

Technická dokumentácia

Globe aplikácia

DataHolders

LearningCourse:

Trieda obsahujúca atribúty kurzu namapované z databázy. Je používaná len na ukladanie údajov a neposkytuje žiadnu funkcionálnosť.

Test:

Trieda obsahujúca atribúty testu namapované z databázy. Je používaná len na ukladanie údajov a neposkytuje žiadnu funkcionálnosť.

UserProfile:

Trieda obsahujúca atribúty používateľa namapované z databázy. Je používaná len na ukladanie údajov a neposkytuje žiadnu funkcionálnosť.

Points:

Triedy obsahujúce pozície vo formáte GPS. Sú používané na ukladanie dát a neposkytujú žiadnu funkcionálnosť. Typy pozícií sú nasledovné:

- `public class GPSPoint`: jeden bod s hodnotami longitude a latitude.
- `public class GPSPolygon`: množina bodov, ktoré ohraničujú oblasť.
- `public class GPSLineString`: množina bodov, ktoré reprezentujú čiaru.

ITask:

- `bool Init()`; V tejto metóde je potrebné implementovať udalosti, ktoré sa majú vykonať pri začiatku úlohy, ako sú zobrazenie prvkov UI, grafických prvkov na mape atď. Vracia true ak sa úlohu podarilo inicializovať.
- `bool Update()`; V tejto metóde je potrebné implementovať udalosti, ktoré sa vykonávajú každú snímku, ako napríklad aktualizácia časomier, presunutie vľavky na pozíciu myši a podobne. Taktiež môže sledovať, či bola vykonaná akcia (napr. klik myšou na špecifické miesto), ktorá by ukončila úlohu. Vracia true ak takáto akcia bola vykonaná.
- `void Complete()`; V tejto metóde je potrebné implementovať udalosti, ktoré sa majú vykonať na konci úlohy, ako zavretie prvkov UI, odstránenie grafických prvkov na mape atď.
- `void ClearAnswers()`; V tejto metóde je potrebné implementovať vrátenie úlohy do stavu, v ktorom neobsahuje žiadne riešenie, avšak bez reštartu časomier a podobne.
- `double Evaluate()`; V tejto metóde je potrebné vyhodnotiť odpovede a vrátiť získané body.

Task:

Abstraktná trieda implementujúca ITask. Navyše obsahuje dáta, ktoré Task musí obsahovať:

- `public string Name`: Meno úlohy, ktoré sa bude zobrazovať v UI
- `public string Description`: Popis so zadaním úlohy, ktorý sa bude zobrazovať v UI
- `public bool Locked`: udáva, či je úloha zamknutá. Zamknutie prebehne ak uplynula časomiera, alebo ak bol test odovzdaný. Ak je Task zamknutý a bol zavolaný Init(), je potrebné zobrazovať správne odpovede.
- `public bool OutOfTime`: udáva či zamknutie úlohy bolo spôsobené uplynutím časomier.

PointTask:

Trieda rozširujúca abstraktnú triedu Task a teda implementujúcu rozhranie ITask. V tejto triede je implementovaná funkcionálnosť úlohy, pri ktorej je potrebné určiť jeden bod na mape.

