

Kelime Oyunu Uygulaması

1)Sude DEMİRTAŞ

210501012

sudemiirtas@gmail.com

2) İclal CENGEL

21050013

iclalcengel63@gmail.com

Abstract—Bu projenin amacı Teve2 kanalında, A. İhsan Varol tarafından sunulan Kelime Oyunu isimli program için bir uygulama geliştirilmesidir.

Index Terms—dosya işlemleri

I. PROJE DETAYI

- Bu uygulama en az iki bölümden oluşmalıdır. Soru/Kelime ekleme, Oynama, vb... Sorular ve cevapları yerel dosya(txt,xml vs) yada veritabanında tutulmalıdır Programda, kullanıcıdan 14 adet (4 harften başlayıp 10 harfe kadar olacak şekilde her birinden 2 tane) kelimeyi toplamda 4 dakikalık bir süre zarfında bulması istenmelidir. Doğru bilinen her kelime, kullanıcının açtığı her harf başına, kullanıcı 100 puan kazanmalıdır.
- Programda kelimeler, ilgili dosya/dosyalardan rastgele bir şekilde seçilmeli ve sorulmalıdır.
- Oyuncunun harf isteme hakkı olmalıdır. İstendiğinde, kelimenin rastgele bir harfi belirlmelidir.
- Tahmin etmek istendiğinde, 4 dakikalık süre durdurulmalıdır. 20 saniye içinde doğru cevabın yazılması gerekmektedir.
- Bu sürede doğru cevap yazılamadığı taktirde kullanıcı puanından açılmayan harf sayısı ile orantılı bir şekilde puan düşürülmelidir.
- Oyun bittiğinde oyuncunun adı, puanı, kalan süresi, oyun oynama tarih ve zamanı ekranda gösterilmelidir.
- Soru/Kelime ekleme arayüzünün kullanılabilirliği ve düzeni (10 puan)
- Oyun mantığının doğru bir şekilde kurulması (30 puan)
- Oyun arayüzünün kullanılabilirliği ve düzeni (10 puan)
- Ekstra özellikler (15 puan)
- Yazılan kodlara olan hakimiyet (35 puan)

II. PROGRAM DILI

A. C++

C++ programlama dili, yüksek seviyeli bir genel amaçlı programlama dilidir. C++'ın temeli, C programlama diline dayanırken, ona nesne yönelimli programlama (OOP) özelliklerini ekler. Bu nedenle C++ hem prosedürel programlama hem de nesne yönelimli programlama paradigmasını destekler.

Identify applicable funding agency here. If none, delete this.

- Nesne Yönelimli Programlama (OOP): C++'ın en önemli özelliklerinden biri, nesne yönelimli programlamayı desteklemesidir. Bu, programın yapılandırılmasını, veri ve işlevlerin bir araya getirilmesini ve tekrar kullanılabilirlik ve modülerlik sağlamak için sınıflar ve nesnelerin kullanılmasını içerir.
- Genişletilebilirlik: C++ dilinin bir başka önemli özelliği, diğer programlama dilleriyle uyumlu olması ve kullanıcıların kendi veri türlerini ve fonksiyonlarını tanımlayabilmesidir. Bu, C++'ın genişletilebilir bir dil olmasını sağlar ve farklı programlama ihtiyaçlarına uygun çözümler sunar.
- Verimlilik: C++ dili, düşük seviye programlama yetenekleri sunar ve bellek yönetimi üzerinde doğrudan kontrol sağlar. Bu, C++'ı verimli ve hızlı çalışan uygulamalar geliştirmek için tercih edilen bir dil haline getirir.
- Şablonlar: C++ dilinde şablonlar, genel programlama teknikleri için kullanılır. Şablonlar, türden bağımsız bir şekilde çalışan fonksiyonlar ve sınıflar oluşturmanızı sağlar. Bu, kod tekrarını azaltır ve farklı veri tipleriyle çalışabilen esnek yapılar oluşturmanızı sağlar.
- Taşınabilirlik: C++ dili, platformdan bağımsız bir şekilde çalışabilir. Bu, C++'ın farklı işletim sistemleri ve donanım mimarileri üzerinde çalışabilme yeteneği sağlar. Bu da C++'ı taşınabilir ve çok platformlu uygulamaların geliştirilmesinde tercih edilen bir dil yapar.

