제목 : 행맨 게임

소프트웨어학부, 1학년, 2024403505, 플레시카마리아

1. 과제 개요 :

이 프로젝트는 '행맨(헹맨)' 게임을 구현하는 것이다. 이 게임은 사용자가 주어진 단어에서 글자를 맞추는 게임으로, 틀린 글자를 선택할 때마다 '행맨' 그림이 점차적으로 그려지며, 최대 10번의 틀린 시도가 주어지다. 게임은 파이썬과 터틀 그래픽스를 사용하여 구현되며, 사용자가 게임을 입력할 수 있는 창을 띄워서 진행된다. 게임이 끝난 후, 사용자는 게임을 다시 시작할지 여부를 선택할 수 있다.

1. 문제 정의와 과제의 필요성

이 과제의 목적은 게임을 통해 프로그래밍의 기본적인 개념을 학습하고, 이를 이용해 재미있는 프로젝트를 완성하는 것이다. "행맨" 게임은 간단하면서도 사용자가 반복적으로 플레이할 수 있는 재미있는 요소를 제공하며, 텍스트 기반으로 작동하여 그래픽의 복잡성 없이 직관적으로 플레이할 수 있다.

1. 개발 개요

이 프로젝트는 파이썬을 사용하여 '행맨' 게임을 구현합니다. 주요 기술은 다음과 같습니다: 터틀 그래픽스, 랜덤 모듈, 입력 함수, Tkinter.

게임이 시작되면 무작위로 단어가 선택되다. 사용자는 알파벳을 입력하여 단어를 추측한다. 틀린 답을 할 때마다 '행맨' 그림이 단계별로 그려지다. 사용자가 단어를 맞추거나 최대 10번의 기회를 소진하면 게임이 종료되다. 게임이 끝난 후 다시 시작할지 여부를 묻는다.

1. 개발 내용(구체적으로 설명)

이 프로젝트는 파이썬과 Tkinter를 활용하여 직관적인 GUI 기반의 행맨 게임을 개발하였다.

프로젝트는 다음과 같은 주요 내용으로 구성된다:

첫째, 난이도 설정 기능을 통해 사용자는 쉬움, 중간, 어려움 중 원하는 난이도를 선택할 수 있다. 난이도는 단어의 길이와 허용된 시도 횟수에 영향을 미치며, 게임의 도전 수준을 조정할 수 있다.

둘째, REST API를 활용하여 무작위 단어를 가져오는 기능이 포함되어 있다. 이 API는 게임의 단어를 난이도에 따라 동적으로 생성하며, 사용자 경험을 풍부하게 만들어 준다.

셋째, Tkinter를 활용한 GUI는 사용자가 입력창을 통해 단어를 추측하고, 남은 시도 횟수와 잘못된 문자를 확인할 수 있는 직관적인 인터페이스를 제공한다.

넷째, 잘못된 시도 횟수에 따라 행맨 그림이 단계적으로 완성되는 기능이 포함되어 있다. 이 과정은 난이도별로 세분화되어 있어 쉬운 난이도에서는 더 많은 단계를, 어려운 난이도에서는 적은 단계를 거쳐 게임이 진행된다.

다섯째, Enter 키 입력 이벤트를 처리하여 사용자가 보다 직관적으로 문자를 입력하고 확인할 수 있도록 설계되었다. 이로 인해 게임의 접근성과 사용 편의성이 크게 향상되었다.

마지막으로, 사용자가 게임에서 이기거나 지는 결과가 발생할 경우 통계를 즉시 업데이트하고, 승리 및 패배 횟수를 실시간으로 확인할 수 있도록 구현하였다. 이를 통해 게임의 진행 상황과 성과를 시각적으로 제공하여 사용자 몰입도를 높였다.중간 계획과 최종 과제 사이의 차이점 구체적 기술(변경/수정 내용이 있다면 그 이유도 함께 기술)

1. 중간 계획과 최종 과제 사이의 차이점 구체적 기술(변경/수정 내용이 있다면 그 이유도 함께 기술)

최종 과제는 중간 계획과 비교하여 여러 가지 면에서 향상되었다.

