



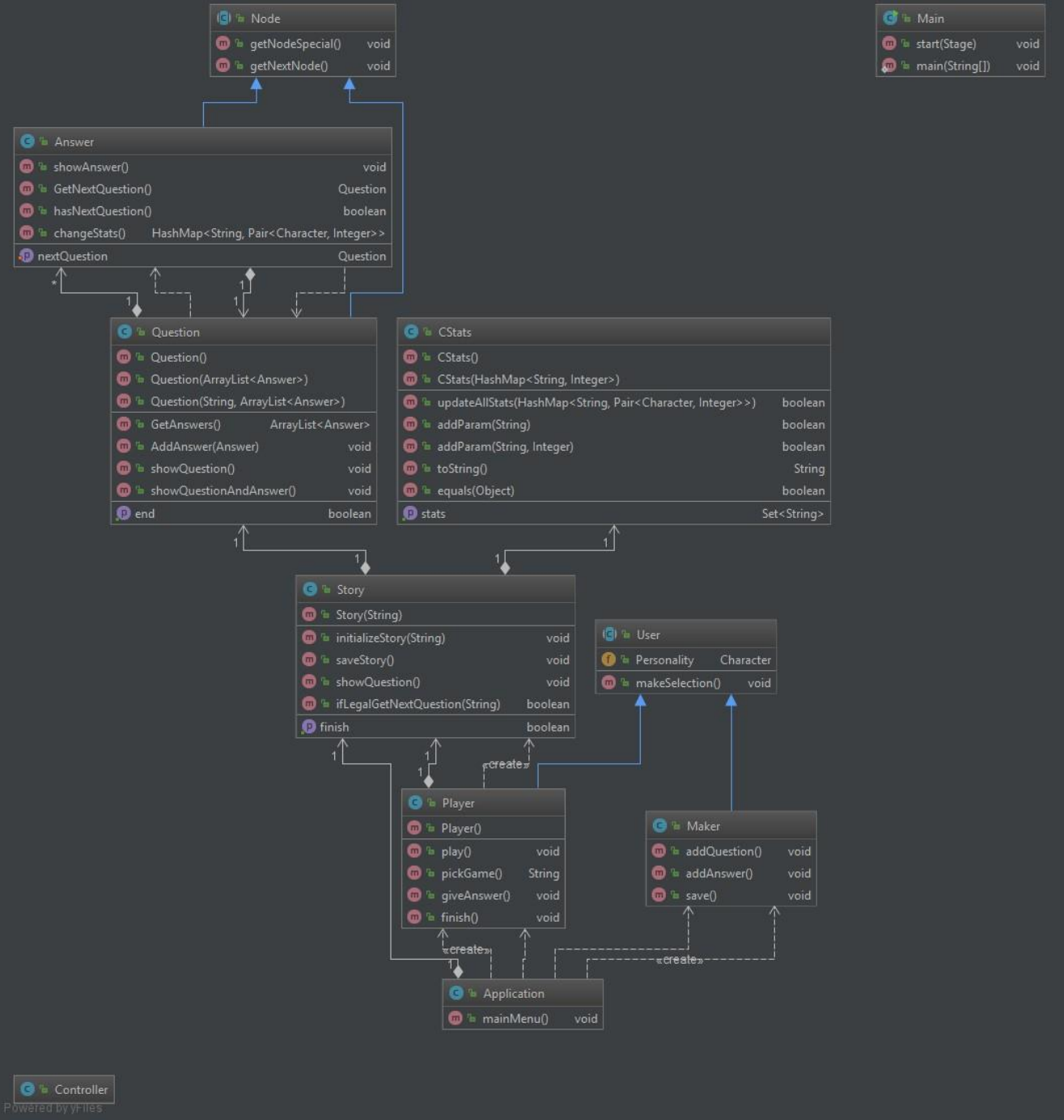
ÇOK GÜZEL RAPOR3

0X002A (METİN TABANLI OYUN YAPMA ARACI)

Bu dosya (BİL222) Veri Yapıları Ve Algoritmaları Dersi 2017-2018 (Bahar Dönemi) kapsamında yapılan proje çalışmasının ikinci raporudur. Rapor "Çok Güzel" grubunun 0x002a adlı projesine aittir.

ÇOK GÜZEL RAPOR3

0X002A (METİN TABANLI OYUN YAPMA ARACI)



SINIF DİYAGRAMI

İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.

