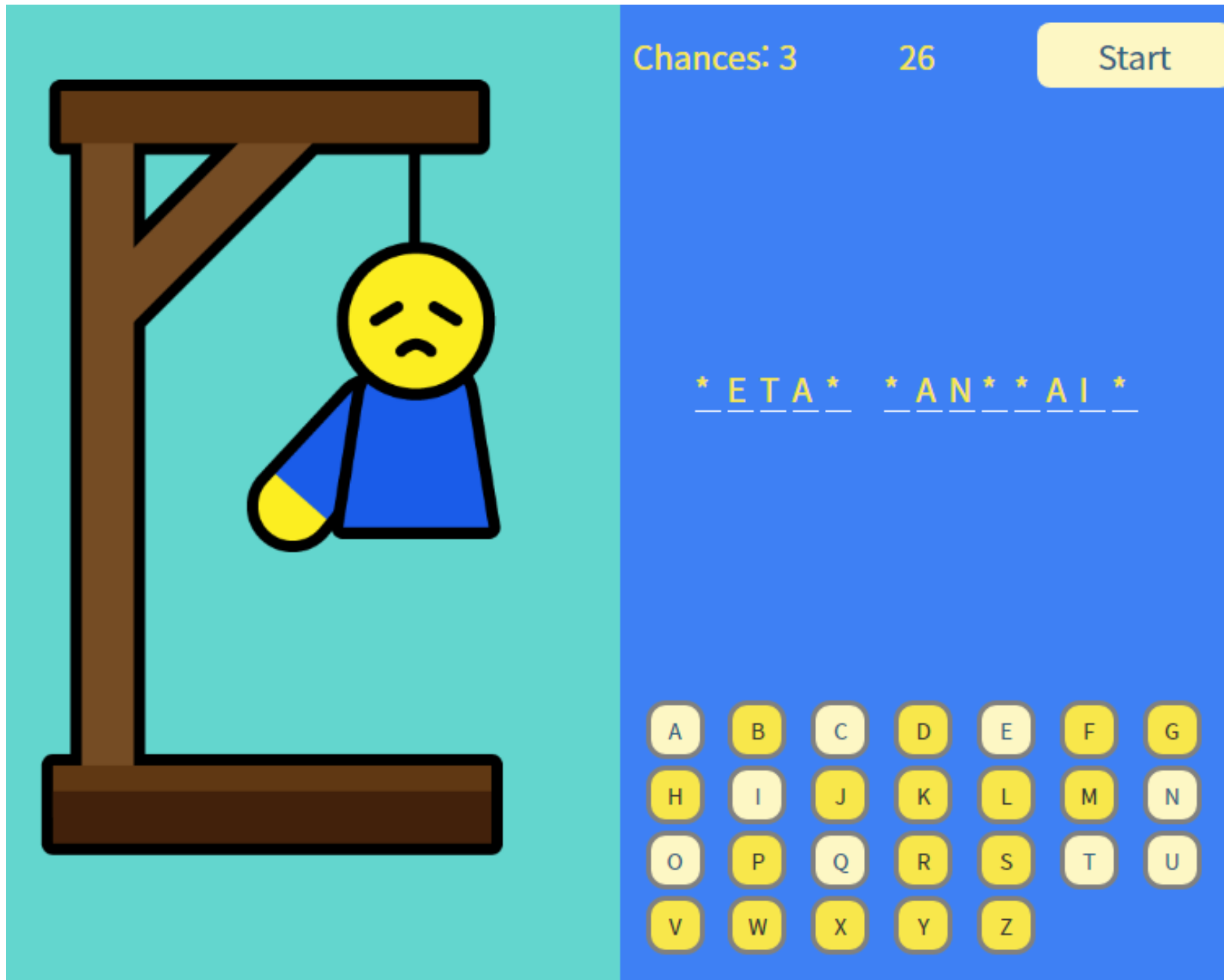
- 게임이 시작된 화면.
- 단어의 길이와 선택할 수 있는 알파벳들이 주어진다.
- 남은 기회, 남은 시간이 함께 표시된다.