1. Informácie o hľadanej pozícií:
 - `public string` Id: Id úlohy v databáze
 - `public int` Type: Id typu úlohy v databáze
 - `public int` Time: Povolený čas z databázy
 - `public string` CityName: Meno hľadaného bodu v databáze
 - `public string` CityDescription: Dodatočný popis hľadaného bodu
 - `public int` Population: populácia (ak je mesto)
 - `public GPSPoint` Point: GPS súradnice hľadaného bodu
 - `public string` PointName: Meno hľadaného bodu (môže byť upravené oproti názvu objektu v databáze)
2. Podmienky úlohy
 - `public TimeSpan` TimeLimit: čas prevedený z formátu string na TimeSpan
 - `public double` Score: získané body podľa Evaluate() metódy.
 - `public GPSPoint` Answer: odpoveď, ktorú používateľ zadal.
 - `private float` SpentTime: uplynutý čas, k tejto hodnote sa pripočítava delta trvania jednej snímky každý Update cyklus. Ak je tento čas rovný alebo väčší ako TimeLimit, časový limit bol dosiahnutý a úloha sa zamkne.
3. Grafické komponenty
 - `private UIManagerScript` UIManagerScriptObject: referencia na objekt, v ktorom sa nachádzajú všetky UI komponenty.
 - `private InfoBoxManagerScript` Ui: referencia na objekt InfoBox v UI.
 - `private ActiveTaskManagerScript` ActiveTask: referencia na objekt ActiveTask v UI.
 - `private GPSPolygon` Line: čiara, ktorá je vykreslená medzi hľadaným bodom a odpoveďou, ak odpoveď nebola správna.
 - `private LineRenderer` Renderer: objekt, ktorý vykreslí čiaru reprezentovanú v premennej Line.
 - `private GameObject` labelHolder: objekt, ktorý vykresľuje na mape text obsahujúci meno hľadaného bodu, a vzdialenosť odpovede od hľadaného bodu.

Metódy tejto triedy implementujú funkcionality vyžadovanú rozhraním ITask.

TaskMultichoice:

Trieda rozširujúca abstraktnú triedu Task a teda implementujúcu rozhranie ITask. V tejto triede je implementovaná funkcionality úlohy, kde je študentovi položená otázka a má vybrať správne možnosti.

- `public string` Id: Id úlohy v databáze
- `public int` Type: Typ úlohy v databáze
- `public int` Time: Povolený čas z databázy
- `public Dictionary<string, bool>` Answers: Možnosti v pároch string a bool. String udáva text možnosti, bool či je správna.
- `public Dictionary<int, string>` Hints: Nápoveda (NYI) k možnostiam v Answers. Int reprezentuje index možnosti ku ktorej sa vzťahuje, string text nápovedy.
- `public TimeSpan` TimeLimit: čas prevedený z formátu string na TimeSpan
- `private UIManagerScript` UIManagerScriptObject: referencia na objekt, v ktorom sa nachádzajú všetky UI komponenty.
- `private InfoBoxManagerScript` Ui: referencia na objekt InfoBox v UI.
- `private ActiveTaskManagerScript` ActiveTask: referencia na objekt ActiveTask v UI.

- `public float SpentTime`: uplynutý čas, k tejto hodnote sa pripočítava delta trvania jednej snímky každý Update cyklus. Ak je tento čas rovný alebo väčší ako TimeLimit, časový limit bol dosiahnutý a úloha sa zamkne.
- `public List<Toggle> Toggles`: Referencia na zaškrťavacie políčka s možnosťami v UI
- `private List<bool> SelectedAnswers`: zoznam zaškrtnutých políčok. Pre zaškrtnuté políčko obsahuje true a pre nezaškrtnuté false. Poradie sa zhoduje s poradím v Toggles.