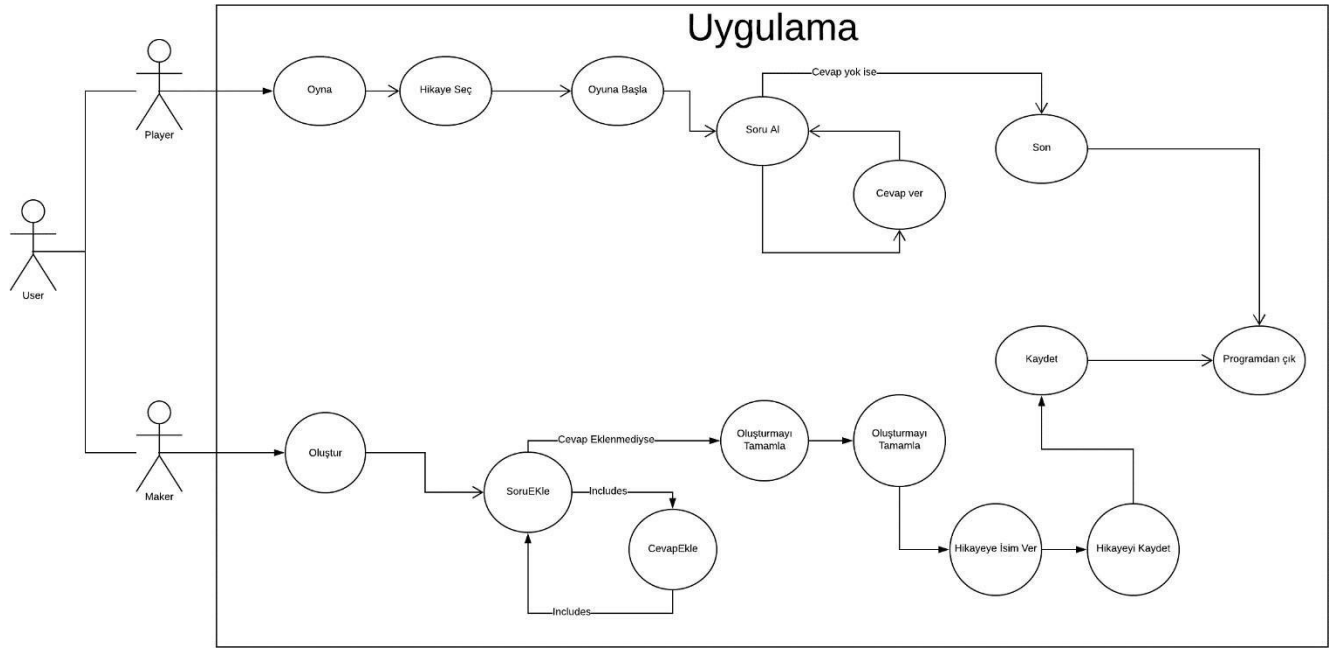
42 Sequence Diagram

```
sequenceDiagram
    actor User
    participant GUI
    participant Uygulama
    participant Soru
    participant Cevap

    Note over User, GUI: YAPIM
    User->>GUI: Menuden "yap" seçimi
    activate GUI
    GUI->>Uygulama: Yapım Modu
    deactivate GUI
    Uygulama->>Soru: Soru kayıt
    activate Soru
    Soru->>Cevap: Soru-Cevap Bağlantı
    deactivate Soru
    Uygulama->>Cevap: Cevap Kayıt
    activate Cevap
    Cevap->>Uygulama: 
    deactivate Cevap
    Uygulama->>Uygulama: Oyun dosyası oluşturma
    deactivate Uygulama
    Uygulama->>GUI: Karakter İklendirme
    activate GUI
    GUI->>Uygulama: Soru-cevap kaydet
    activate Uygulama
    Uygulama->>Uygulama: 
    deactivate Uygulama
    Uygulama->>GUI: Önizleme
    deactivate Uygulama
    deactivate GUI

    Note over User, GUI: OYUN
    User->>GUI: Menuden "oyna" seçimi
    activate GUI
    GUI->>Uygulama: Oyun Modu
    deactivate GUI
    Uygulama->>Uygulama: Oyun Verilerinin Yüklmesi
    activate Uygulama
    Uygulama->>Soru: Soruların Yüklmesi
    activate Soru
    Soru->>Cevap: Cevapların Yüklmesi
    activate Cevap
    Cevap->>Uygulama: 
    deactivate Cevap
    Uygulama->>Uygulama: 
    deactivate Uygulama
    Uygulama->>GUI: Oyun dosyası seçimi
    activate GUI
    GUI->>Uygulama: 
    deactivate GUI
    Uygulama->>Uygulama: 
    activate Uygulama
    Uygulama->>Soru: Cevap - Soru Alışverişi
    activate Soru
    Soru->>Cevap: 
    deactivate Soru
    Uygulama->>Cevap: 
    activate Cevap
    Cevap->>Uygulama: 
    deactivate Cevap
    Uygulama->>Uygulama: 
    deactivate Uygulama
    Uygulama->>GUI: Soru-Cevap Alışverişi
    activate GUI
    GUI->>Uygulama: Cevap verme
    activate Uygulama
    Uygulama->>Uygulama: 
    deactivate Uygulama
    Uygulama->>GUI: Sonuç Basma
    deactivate Uygulama
    Uygulama->>Uygulama: Karakter Güncelleme
    activate Uygulama
    Uygulama->>Uygulama: Sonucu Hazırlama
    deactivate Uygulama
    Uygulama->>GUI: Oyun Bitti
    deactivate Uygulama
    deactivate GUI
```

İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.



KULLANICI - SENARYO DİYAGRAMI (USE-CASE DİAGRAM)

İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.

42 programı 2 bölümden oluşur, Oynama ve Oluşturma. Kullanıcı Oynama veya oluşturma modlarından birini seçer.

Oynama moduna tıklayınca ekranda oynayan kişinin seçmesi için mevcut senaryolar listelenir. Kullanıcı bunlardan bir tanesini seçtikten sonra karşısına ilk soru gelir ve oyun başlar. İlk sorudan itibaren her soru, verilen cevaplara göre farklı bir soruya ilerler. Oynanılan senaryo, oluşturan kişi tarafından belirli bir hedefe yönelik oluşturulmuş olabilir. Bu hedeflere göre verilen her cevap kullanıcının özelliklerini günceller ve gelen sorular bu özellikler doğrultusunda değişebilir. Kullanıcı gelen soruları cevaplayarak senaryoda ilerler ve senaryo oluşturanın belirlediği sonucu alır ve oyun sona erer.

Oluşturma modu seçildiğinde senaryoyu oluşturan kişinin karşısına soru ekleme, sorulara ön şartlar ekleme, sorulara cevap ekleme ve cevaplara yan etki ekleme gibi seçenekler gelir. Senaryoyu oluşturan kişi istediği soruları ekledikten sonra test sonuçlarını ekledikten sonra senaryosunu tamamlar, senaryosuna isim verir ve senaryosunu kaydederek uygulamadan çıkar

SINIFLAR

Application

Application sınıfı senaryo oluşturmak ve oynamak için gerekli verileri içinde tutan bir çerçeve görevi görür. Diğer sınıflar arasındaki iletişimi sağlar.

Metotları:

MainMenu(): Uygulamanın başlaması için gerekli metodları çağırır.

Sınıf Değişkenleri:

Story story: Hikayeyi tutar.

AÇIKLAMA

Bu bölüm de Sınıf Diyagramı'nda bulunan sınıfların ve sınıfların içerdği metot ve değişkenlerin açıklaması yapılmıştır.

User

User soyut sınıfı, Application içerisinde kullanılmaktadır. Kullanıcının Maker ya da Player olduğunu anlamamızı sağlar.

Maker ve Player kendisinden türemiş olup Application içinde "aşağı atama" (**downcasting**) metodu ile kullanılacaktır.

Ayrıca User sınıfı değişken (**variable**) olarak **Character** sınıfını da barındırmaktadır.

Metotları:

MakeSelection(): Kullanıcının soruya cevap vermesi veya soru eklemesi sırasında App ile etkileşime girerek orada yapacağı işlemleri kendi içerisinde görmesini sağlar. Yapılan işlem sonucunda oluşan yeni Node (Maker için yeni oluşturulmuş soru, Player için cevap verildikten sonra gidilecek yeni soru) geri döndürülür.

Sınıf değişkenleri:

Application tipinde App: Kendisini oluşturan ana uygulamanın ve bu ana uygulama sayesinde sorulara ve cevaplara erişimin sağlanması için oluşturulmuş bir değişken.

Karakter tipinde Table: Kullanıcının ana uygulamada yaptığı adımlar sonrası değiştireceği özellikler listesi. Maker için bu liste üzerinde çalışılacak bir taslak alan görevi görmek üzere oluşturulurken, Player için bu liste karakterin kendi özelliklerini gösterir.

