

Милионер

Структура на проектот

Во оваа ппроектна задача е имплементирана играта Кој сака да биде милионер која служи за тестирање на знаењето на играчите. Играчот има цел да одговори 15 прашања со понудени четири одговори, а за помош може да се послужи со еден од трите џокери:

Повикај пријател - на играчот му се дава одговор и процент на сигурност во одговорот.

50/50 - од четирите понудени одговори се кријат два неточни

Прашај ја публиката -за секој одговор, на играчот му се дава процент на сигурност дека одговорот е точен

Со секој точен одговор играчот е поблиску до целта, но секое наредно прашање е потешко од претходното.

Со цел полесно да се разбере кодот се користење сервисно слојниот шаблон за организација на логиката. Бизнис логиката е организирана во два сервиси: QuizService.cs и FileService.cs.

Квиз сервисот содржи методи кои се одговорни за прашањата:

Најпрво се иницијализира сервисот и тој пристапува до фајлот questions.json во кој се содржат прашањата. Тие се организирани во повеќе игри каде што секоја игра има сет од 15 прашања кои го содржат текстот на прашањето, листа од одговори и индекс на точниот одговор. Дополнително во GameState класата се чуваат информации за тековната игра:

- Искористени џокери
- Индекс на сетот на прашања
- Индекс на последното одговорено прашање

После иницијализација можат да се користат другите методи на сервисот:

Question GetQuestion(int index); - за даден индекс враќа прашање.

bool ValidateAnswer(int questionIndex, int answerIndex); - валидира прашање и одговор за даден индекс

AskAFriendModel CallAFriend(int questionIndex); - за дадено прашање дава објект што содржи одговор и колкав е процентот на сигурност во тој одговор.

Работи така што во 9 од 10 случаеви го избира точниот одговор и има голема сигурност за тоа, а во 1 од 10 случаеви избира погрешен одговор и има мала сигурност во тоа. Во двата случаеви може да се има иста сигурност, па поради тоа не можеме секој пат да се ослониме на тоа.

List<int> AskAudience(int questionIndex); - за дадено прашање дава листа од четири целосни броеви каде што секој број претставува сигурност во одговорот со истиот реден број. Работи така што за секое од прашањата генерира случаен број од 0 до 100 каде што точниот одговор е од 30 до 60, а погрешните од 5 до 30. Одкако ќе се земат броевите, се пресметува нивната сума и разликата се додава/одзема на случаен одговор со цел сумата да биде 100.

List<int> FiftyFifty(int questionIndex); - за дадено прашање дава листа од два целосни броја каде што едниот е точниот одговор, а друг е случаен одговор.

Вториот сервис е `FileService.cs` кој во себе содржи методи за вчитување и запишување на моменталната состојба на играта во фајл со цел снимање и продолжување на играта. Тоа го прави така што

```
public static GameState LoadGameStateFromFile() - ги чита податоците на фајлот  
gameState.json и ги серијализира. Оваа метода се повикува при клик на копчето Load  
game
```

```
public static void SaveGameStateToFile(GameState gameState) - добиената состојба на игра  
ја зачувува во gameState.json. Оваа метода се повикува при клик на копчето за гасење на  
играта
```

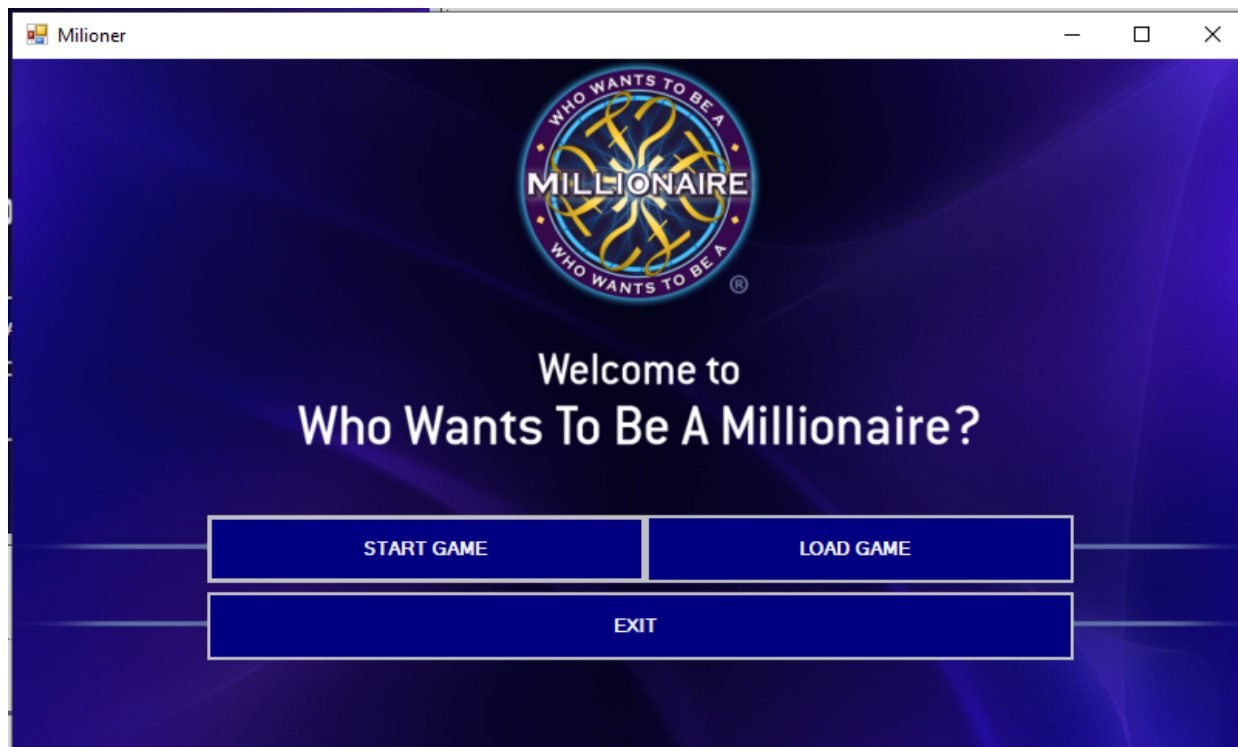
При старт на играта се популираат текстот на прашањето со индексот што се чува во `gameState` и неговите одговори. `ListBox` компонентата со прашања се популира и прашањата што биле пред тоа со тековниот индекс се штиклираат.