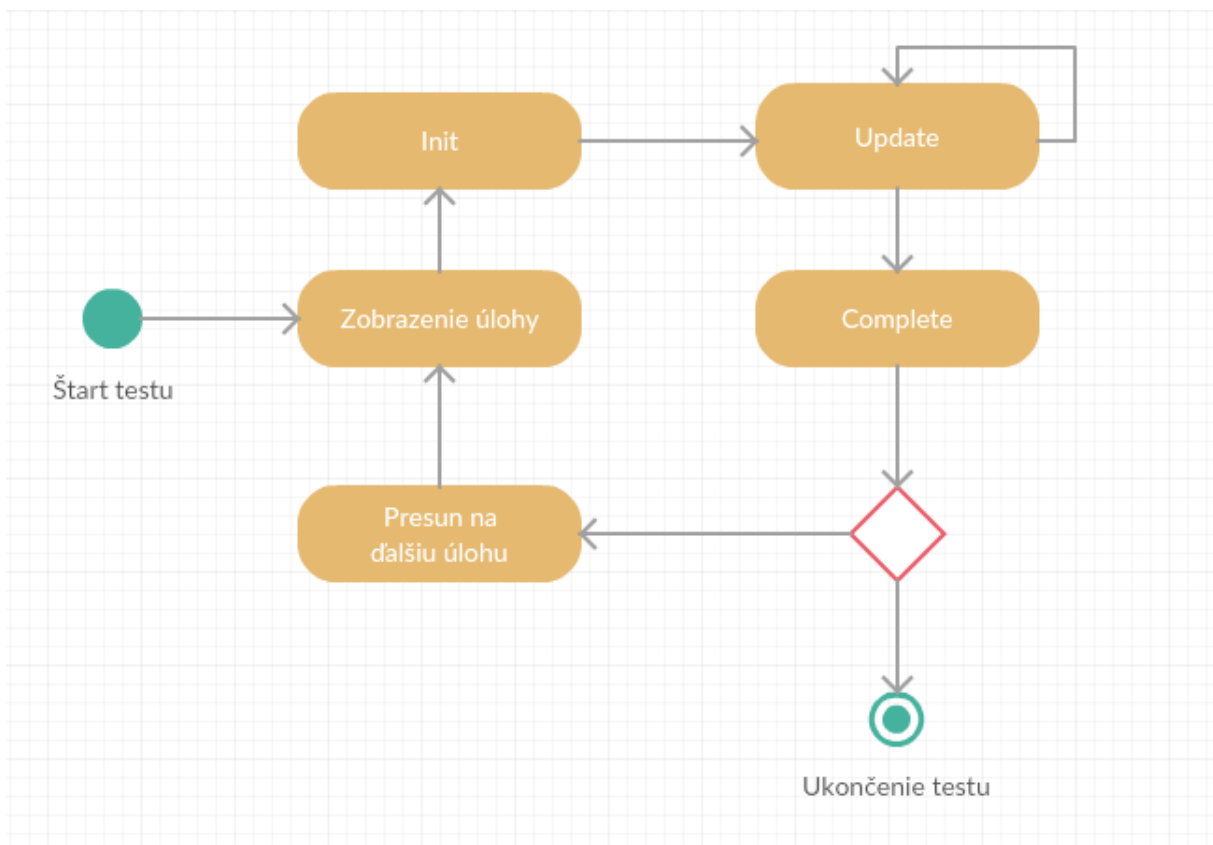
Metódy tejto triedy implementujú funkcionality vyžadovanú rozhraním ITask.

TaskPolygon:

Táto trieda je zatiaľ neimplementovaná. Mala by obsahovať úlohu, kde študent má nájsť bod, ktorý sa nachádza v polygóne.

TaskManager:

Táto trieda manažuje priebeh testu. Zobrazuje postupne úlohy. Pri štarte testu sa zobrazí prvá úloha pomocou jej volania Init(). Následne Update() TaskManagera volá Update() aktuálnej úlohy pri každej snímke. Keď je vykonaná akcia ktorá úlohu ukončí (presun pomocou tlačidiel v UI, odovzdanie testu), zavolá sa príslušná metóda v TaskManagere MoveBackwards(), MoveForward() alebo Submit() tak sa zavolá Complete() aktuálnej úlohy. Ak úloha bola ukončená volaním MoveBackwards() alebo MoveForward(), pokračuje sa ďalším volaním Init() na novú úlohu.