Player

Player, oyunu oynama modunda çalıştırmasını sağlayan bir User'dır. Oyun açıldığında Oyna modu seçilirse Application sınıfında Player'ın objesi oluşturulur ve Player sınıfında 3 metot bulunur:

Metotları:

giveAnswer(Question): Oyuncunun kendisine girdi olarak verilen soruya cevap vermesini sağlar, verilen cevabı döndürür.

pickGame(): Oyuncunun girdiği kayıt dosyası(daha önce maker da yapılmış ya da kayıt formatına uygun) Application'a iletir.

getEffect(LinkedList<Pair<String,Pair<Char,Integer>>>): Kendisine girdi olarak verilmiş olan bağlı liste doğrultusunda kendi karakterini günceller.

finish(): Oyunun bitmesi durumunda oyunun sonunda oluşan Karakteristik özellikleri döndürür

Sınıf Değişkenleri:

Node tipinde Kuyruk, Geçmiş: Sınıf içerisinde şu ana kadar ilerlenen bütün Node'ları içerisinde barındırır.

Maker

Maker, oyunu senaryo oluşturma modunda çalıştırmasını sağlayan bir User'dır. Oyun açıldığında Oluştur modu seçilirse Application sınıfında Maker objesi oluşturulur ve Maker sınıfında 3 method bulunur:

Metotları:

addQuestion(Answer): Verilen cevaba soru ekler ve eklenen soruyu geri döndürür. Eğer parametre olan answer yerine null verilirse, oyunun başlangıç sorusu olarak eklenir.

addAnswer(Question): Metotlara cevap ekler ve eklenen cevabı geri döndürür.

save(String): Oluşturulan senaryoyu girdi olarak verilen Karakter dizisi'nin gösterdiği isimde bir dosya oluşturup oraya kaydeder.

createStory(): Tüm yaratma operasyonlarının gerçekleşeceği genel metot.

Sınıf Değişkenleri:

HashMap<String, Integer> tipinde gameChar: Karakter özelliklerinin ismini ve başlangıç değerlerini tutar. Story newStory yaratılan oyunu kendi içerisinde tutarak kaydetme işlemini kolaylaştırır.

Story tipinde newStory: Yaratılan oyunu kendi içerisinde tutarak, kaydetme ve oluşturma işlemlerinde kullanılır.

Node

Node, oyundaki adımların her birini oluşturan ve içinde Quest ve Answer objelerini bulunduran bir sınıftır.

Her Node (Oyunun son Node'u hariç) kendinden sonraki bir veya birden çok Node'a bağlıdır. Bu bağlantı Node'un içinde ki Answer objelerinin bir sonraki Node'un adresini tutarak sağlamaktadır. Ve bu Nodelar LinkedList yapısında tutulmaktadır.

Metotları:

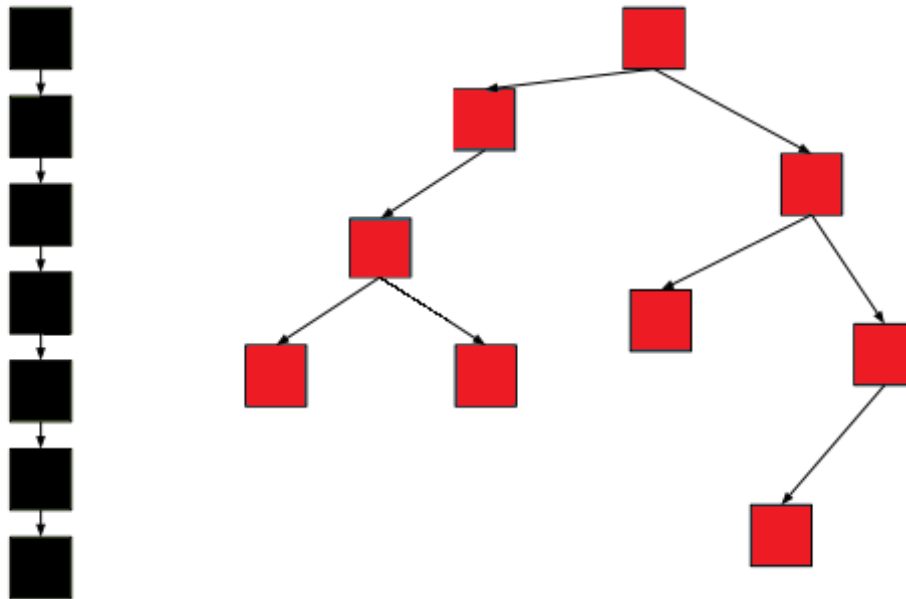
getNodeSpecial(): O Node' a ait özel karakteristik durum bilgisini Bağlantılı Liste şeklinde döndürür.

getNextNode(): O Node' a ait olan bir sonraki Node'lar listesi içerisinde, kullanıcının tercihine bağlı olarak, seçilen bir sonraki Node geri döndürülür.