✓ 행맨 게임 소개



- 잘못된 알파벳을 선택하면 행맨이 만들어진다.
- 7번의 기회가 주어진다.

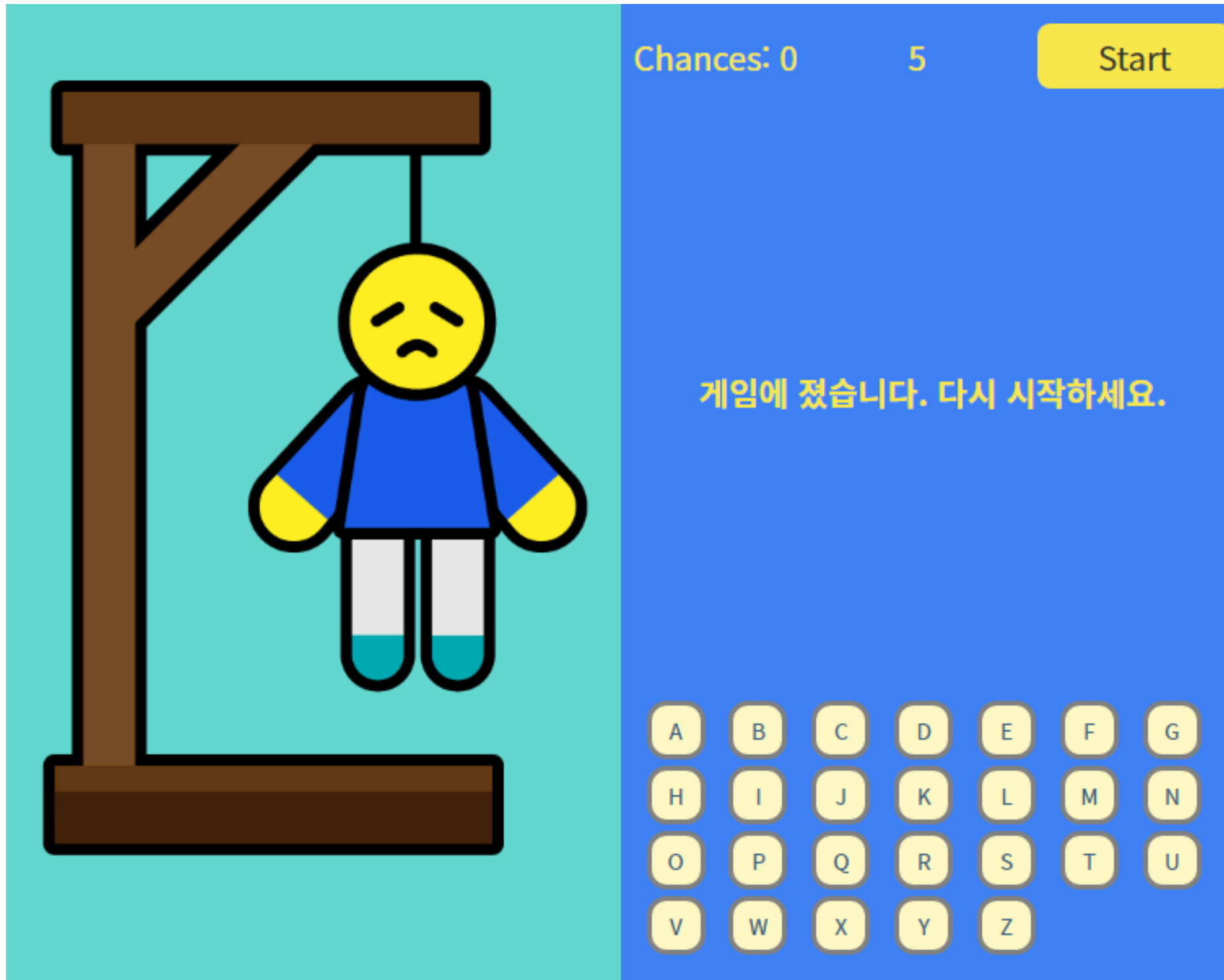
## ✓ 행맨 게임 소개



- 단어에 속한 알파벳을 선택하면, 그 위치가 드러난다.
- 단어에 들어 있는 모든 알파벳을 선택하면 게임에서 승리한다.