- `public Test Test`: test získaný z databázy, ktorý práve prebieha
- `public int TaskIndex`: index úlohy, na ktorej sa test práve nachádza
- `private bool TestRunning`: udáva či je aktívny test, aby sa predišlo zbytočným volaniam Update();
- `private UIManagerScript UIManagerScriptObject`: referencia na UI objekty

- `private` InfoBoxManagerScript Ui: referencia na InfoBox v UI
- `private` ActiveTaskManagerScript ActiveTask: referencia na ActiveTask v UI
- `public bool` Submitted: udáva či už bol test odovzdaný. V odovzdanom teste nie je možné robiť zmeny.

Metódy:

- `public void` StartTest(`int` id): táto funkcia vyžiada test s ID z databázy a nastaví prvú úlohu ako aktívnu úlohu. Na to je potrebné nastaviť TaskIndex a zavolať Init() na úlohu v Test.Tasks[TaskIndex]. Taktiež je potrebné zobrazíť UI komponenty zobrazujúce všeobecné informácie o teste, horný panel obsahujúci meno aktuálnej úlohy, číslo úlohy a čas.
- `public void` Update(): táto funkcia sa spúšťa pri každej snímke a volá Update() momentálnej úlohy uloženej v TaskIndex
- `public void` MoveBackwards(): finalizuje aktuálnu úlohu a presunie sa na úlohu s o jedna menším indexom. Taktiež volá funkciu RefreshMoveButtons();
- `public void` MoveForward(): finalizuje aktuálnu úlohu a presunie sa na úlohu s o jedna väčším indexom. Taktiež volá funkciu RefreshMoveButtons();
- `private void` RefreshMoveButtons(): Ak je TaskIndex na prvej úlohe, schová tlačidlo umožňujúce pohyb doľava. Ak je TaskIndex na poslednej úlohe, schová tlačidlo umožňujúce pohyb doprava. Ak je TaskIndex na inej úlohe, zobrazí obe tlačidlá.
- `public void` ClearAnswers(): zavolá ClearAnswers() na aktuálnu úlohu
- `public void` Submit(): zavolá Complete() na aktuálnu úlohu a Evaluate() na všetky úlohy. Získané body zobrazí v UI.
- `public void` CheckAnswers(): zamkne test aby sa nedal upravovať a zobrazí prvú úlohu aj so správnymi odpoveďami. Pohyb dopredu a dozadu v zozname úloh je naďalej možný, avšak nedá sa upravovať odpoveď.
- `public void` FinishTest(): ukončí prehliadanie odpovedí a vráti sa do menu.

StagitCamControler

Táto trieda implementuje funkcie kamery.

- `public Transform` target; cieľ kam sa má kamera pozerať, teda model Zeme
- `public float` distance = 2.0f; vzdialenosť od Zeme
- `public` Vector3 startRotation; počiatočná rotácia kamery
- `public float` xSpeed = 120.0f; rýchlosť otáčania v smere osi x
- `public float` ySpeed = 120.0f; rýchlosť otáčania v smere osi y
- `public float` distanceMin = 0.2f; minimálna vzdialenosť od Zeme pri zoome
- `public float` distanceMax = 30f; maximálna vzdialenosť od Zeme pri zoome
- `private float` AxisX = 0.0f; počiatočná pozícia kamery v scénových súradniciach
- `private float` AxisY = 0.0f; počiatočná pozícia kamery v scénových súradniciach
- `private float` lookAngle = -25.0f; náklon kamery
- `private bool` ControlEnabled: udáva či je možné ovládať kameru (kamera je zamknutá v menu a pri prezeraní odpovedí)
- `private GPSPoint` viewPoint: GPS point, kde sa kamera má pozerať
- `float` x = 16.0f; náklon kamery v x smere
- `float` y = 228.0f; náklon kamery v y smere

Metódy:

- `void` LateUpdate (): vykonáva zmeny pozície kamery podľa vstupov z klávesnice a myši
- `public void` EnableControl(): umožní ovládanie kamery pomocou klávesnice a myši