Sınıf Değişkenleri:

Application tipinde App: Ana uygulama ile karşılıklı etkileşimin sağlanması için kullanılacak değişken. Karakter dizisi(String) Kullanıcıya bildirilecek olan metnin tutulacağı değişken.

Temsili Oyun Node Düzeni/ LinkedList Yapısı



Answer

Answer, senaryodaki sorulara verilebilecek cevapların bilgilerini içerir. Bunlardan ilki cevabın oluşturacağı sonuçlar (Ağaçta kalan bir kediyi kurtarınca insanlık değerinizin artması gibi) Çift(Pair) tipinde bağlantılı liste (LinkedList) içinde, değiştirilecek özelliğin ismi, yapılacak işlemin sembolü ve etkisi şeklinde saklanır, diğeri ise verilen cevabın götürebileceği yeni sorular, bu sorular ise LinkedList içinde saklanır.

Metotları:

GetSideEffect(): Oluşturan kişinin belirlediği yan etkileri Maker'dan alır.

GetNextQuestion(): Durumu uygun olan sorulardan birini soru listesine ekler.

GetAvailableQuestions(): Seçilen cevaba göre karşısına çıkabilecek soruları alır.

AddSideEffect(): Yan etkileri cevaba ekler.

Sınıf Değişkenleri:

Karakter Özelliği: LinkedList<Pair<String, Pair<Char, Integer>>> Answer Node'unun özel karakteristik durum bilgisini içerir. Bu Node'un seçilmesi halinde;

- **String alanı:** Kullanıcının hangi bilgisinin değişeceği,
- **Char:** (+, -, *, /, %) olmak üzere belirtilen bilginin ne yönde değişeceği bilgisi
- **Integer:** Değişimin ne kadar olacağı bilgisi.

İçeren değişkenler listesini barındırır.

Soru: bir sonraki öncelikteki sorusunun numarasını tutar

Question

Question, senaryoda kullanıcıya sorulan soruların bilgilerini içeren sınıftır. Bu bilgiler; sorunun kendine özel kimliğinin (ID'sinin) Integer tipinde tutulduğu bir id değişkeni (ilk sorunun idsi 1 olacak şekilde), sorunun cevaplarının (örn. Yoluma devam ediyorum) Set şeklinde tutulduğu bir Answers değişkeni ve sorunun metninin (örn. Yolda yaralanmış bir kedi buldun, ne yapacaksın?) String tipinde tutulduğu bir QuestionText değişkenidir. Cevaplar için Set kullanılmasının nedeni, cevaplar arasında herhangi bir bağlantı ya da öncelik bulunmaması ve her sorunun eşsiz olmasıdır. Set veri tipi, cevapların tutulması için uygun görülmüştür. Bunlarla birlikte :

Metotlar:

IsEnd(): Sorunun cevabı yoksa, senaryoda sona gelinmiş demektir. Bu metot, belirli sorunun son olup olmadığını kontrol eden metottur.

GetAnswers(): Sorunun cevaplarını Set şeklinde döndüren metottur.

AddAnswer(): Soruya bir cevap eklemek için kullanılan metottur.

ShowQuestion(): Sorunun metnini terminale bastıran metottur.

ShowQuestionAndAnswer(): Soruyu cevapları ile birlikte terminale bastıran metottur.

Sınıf Değişkenleri:

id: Soruya ait kimliği (her soru için eşsiz bir kimlik oluşturulur) Integer tipinde tutan değişkendir.

Answers: Sorunun cevaplarını Set ile tutan değişkendir.

QuestionText: Sorunun metnini String tipinde tutan değişkendir.

Cstats

Uygulama içerisinde yaratılacak olan hikaye doğrultusunda, yaratıcı kişinin araştırmak istediği değerlerin saklanacağı sınıftır. Bu sınıf değerlerini HashMap içerisinde String-Integer ikilileri ile tutar. Tutulan değerler, gelen diğer ikililer ile güncellenir ve yeni değerlere geçiş sağlanır. Değerlerin ilerleyişi sonunda özelliklerin değerleri aşamalara göre güncellenmiş olur.

HashMap tercih etme sebebi: String değişkenleri, ilk atanışlarından sonra sabit birer anahtar olacaklardır. Anahtarlar, karşılıklarında sadece birer Integer değer tutacakları için ve bu yapıyı sağlayan sınıfın HashMap olmasından dolayı, bu kullanıma karar verilmiştir.