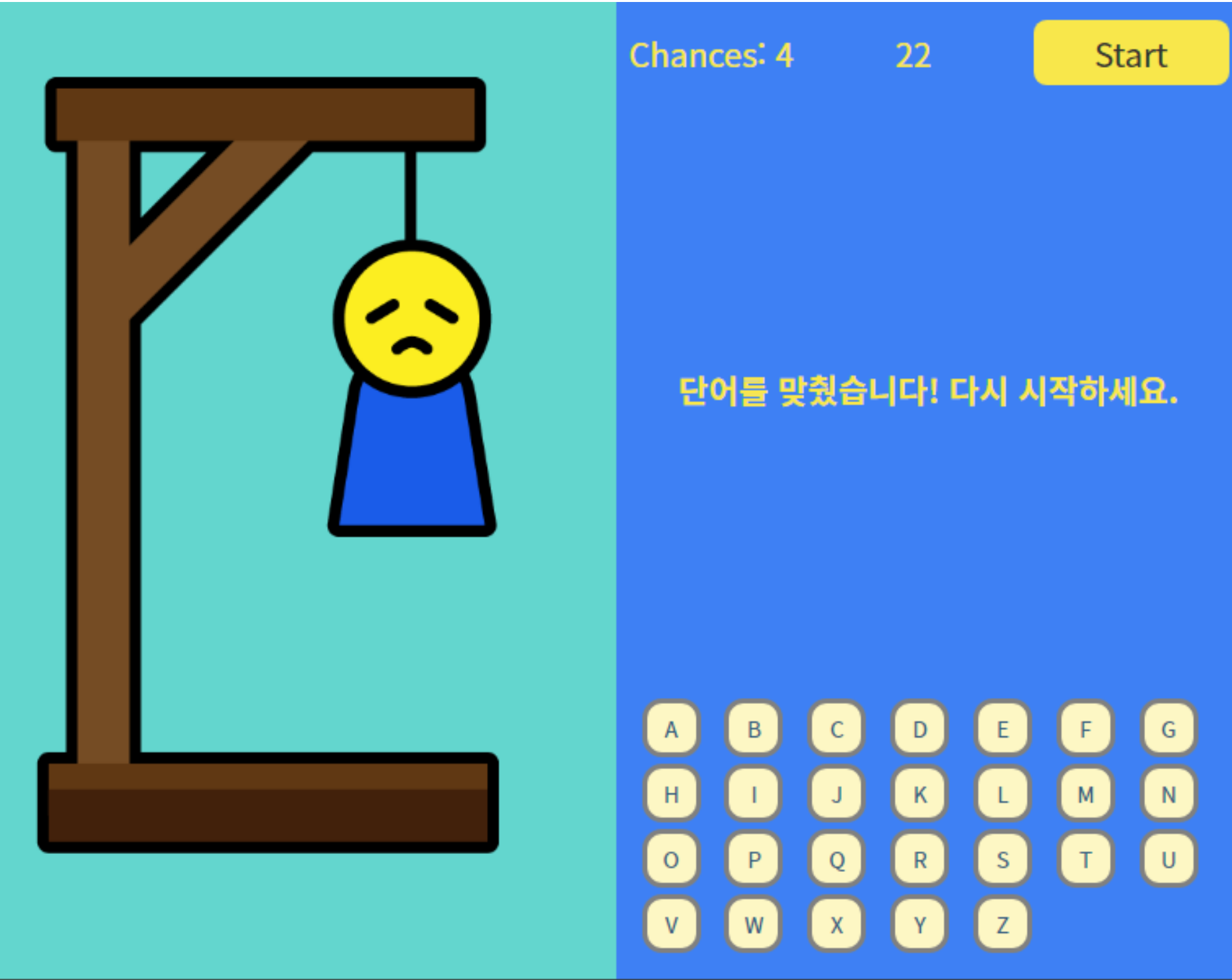


## ✓ 행맨 게임 소개



- 기회를 모두 잃거나, 시간이 지나면 게임에서 패배하고 행맨이 완성된다.

