

Virtuálne výučbové prostredia riadené Petriho sieťami - od prototypu k všeobecnému softvérovému systému

Štefan Korečko, 2023

verzia 2.0, apríl 2023

Tento dokument obsahuje vybrané detaily špecifikácie všeobecnejšieho softvérového výučbového systému pre webovú virtuálnu realitu, postaveného na prototypu popísanom v práci https://hron.fe.i.tuke.sk/~korecko/zavPr/DP_Kasela.pdf. Prototyp je dostupný na adrese <http://hron.fe.i.tuke.sk/~korecko/VRHistoryClass/views/>.

Používatelia systému (Users)

Budeme mať 3 role:

- Študent/ka (Student) -
 - Riešenie vybranej úlohy (Ale len z tých, ktoré sú mu/jej priradené).
 - Zadanie a editácia údajov o svojom používateľskom účte.
 - Prezeranie a analýza svojich výsledkov riešenia úloh.
- Učiteľ/ka (Teacher)
 - Vytvorenie, editácia, vymazanie úlohy.
 - Priradenie / odobranie úlohy študentovi.
 - Prezeranie a analýza výsledkov riešenia úloh (študentských aj svojich).
 - Riešenie vybranej úlohy.
 - Zadanie a editácia údajov o svojom používateľskom účte.
- Administrátor/ka (Admin) -
 - Všetko to, čo učiteľ/ka
 - Pridávanie, odoberanie a editácia používateľov, vrátane zmeny ich role.

Úloha (Task)

Úloha je aktivita používateľa/lky (väčšinou študenta/tky) vo virtuálnej scéne, riadená príslušným scenárom.

- Virtuálna scéna - interaktívna 3D scéna, vytvorená v A-Frame (html, css, js, assets)
- Scenár - Petriho sieť, na úrovni PT siete (tokeny typu UNIT), v PNML formáte exportovanom z CPN Tools a textové súbory s popiskami

Udalosti vo virtuálnej scéne

Používateľ môže svojou aktivitou v scéne (pohyb, interakcia s objektami v scéne) vyvolať 2 druhy udalostí:

- Udalosti nezávislé od scenára - nemajú žiadne prepojenie so scenárom (Petriho sieťou). Ich obsluha (event handling) závisí iba od príslušného js kódu.
- Udalosti závislé od scenára - Každá z nich je naviazaná na prechod v Petriho sieti. Pri výskyte takejto udalosti dôjde k pokusu príslušný prechod odpáliť. Obsluha udalosti závisí od toho, či sa prechod podarí alebo nepodarí odpáliť.

Definovanie a realizácia obsluhy udalostí závislých od scenára

Spôsob definovania a realizácia obsluhy udalostí závislých od scenára musí byť jednotný pre všetky úlohy, teda 3D scény a scenáre.

Definovanie obsluhy

Na úrovni kódu prechody Petriho siete a súvisiace závislé udalosti danej 3D scény popíšeme poľom objektov, pričom tieto objekty budú mať nasledovnú štruktúru:

```
{
  transitionName: String,
  ifTransitionFound: function(targets, replay),
  ifTransitionEnabled: function(targets, replay),
  ifTransitionDisabled: function(targets, replay),
  ifTransitionNotFound: function(targets, replay),
}
```

Ak nechceme urobiť nič, bude daná položka prázdna. Samozrejme, môže byť použitá viacnásobne tá istá funkcia (napr. pre `ifTransitionNotFound` a `ifTransitionFound`).

Parametre:

- `targets` - array of strings, id elementov v scéne, s ktorými bude obsluha pracovať. Takto umožníme priradenie toho istého prechodu rôznym elementom.
- `replay` - boolean, `true` ak k vykonaniu funkcie dochádza na základe aktivity používateľa, `false` ak zo záznamu.

Toto pole pri načítaní scény môžeme pridať k objektu `window`.

Realizácia obsluhy

Realizácia obsluhy, vykonávaná v hlavnej slučke 3D scény (update) potom bude nasledovná:

1. Zisti, či Petriho sieť obsahuje prechod `transitionName`. Ak nie, choď na krok 5.
2. Vykonaj funkciu `ifTransitionFound`.
3. Ak je prechod `transitionName` odpáliteľný, odpál ho a vykonaj funkciu `ifTransitionEnabled`. Choď na krok 6.
4. Ak prechod `transitionName` nie je odpáliteľný, vykonaj funkciu `ifTransitionDisabled`. Choď na krok 9.
5. Vykonaj funkciu `ifTransitionNotFound`. Choď na krok 9.
6. Medzi finálnymi prechodmi hľadaj odpáliteľný prechod.
7. Ak nenájdeš odpáliteľný finálny prechod, choď na krok 9.
8. Ak nájdeš odpáliteľný finálny prechod, obslúž ho podľa tohto postupu, okrem krokov 6-8.
9. Koniec obsluhy.