III. PROJEYE BAŞLAMADAN ÖNCE NELER YAPTIK

İlk olarak projenin ana hedeflerini ve kapsamını netleştirmek için bir proje tanımı ve proje planı oluşturduk. Bu adımda, projenin neyi başarmayı hedeflediğini, hangi sonuçları elde etmek istediğimizi ve proje sürecinin nasıl ilerleyeceğini belirledik. Proje için gerekli kaynakları belirledik ve planladık. Projede yer alan tüm paydaşlar arasında etkili iletişimi sağlamak için bir iletişim planı oluşturduk. Bu plan, iletişim kanallarını, toplantılarını ve raporlama süreçlerini belirlemeyi

içerir. Ayrıca paydaşların bilgilendirilmesi ve güncel tutulması için iletişim stratejileri geliştirdik. Proje sürecini yönetmek ve ilerlemeyi takip etmek için bir zaman çizelgesi oluşturduk. Bu zaman çizelgesi, projenin başlangıç ve bitiş tarihlerini, aşamalarını ve kilometre taşlarını içerir. Böylece proje ilerlemesini takip edebilir ve gerekirse planı güncelleyebiliriz. Projeye başlamadan önce gereken kaynakları temin ettik veya hazırladık. Bu kaynaklar, proje için gerekli olan malzemeleri, ekipmanları veya yazılımı içerebilir. Bu adım, proje başladığında kaynak eksikliği nedeniyle gecikmeleri önlemek için önemlidir. Bu konuya dair videolar bulup izledik. Yarışmanın nasıl ilerlediğini öğrenmek için program bölümlerini izledik.

IV. BU PROJE SAYESİNDE YENİ ÖĞRENDİĞİMİZ ŞEYLER

- * Projemizin konusuyla ilgili olarak derinlemesine bir bilgi birikimi edindik. Araştırma yaparken, konuyla ilgili güncel trendleri, en iyi uygulamaları ve sektördeki gelişmeleri yakından takip ettik. Bu sayede, konu hakkında daha geniş bir perspektif kazandık ve uzmanlığımızı artırdık.
- * Bu projeyi yönetme sürecinde, proje yönetimi becerilerimizi geliştirdik. Zaman yönetimi, kaynak planlaması, risk yönetimi ve iletişim becerileri gibi önemli becerileri pratiğe döktük. Proje takvimini oluşturma, takımı yönetme ve etkili iletişim kurma konularında daha yetkin hale geldik.
- * Projede kullanılan belirli teknikler veya araçlar üzerinde çalışarak yeni teknik yetkinlikler kazandık. Örneğin, proje yönetimi yazılımları, veri analizi araçları veya tasarım programları gibi özel araçlarla çalışarak bu alanlardaki becerilerimizi artırdık. Bu projenin gerektirdiği teknikleri öğrenerek, kendimizi teknolojik açıdan da geliştirdik.
- * Projede bir ekip olarak çalışarak takım çalışması becerilerimizi geliştirdik. Ekip üyeleriyle etkili iletişim kurma, görevleri koordine etme, işbirliği yapma ve çözüm odaklı çalışma gibi yetenekleri pekiştirdik. Birlikte çalışmanın önemini ve başarılı bir şekilde işbirliği yapmanın yollarını öğrendik.
- * Projenin farklı aşamalarında karşılaştığımız sorunlarla uğraşarak problem çözme yeteneklerimizi geliştirdik. Yaratıcı çözümler bulma, analitik düşünme, karar verme ve esneklik gibi becerilerimizi kullanarak zorlukları aştık. Bu deneyimler, gelecekteki projelerde karşılaşılabileceğimiz sorunlara daha etkili bir şekilde yaklaşmamızı sağladı.
- * Projeyi tamamlarken, elde ettiğimiz sonuçları etkili bir şekilde sunma ve raporlama becerilerimizi geliştirdik.