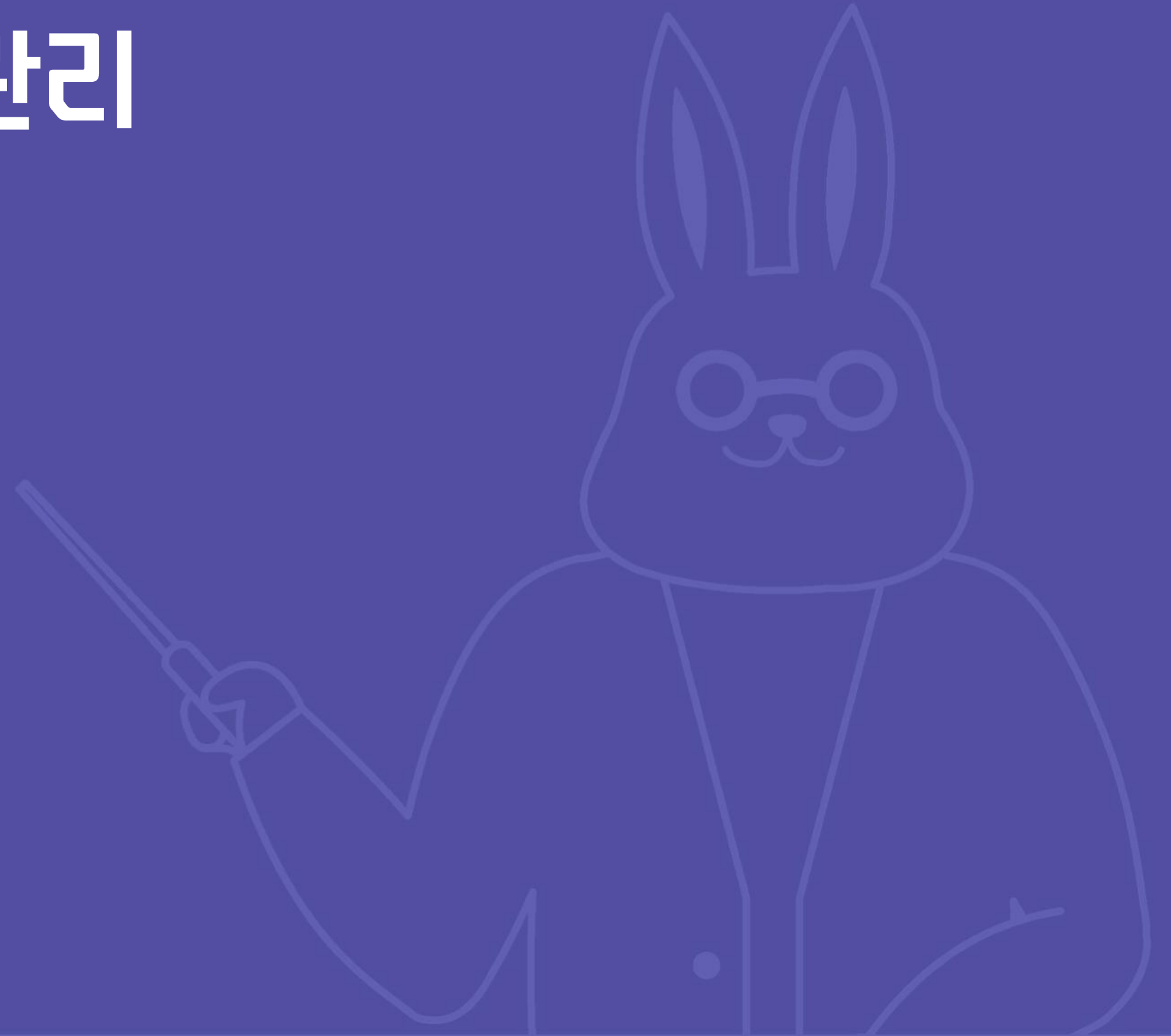
✓ 행맨 게임 소개



- 모든 알파벳을 맞추면 승리한다.

02

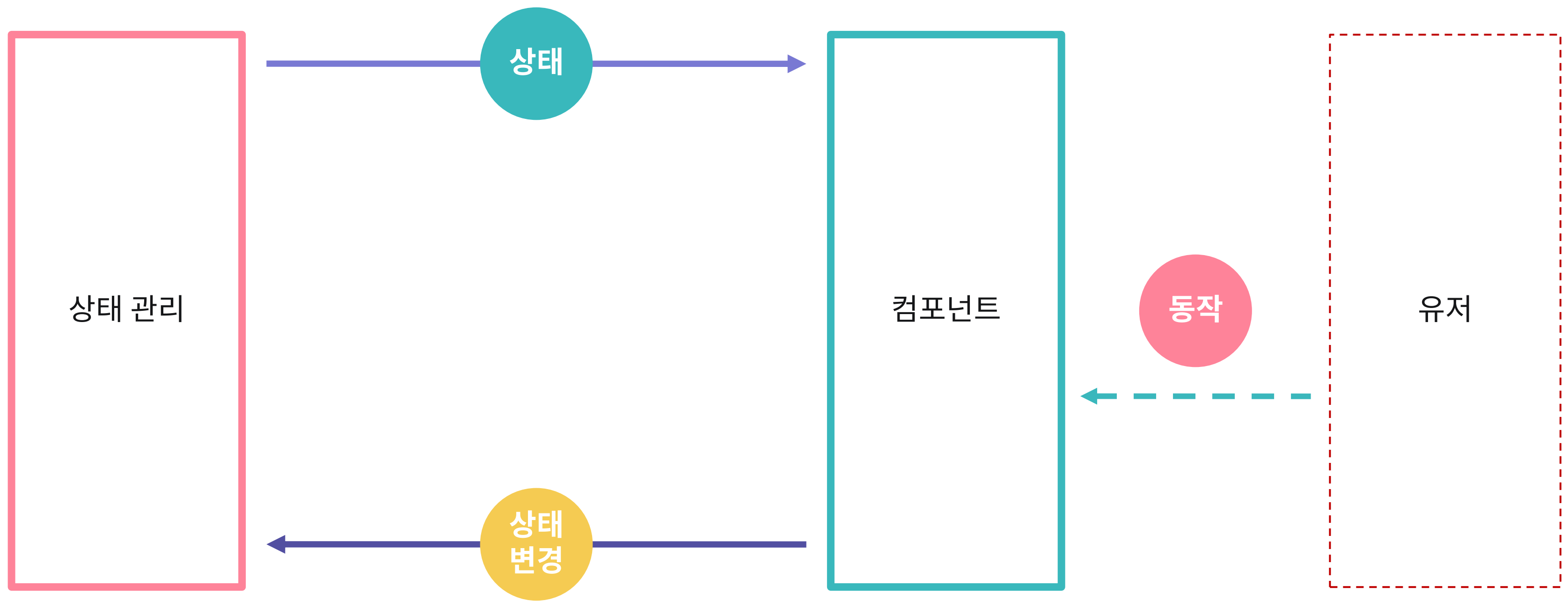
# 행맨 게임 코드 구조 - 상태 관리



## ✓ 상태 관리란

- 행맨 게임에서는 남은 시간, 남은 기회, 선택한 알파벳, 게임의 진행 상태 등의 데이터를 저장하고, 그 정보에 따라 필요한 UI를 화면에 그려야 한다.
- 이때, 데이터를 앱의 상태라 한다.
- 앱의 상태는 하나의 State 객체로 관리한다.