Metotları:

Umumi updateAllStats metodu: parametre olarak HashMap<String, Pair<Character, Integer>> alır ve aldığı parametre içerisindeki özelliklerin (String) kendi içerisindeki uygun özellikler ile eşleştirerek, parametredeki özelliğin karşılığı olan değerdeki işlemin yapılması için yardımcı metoda gönderilir.

Gizli updateStats metodu: parametre olarak Pair<String, Pair<Character, Integer>> alır. Pair<Character, Integer> ikilisinin Char içeren (soldaki) değerini inceler. İncelenen değer karşılığında tanımlı olan işlem, Sınıfın içerisindeki değer ile eşleşen parametredeki (Pair<String,...) String değerine uygulanacaktır. Bu işlemin yapılması için gerekli olan sayı değeri de parametre olarak alınan Pair<Character, Integer> değerindeki ikilinin Integer (sağdaki) değeri ile ele alınacaktır.

Umumi addParam metodu: parametre olarak String alır. Kullanıcı tarafından sınıfa yeni bir özellik değeri eklenmesi için kullanılacak metoddur, yardımcı metodun çağırılması ile işlemini tamamlar, Olumlu ya da olumsuz bir durum bilgisi döndürür.

Umumi addParam metodu: parametre olarak String ve Integer alır. Özellik olarak ekleyeceği String değerine bir ilk değer atamak isteyen kullanıcı tarafından kullanılır. Olumlu ya da olumsuz bir durum bilgisi döndürür.

Umumi getStats metodu: parametre almaz. Özelliklerin isimlerini bir Küme (Set) şeklinde geri döndürür.

Sınıf Değişkenleri:

HashMap<String, Integer> tipinde status değişkeni: İçerisinde, üzerinde araştırma yapılacağı özelliklerin isimlerini String, her bir özelliğin kendine ait özel değerini Integer olarak tutar.

Static final int tipinde MAX_VALUE sabiti = 4: Kullanım olarak verilebilecek en fazla değişken sayısını bildirmek için vardır.

Static final int tipinde DEFAULT sabiti = 0: Eklenen özellik bilgilerinin temel olarak atanacağı değeri bildirmek için kullanılacak sabit.

Story

İçinde bütün veriyi tutup manipüle etmeye yarayan sınıf. Maker sınıfı tarafından yapılan bir hikâyeyi dosyaya kaydeder. Player sınıfı için bir hikâyeyi dosyadan okuyup oynanabilir hale getirir.

Metotları:

InitializeStory(String fileName): Parametre olarak verilen dosya isminden hikayeyi okuyarak oynanabilir hale getirir.

saveStory(): İçinde tuttuğu veriyi belirli bir formatta dosyaya yazar.

showQuestion(): currQuestion değişkeni içinde tutulan sorunun showQuestion() metodunu çağırır.

IsFinish(): currQuestion değişkeni içindeki sorunun bir son olup olmadığını geri döndürür.

isAnswerLegal(String givenAnswer): currQuestion değişkeninin cevap listesinde verilen cevap yoksa yanlış geri döndürür.

addQuestion(): Soru ekler.

GetQuestionByID(int id): id'ye göre soru alır.

Sınıf Değişkenleri:

BinarySearchTree<Question> QuestionSearchTree: Maker sınıfı soru ekleyip o soruyu başka bir soruya bağlamadığı sürece bu ağaçta tutulur.(Question sınıfının ID değişkenine göre sıralanır)

PriortyQueue<Node> storyQueue: İlk soru en yüksek önceliğe sahip olmak üzere soruları tutar.