- `public void DisableControl()`: znemožní ovládanie kamery pomocou klávesnice a myši
- `public IEnumerator MoveToGPS(GPSPoint point, float delay)`: presun na GPS súradnice počas času delay
- `IEnumerator moveCameraToObject(GameObject obj, GPSPoint gpsPoint, float delay)`: presun na pozíciu objektu počas času delay
- `public void MoveCameraToFlag(GameObject flag)`: presun na pozíciu vlajky označujúcej odpoveď

ActivityVerbs

Tento enumerátor definuje kľúčové slová (tzv. Verbs), ktoré je možné použiť pri zaznamenávaní aktivity študenta do LRS. Zadávajú sa ako argumenty do `LogActivity` metódy `Server Communicator`. Použitie iných kľúčových slov ako tých v enumerátore pri zaznamenávaní aktivity bude viesť k chybe na strane servera, ktorá spôsobí, že daný záznam nebude uložený.

ServerCommunicator

Táto trieda zabezpečuje všetku komunikáciu s API `EduVirtual`. Trieda sa stará o uloženie autorizačného tokenu, takže je stavová. Trieda je statická, takže je zabránené vytvoreniu viacerých inštancií. `Public` metódy sú určené ostatným častiam aplikácie na komunikáciu so serverom, `private` metódy zabezpečujú vykonanie interných operácií `server communicator`.

- `public Transform target`; cieľ kam sa má kamera pozeráť, teda model Zeme
- `private static string authorizationToken`;
- `private static UserProfile user`;
- `private static string serverIp = "http://80.241.209.214:3000"`;

Metódy:

- `static ServerCommunicator()`: konštruktor triedy `ServerCommunicator`
- `public static UserProfile Login(string loginMail, string userPassword)`: prihlásenie používateľa na server a získanie profilového objektu a uloženie prístupového tokenu
- `public static UserProfile GetUserDetails()`: vráti profilový objekt s podrobnosťami o práve prihlásenom používateľovi
- `private static void UpdateUserProfileObject()`: získa zo servera údaje o prihlásenom používateľovi
- `public static List<LearningCourse> GetAvailableCourses()`: vráti kurzy a cvičenia dostupné práve prihlásenému používateľovi, metóda nevracia obsah testov v cvičeniach
- `public static Test GetSelectedTests(string exerciseId)`: získa a vráti vybraný test spolu s celým jeho obsahom
- `private static string SendPostRequest(string apiAddress, string requestJson, bool isRequestLogin = false)`: metóda odošle POST request na serverovú API a získa JSON s odpoveďou

- `private static void CheckRequestSuccess(string responseJson, bool wasRequestLogin = false)`: zistí, či server vrátil na request pozitívnu alebo negatívnu odpoveď
- `public static byte[] GetProfilePicture()`: získa zo servera profilový obrázok práve prihláseného používateľa
- `public static void LogActivity(ActivityVerbs verb, int courseId = 0, int exerciseId = 0, int taskId = 0, int hintId = 0)`: pošle na server príkaz na zaznamenanie aktivity do LRS
- `private static void SaveScore(int spentTime, bool correct, int points, int exerciseId, int taskId, double distance = -1)`: uloží výsledné skóre testu do databázy, momentálne nepoužívaná metóda
- `private static UserProfile FastLogin(string loginToken)`: prihlásenie používateľa pomocou QR kódu z edukačnej webovej stránky
- `public static UserProfile CrossLogin(string loginToken)`: prihlásenie používateľa pri príchode z edukačnej webovej stránky
- `public static Dictionary<string, CourseContent> ListCoursesAndTests()`: vráti mená všetkých kurzov, cvičení a testov
- `public static string OpenWebMaterials()`: otvorí edukačnú webovú stránku a prenesie prihlasovacie údaje
- `private static string GenerateSHA256String(string inputString)`: prevedie heslo na SHA256 hash
- `private static string GetStringFromHash(byte[] hash)`: z hashu získa a vráti string