✓ 상태 관리 개념



## ✓ 상태 관리 패턴

- 상태를 관리하는 객체는 변하지 않는다. 이를 불변 객체라고 한다.
- 상태가 변경되면 State는 전혀 다른 객체가 된다.
- 상태를 변경하는 코드는 컴포넌트 코드와 섞이지 않도록 한다.
- 컴포넌트는 상태를 어떻게 변경할지 알지 못한다.
- 상태를 변경하고자 하면, 특정 메서드를 호출하도록 한다.

03

# 행맨 게임 코드 구조 - 컴포넌트



## ✓ 컴포넌트란

- 컴포넌트는 특정 UI의 영역을 담당하는 코드이다.
- 행맨 게임에서는 메시지를 그리는 컴포넌트, 키보드를 그리는 컴포넌트 등이 있다.
- 컴포넌트를 잘게 쪼개어 화면을 구성한다.



## ✓ 컴포넌트 패턴

- 컴포넌트는 상태를 직접 변경해서는 안 된다.
- 컴포넌트는 상태를 받아 사용하며, 상태 변경을 요청한다.
- 상태 변경은 상태를 관리하는 코드에서 한다.
- ex) 컴포넌트가 키보드 입력을 요청하면, 상태 관리

✓ 행맨 게임의 컴포넌트

행맨 이미지



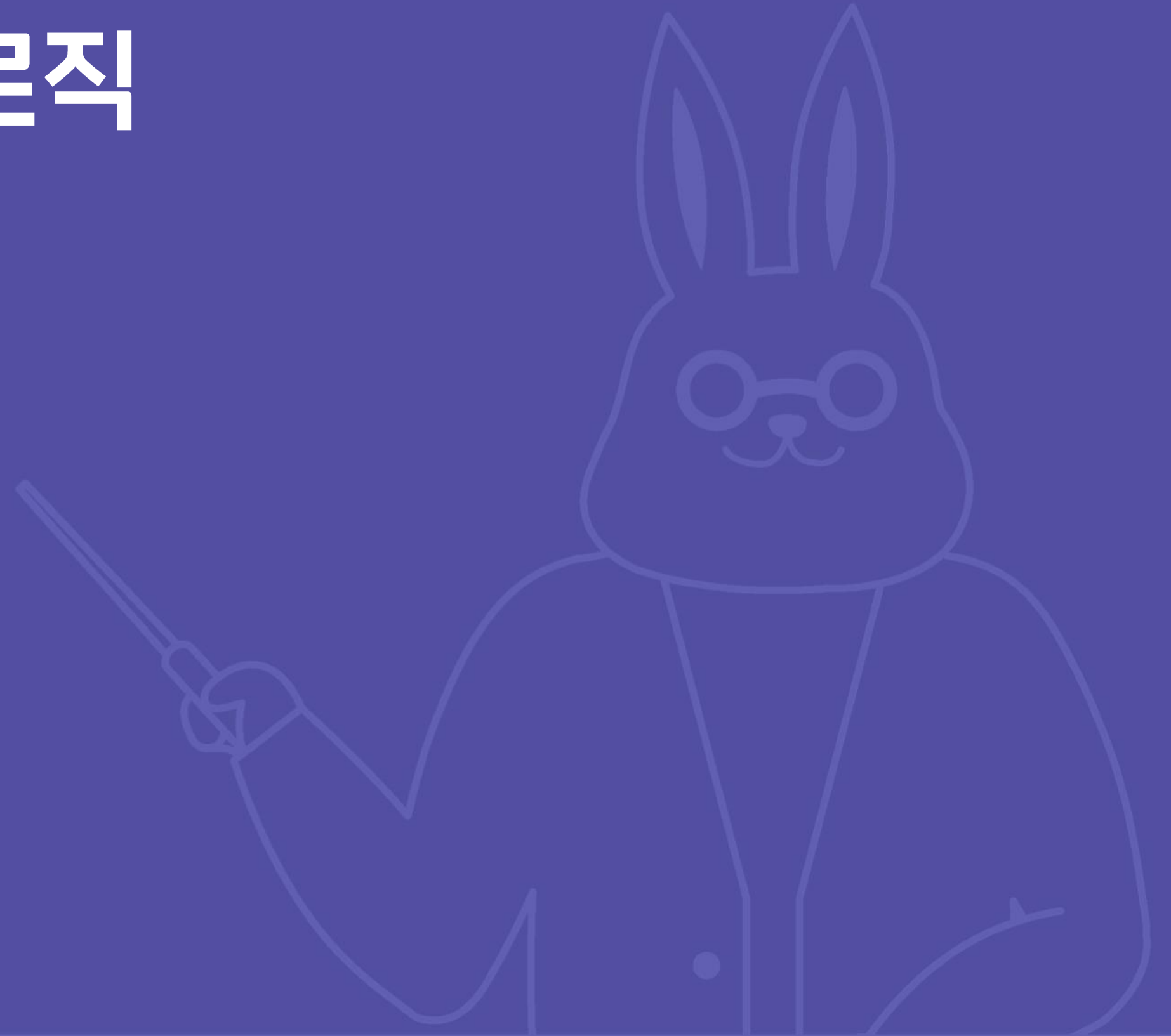
게임 조작

게임 메시지

키보드 레이아웃

04

# 행맨 게임 코드 구조 - 게임 로직



## ✓ 앱 초기화

- 처음 게임을 실행하면, App 코드가 실행된다.
- App은 상태를 초기화한다.
- App에 이미지를 미리 로드하여 저장한다.
- 이미지 로드가 끝나면 화면을 그린다.

## ✓ 게임의 상태 - 1

- state는 게임의 상태를 담는 객체이다.
- gameStatus는 게임의 현재 상태이다. LOSE, WIN, READY, START 등이 된다.
- timer는 게임이 끝날 때까지 남은 시간이다.