Question currQuestion: Üzerinde işlem yapılan soru

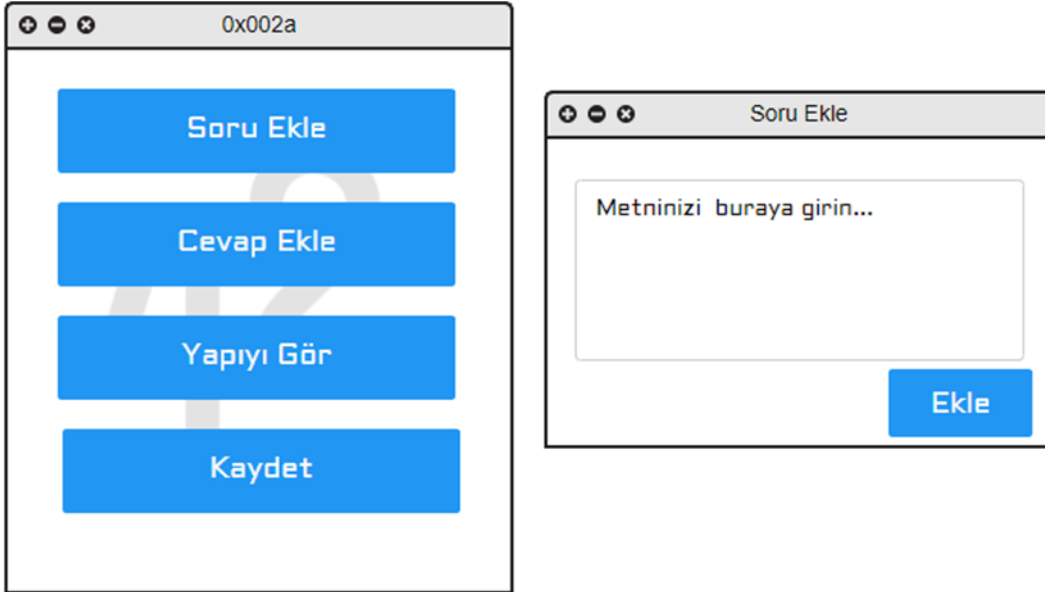
CStats GameChar: Oyun için kullanılan karakterin özellikleri

GUI ÖRNEKLERİ

Temsili Maker Görselleri



İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.



İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.

Cevap Ekle

Soru numarası

Metninizi buraya girin...

Özellige etki ekle

Metninizi buraya girin...

Özellige etki ekle

Ekle

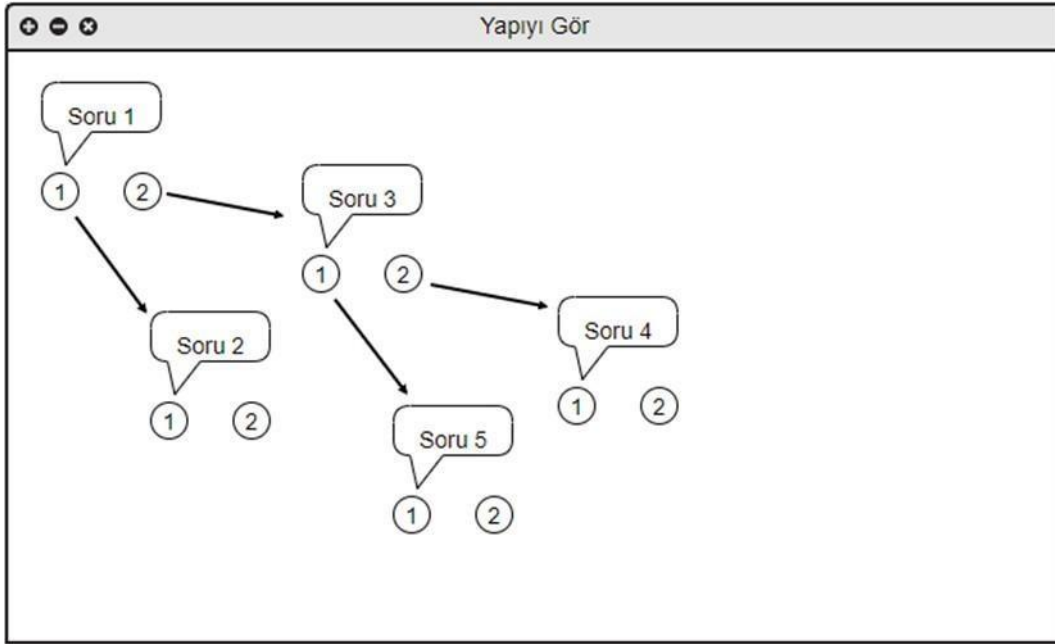
Etki

Özellige etki ekle

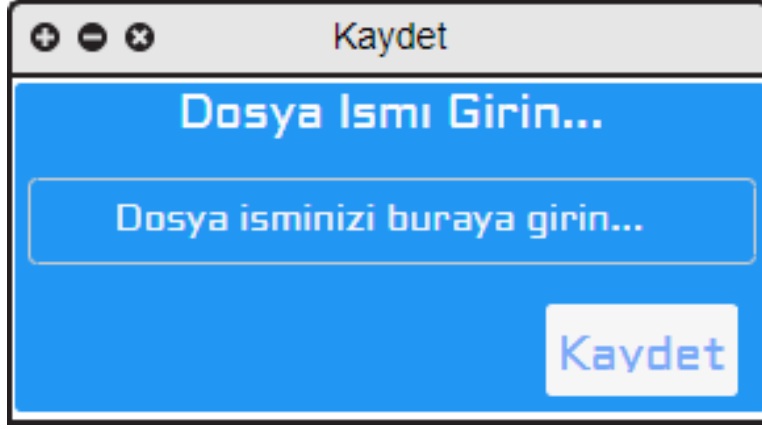
Özellik ismi Etki Puan

Ekle

İçeriği daha iyi görüntüleyebilmek için yakınlaştırdınız.