먼저, 터틀 그래픽스를 사용한 단순한 그림 구현에서 벗어나 Tkinter 기반의 GUI를 통해 보다 직관적이고 시각적으로 풍부한 인터페이스를 제공한다. 초기 화면에서 사용자가 난이도를 선택할 수 있으며, 각 난이도는 단어의 길이와 허용된 시도 횟수를 달리하여 게임의 도전 수준을 조정한다.

또한, 단어 선택 방식도 개선되었다. 중간 계획에서는 사전에 정의된 단어 목록에서 무작위로 선택했지만, 최종 과제에서는 REST API를 활용하여 난이도에 적합한 단어를 동적으로 가져오는 기능이 추가되었다. 이를 통해 단어의 다양성이 크게 향상되었고, 반복 플레이 시 신선함을 유지할 수 있다.

게임 진행 방식에서도 변화가 있다. 잘못된 추측 횟수에 따라 행맨 그림이 단계별로 그려지는 점은 유지하되, 난이도별로 그림의 세분화 단계를 조정하여 더 높은 몰입감을 제공한다. 또한, 사용자가 잘못 입력한 문자를 화면에 표시하는 기능과 Enter 키를 통해 문자를 확인할 수 있는 이벤트 처리 기능이 추가되어 사용자 경험이 개선되었다.

게임이 종료된 후에는 승리 및 패배 통계가 업데이트되며, 통계를 통해 사용자가 자신의 성과를 확인할 수 있도록 했다. 새로운 게임을 시작할 경우, 난이도 선택 화면으로 돌아가거나 동일한 난이도에서 이어서 플레이할 수 있는 선택지를 제공함으로써 유연성을 높였다.

최종적으로, 이러한 개선 사항들은 중간 계획의 기본 구조를 유지하면서도 게임의 전반적인 품질과 재미 요소를 크게 향상시켰다.

1. 결과

이 프로젝트의 결과물은 사용자 친화적인 GUI 기반의 행맨 게임으로, 다양한 난이도와 직관적인 인터페이스를 제공한다. 사용자는 쉬움, 중간, 어려움 중 원하는 난이도를 선택할 수 있으며, 난이도에 따라 단어의 길이와 시도 횟수가 조정된다. REST API를 활용해 단어를 랜덤으로 가져오는 기능은 게임의 다양성과 신선함을 더해준다.

1. 결론

이 프로젝트는 행맨 게임을 단순한 텍스트 게임에서 직관적이고 시각적으로 풍부한 GUI 기반 게임으로 발전시켰다. 이 과정에서 프로그래밍 기초와 GUI 설계 능력을 동시에 배양하였으며, Python 언어의 다양한 모듈 활용 경험을 쌓았다. 본 프로젝트는 프로그래밍 학습 초기 단계에 이상적인 과제로 평가된다.

게임을 향상시킬 수 있는 주제와 방법을 연구하는 데 많은 시간이 걸렸다. 나는 결과를 보고, 테스트하고, 어떻게 개선할 수 있는지 생각해보는 것이 좋았습니다. 점차적으로 더 많은 개선 사항을 추가하고 이상을 전달합니다. 나는 그것이 마음에 들었다. 앞으로 게임의 디자인을 다듬고, 더 화려하게 만들고, 배경음악을 추가할 수 있을 것이다.

1. 참고자료

* 파이썬 공식 문서 tkinter: <https://docs.python.org/3/library/tkinter.html>
* Python GUI Programming With Tkinter: <https://realpython.com/python-gui-tkinter/>
* 파이썬 공식 문서 random: <https://stackoverflow.com/questions/18834636/random-word-generator-python>
* 파이썬 공식 문서 requests: <https://requests.readthedocs.io/en/latest/>
* REST API 비디오 수업: Python REST API Tutorial for Beginners | How to Build a Flask REST API - <https://youtu.be/z3YMz-Gocmw?si=SUjao1RoaSEYotQ_>
* Random Word API 문서: <https://random-word-api.herokuapp.com/home>