- wordLoading은 단어 로딩 유무를 boolean으로 체크한다.

## ✓ 게임의 상태 - 2

- chancesLeft는 알파벳 선택의 남은 기회이다.
- enteredCharacters는 선택한 알파벳을 객체로 저장한다.
- charMap은, 정답 단어를 객체로 저장한다.
- wordArr은 화면에 그릴 단어를 배열로 저장한다.
- charsLeft는 몇 개의 알파벳을 더 맞추어야 할지의 정보를 저장한다.

## ✓ 게임 시작

- START 버튼을 누르면 게임이 시작된다.
- 키보드를 누르면, 단어 안에 있는 알파벳인지를 검사한다.
- 단어 안에 있다면, charsLeft와 enteredCharacters 등을 변경하고 게임 종료를 검사한다.
- 단어 안에 있지 않다면, chancesLeft와 enteredCharacters 등을 변경하고 게임 종료를 검사한다.
- 시간이 지날 때마다 게임 종료를 검사한다.
- 모든 상태의 변경 때마다 컴포넌트를 다시 그린다.

## ✓ 게임 종료와 재시작

- 모든 알파벳을 맞추었거나, 시간이 종료되거나, 모든 기회가 끝나면 게임이 종료된다.
- 게임 종료 시 키보드를 더 이상 누를 수 없도록 한다.
- 재시작 시 상태를 다시 초기화하며, 그에 맞추어 UI를 그린다.



# 크레딧

/\* elice \*/

코스 매니저

이재성

콘텐츠 제작자

김일식

강사

김일식

감수자

김일식

디자이너

김루미

# 연락처

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

[contact@elice.io](mailto:contact@elice.io)