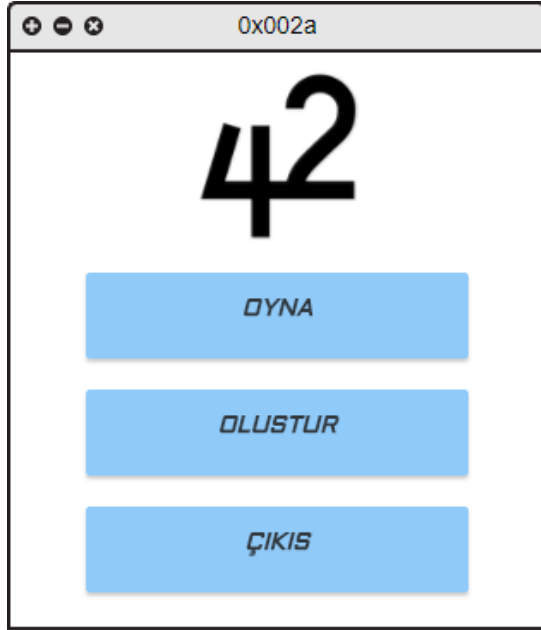


İçeriği daha iyi görüntüleyebilmek için yakınlaştırdınız.

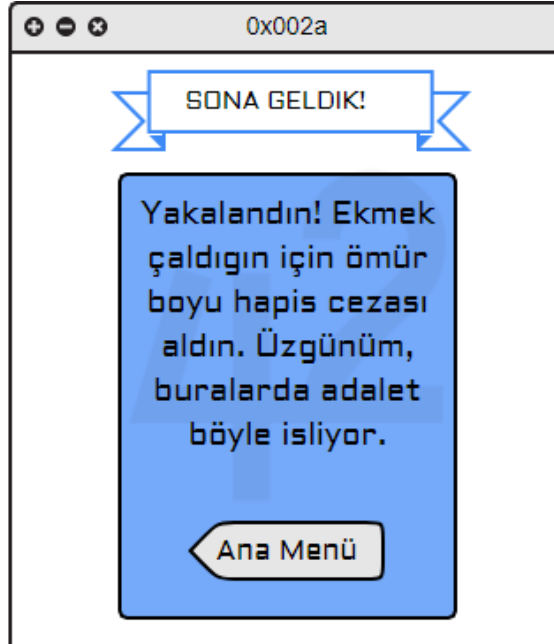
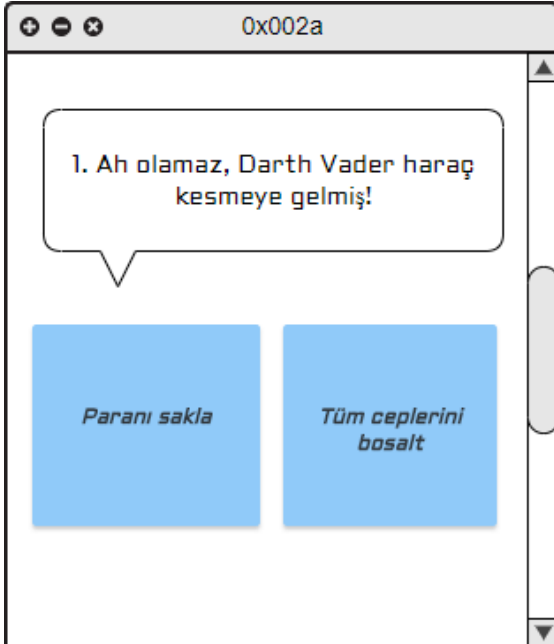


İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.

Temsili Player Görselleri



İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.



İçeriği daha iyi görüntüleyebilmek için yakınlaştırınız.

TAKIM ÜYELERİ

- . *AZMİ UTKU SEZGİN*
- . *BARIŞ ŞAHİN*
- . *DOĞA UYSAL*
- . *GÜNSELİ ÖZKAN*
- . *MUAZ KURT*
- . *ONGUN ALP BABA*
- . *VEFİK FIRAT AKMAN*