При селектирање на одговор се валидира дали тој е точен, а доколку е се проверува дали тековното прашање е последно. Доколку не е индексот се зголемува за еден, се ажурира `ListBox` компонентата и се вчитува прашањето од почеток. Доколку одговорот е погрешен му се прикажува порака на корисникот.

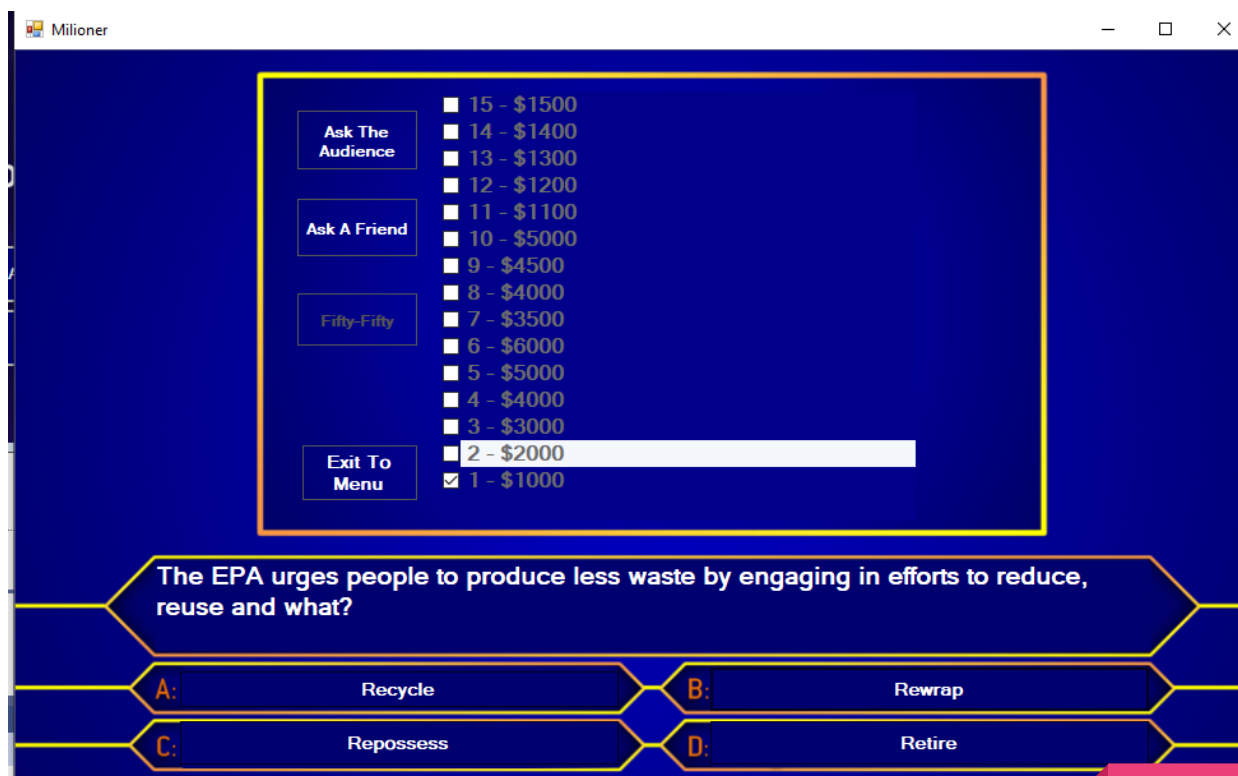
Како се користи играта (упатсво)

При старт на апликацијата се гледа првичниот екран на кој може да се одбере нова игра или пак да се продолжи со претхдно зачувана позиција на квизот. Апликацијата е проследена со звук.

По одбирање на опцијата `Start Game` играчот се пренасочува на екранот за квизот на кој се гледа листа од прашања кои следат со сума која се добива, и соодветно се поместува (штиклира) прашања кои се одговорени. Од левата страна се наоѓаат копчињата за



помош од публиката, повикај пријател и 50-50. Овие џокери се користат само еднаш во текот на



играта и стануваат затемнети по нивно користење. Со кликање на копчето Exit To Menu корисникот се враќа на почетниот екран и моменталната состојба на играта се зачувува и играчот може повторно да продолжи од кадешто претходно застанал. Целта на играта е да се одговорат точно сите прашања.