Finálne prechody

- Ich odpálenie znamená koniec scenára.
- od ostatných sa odlišujú začiatkom mena:
 - `finalSucc` - úspešný koniec
 - `finalFail` - neúspešný koniec
- nesmú byť priradené udalostiam v scéne
- vždy by mal byť vykonateľný maximálne jeden z nich.

Vykonanie obsluhy zo záznamu

Okrem vyvolania vykonania obsluhy aktivitou používateľa k nej môže dôjsť aj zo záznamu. K tejto situácii dochádza vtedy, keď je potrebné, napríklad z dôvodu výpadku spojenia, znovu načítať scénu, v ktorej má používateľ už rozpracované riešenie scenára. Ak by sme túto situáciu nijako neriešili, musel by používateľ po načítaní scény všetky kroky v nej vykonať znova. My však aktivitu používateľa v scéne zaznamenávame do databázy. O každom pokuse o odpálenie prechodu vieme

- či bolo odpálenie úspešné a
- aký prechod (`transitionName`) a na akých elementoch (`targets`) bol odpálený.

Podľa tohto záznamu vieme replikovať celú doterajšiu činnosť používateľa. Sú však činnosti pri ktorých obsluha nemôže byť rovnaká, ako keď aktivitu robil priamo používateľ. Napríklad keď pokus o odpálenie prechodu vyvolala aktivita spojená s prenášaním predmetu v scéne. Vtedy pri replikovaní zo záznamu musíme programovo vykonať aj samotné prenášanie, ktoré by normálne urobil používateľ.

Preto majú funkcie vstupný parameter `rep1ay`, vďaka ktorému vieme špecifikovať ktorá časť kódu sa vykoná iba pri replikovaní činnosti zo záznamu.

Priradenie prechodu k objektu v scéne

Môže byť cez A-Frame komponent, ktorý bude v parametroch mať `transitionName` a `targets`. Ale konkrétny moment vyvolania udalosti závislej od scenára nie je univerzálny, raz to bude kliknutie, inokedy to, že objekt sa niekde ocitne pri prenášaní.

Čo patrí k úlohe

- A-Frame scéna (HTML, CSS, skripty)
- Petriho sieť (PNML z CPN Tools)
- Súbory s popiskami v jednotlivých jazykoch. Popisky sú texty, ktoré sa v scéne zobrazia. Uchovávať ich takto osobitne potrebujeme, lebo význam tej istej závislej udalosti môže byť iný pre rôzne scenáre k tej istej scéne.

Scenár = Petriho sieť + súbory s popiskami

Súbory s popiskami sú textové súbory. Budú vo formáte JSON:

```
{
  "nazovPopisky1": "textPopisky1",
  "nazovPopisky2": "textPopisky2",
  ...
}
```

Popisky budú zodpovedať prinajmenšom prechodom Petriho siete, aj keď nie pre každý musí byť popiska. Názvy takýchto popisiek budú zhodné s názvami príslušných prechodov. V scéne môžu byť ale aj iné popisky (k objektom v scéne, ktoré s úlohou nesúvisia. V našej scéne sú to napríklad exponáty, na ktoré sa nepýtame, ale sú k nim informácie.)

Ako očakávame, že učiteľ bude tvoriť úlohu (scenár).

Optimisticky predpokladáme, že učiteľ bude vedieť vytvoriť novú úlohu (scenár) a to tak, že Petriho sieť vytvorí v CPN Tools a súbor s popiskami napíše v bežnom textovom editore.

K tomu bude potrebné

1. učiteľa zaškoliť (Petriho sieť, CPN Tools, JSON)
2. Poskytnúť učiteľovi vzorový scenár, v ktorom bude
 - Petriho sieť s prechodmi pre každú možnú závislú udalosť v scéne.
 - Súbor s popiskami zo všetkými možnými popiskami v scéne.
3. Mať možnosť spustiť úlohu (scéna + scenár) v návrhovom režime, kde sa pri príslušných objektoch zobrazia aj názvy im zodpovedajúcich prechodov Petriho siete a/alebo názvy popisiek.