V. SONUÇ

- * Bu proje, C++ programlama dilini daha iyi anlama ve kullanma amacıyla gerçekleştirildi. Projemizin sonucunda aşağıdaki başarıları elde ettik:
- * C++ Dilinde Yetkinlik: Proje boyunca C++ dilini daha iyi anlama ve kullanma becerilerimizi geliştirdik. Dilin temel yapılarını, sözdizimini ve en iyi uygulamalarını daha iyi anladık. Bu, C++ dilinde daha etkili ve verimli kod yazmamıza olanak sağladı.
- * Nesne Yönelimli Programlama (OOP): Projede C++'ın nesne yönelimli programlama (OOP) özelliklerini uygulayarak, daha yapılandırılmış ve modüler kodlar oluşturduk. Sınıflar, nesneler, kalıtım, çoklu kalıtım, polimorfizm gibi OOP kavramlarını kullanarak programları daha esnek ve sürdürülebilir hale getirdik.
- * Verimli ve Performanslı Kodlama: C++ dilinin doğrudan bellek yönetimi ve düşük seviye kontrol yeteneklerini kullanarak verimli ve performans odaklı kodlama yapma becerilerimizi geliştirdik. Bellek kullanımını optimize etme, veri yapılarını etkin kullanma ve algoritmalarda hızlı çalışma prensiplerini uygulama konusunda ilerleme kaydettik.
- * Şablonlar ve Genel Programlama: Projede C++'ın şablonlar özelliğini kullanarak genel programlama tekniklerini uyguladık. Şablonlar sayesinde türden bağımsız kod yazma yeteneğini kazandık ve daha esnek ve yeniden kullanılabilir fonksiyonlar ve sınıflar oluşturabildik.
- * Hata Ayıklama ve Sorun Giderme: Projeyi gerçekleştirirken, hata ayıklama ve sorun giderme becerilerimizi geliştirdik. Hataları bulma, hata ayıklama araçlarını kullanma ve kodu optimize etme konularında deneyim kazandık. Bu da daha sağlam ve hatasız kodlar oluşturmamıza yardımcı oldu.
- * Proje sonucunda, C++ programlama dilinde yetkinlik kazanarak daha etkili ve verimli kodlar yazma yeteneğimizi geliştirdik. Aynı zamanda nesne yönelimli programlama, verimli kodlama, genel programlama ve hata ayıklama gibi önemli becerilerimizi pekiştirdik. Bu deneyim, gelecekteki C++ projelerinde daha başarılı olmamıza ve daha karmaşık uygulamalar geliştirmemize yardımcı olacaktır.

VI. KAYNAKÇA

- * C++ Programlama Dili:
"C++ How to Program" (Kitap): Paul Deitel, Harvey Deitel
"The C++ Programming Language" (Kitap): Bjarne Stroustrup
C++ referans dokümantasyonu: <https://en.cppreference.com/>
- * Zamanlama ve Süre Takibi:

C++ time.h kütüphanesi: C++11'de tanıtılan bir zamanlama kütüphanesidir. İşaretçiler ve süre hesaplamaları için kullanılabilir. Örnek kodlar: <https://en.cppreference.com/w/cpp/chrono>
<https://www.geeksforgeeks.org/queue-cpp-stl/>

* Dosya İşlemleri:

C++ fstream kütüphanesi: Dosya okuma ve yazma işlemleri için kullanılır. Örnek kodlar: <https://www.cplusplus.com/doc/tutorial/files>
<https://www.geeksforgeeks.org/file-handling-c-classes/>

* Oyun Mantığı:

Oyun mantığı ve puan hesaplama için projenizin gereksinimlerine ve tercihlerinize bağlı olarak özel algoritmalar veya işlevler oluşturmanız gerekebilir.