Design Document – Motivation-based Adaptation Component

TUGraz – T3.4D – Apache 2.0 – Client side

About this component

This component will be implemented as client-side component. It is used for deciding if an intervention influencing the player's motivational state is needed; it supplies the motivational interventions and instance of these interventions. The component architecture will prevent multiple component creation; only one component per game is needed.

Component mechanics

Rule based mechanics trigger intervention based on the calculated motivation aspect values coming from the Motivation Assessment Component. At first glance, it seems clear, that only critical low motivation aspect values trigger an intervention. After detailed consideration we realize, that also the detection of high motivation aspect values can be meaningful in combination with low values for another motivation aspect value. For example, there might be two types of intervention for low values of attention. The first type is appropriate in case the player is very confident; the second type targets unconfident players. Based on these considerations, we implemented a rule-based mechanic working with propositional logic in combination with the motivation aspect values.

Component interfaces

- The component will deliver a list containing all motivational interventions suitable for a player. Identification strings represent these interventions. A C# representation in the component interface could be:

  ```
  List<String> getInterventions()
  ```

- An instance of an intervention can be accessed for a specific player. A C# representation in the component interface could be:

  ```
  String getInstance(String intervention)
  ```

Component dependencies/requirements

- The Motivation Assessment Component is used for retrieving the current motivational state of the player and the motivation model, which stores the motivation interventions and instances.
- The Game Storage is used as local storage.

Milestones

Milestone 1

- t1.1: Creating the first version of the design document, defining the API and creating a dummy component with the API implemented

Milestone 2

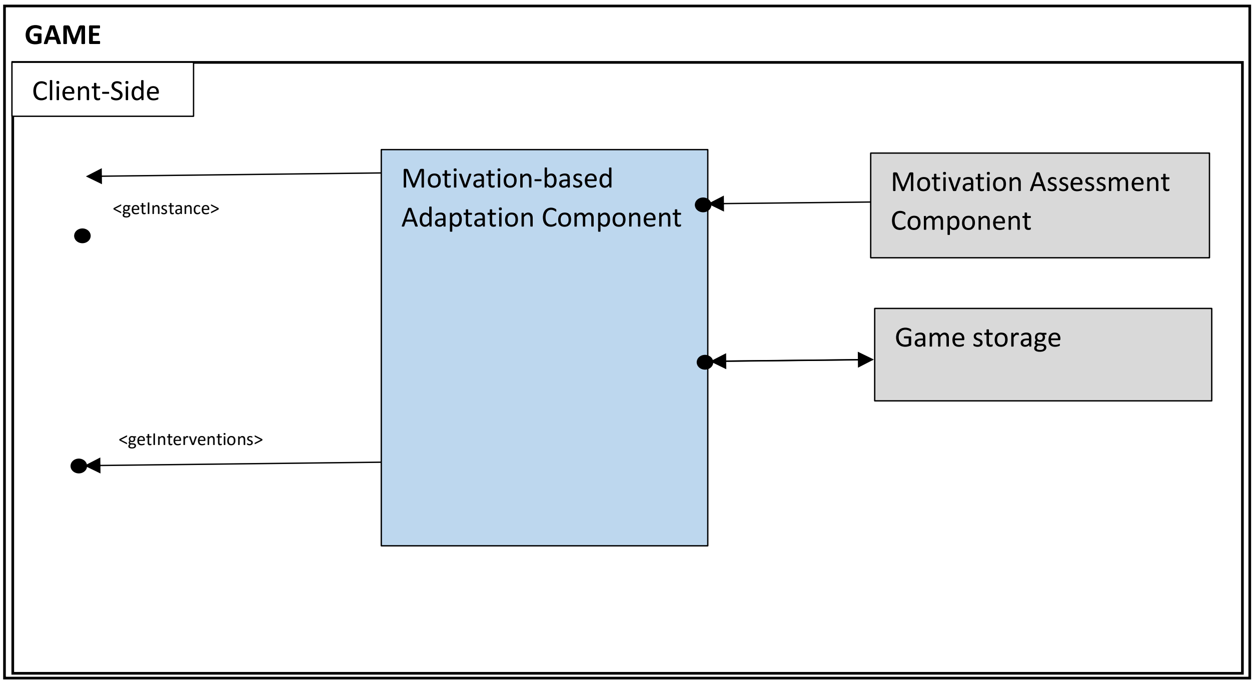
- t2.1: Create Software component in line with Component-Manager infrastructure.
- t2.2: Elaborate Settings-structure within the Component-Manager infrastructure.
- t2.3: Elaborate example data sets (XML structure).

- t2.4: Integrate tracking functionality.
- t2.5: Integrate Game Storage functionality into the component-functionality.

Milestone

- t3.1: testing the component with a game
- t3.2: instructions and scripts for building and deploying

Graphical representation


GAME
Client-Side
Motivation-based Adaptation Component
Motivation Assessment Component
Game storage
<getInstance>
<getInterventions>

← ... Flow of information

● ... Communication start

<*> ... Interface's exposed method

🟦 ... Component

Set up the Component

This component requires the Motivation Assessment Component as an underlying tool for requesting the current motivational state of the player. Furthermore, it just needs to be created:

```
MotivationBasedAdaptationAsset mbaa = MotivationBasedAdaptationAsset.Instance;
```

Use the component

It is now possible to request suitable interventions for the player via the method:

```
List<String> interventionTypeIDs = mbaa.getInterventions();
```

Furthermore, an instance of an intervention can be requested:

```
String interventionInstance = mbaa.getInstance(interventionTypeIDs[0]);
```

Deployment

For the source code the following GitHub-link can be used https://github.com/RAGE-TUGraz/MotivationBasedAssets - it contains the Visual Studio solution of the motivation based component. Furthermore, the broken links to external component DLLs need to be fixed for each project and the Bridge code need to be adopted to the new environment, e.g. changing the IDataStorage path.

For integration into Unity, the resulting DLLs need to put into a folder in the Unity working-directory.

Unit test

For executing unit tests, the source code need to be open in visual studio and all links need to be fixed. In the test-explorer all tests can be executed.