**Blackbear-Consultants for Teaching Tasks**

**Software Architecture Document**

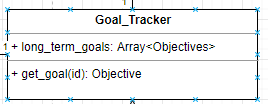
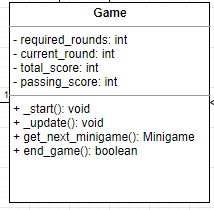
The architecture pattern Blackbear-Consultants decided to use for the Teaching Tasks application was Model-View-Controller (MVC) architecture. This pattern allows for the separation of presentation and interaction from the system data, so we are able to show how our application uses the data it collects in order to display and inform the user. It allows us to break down each component we plan on developing into three simple and logical components (those for Model, View, and Controller).

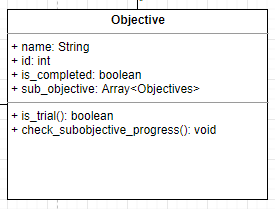
We use the Model components to help manage data for the system along with operations used on that data. The View component on the other hand manages how the data collected will be presented to the user. We have a lot of information we will be collecting, so we want to be able to display this data in a variety of ways to the user so they gain useful information. Finally, the Controller component helps and assists with user interaction (such as finger pressing, text input, etc.) and sends these interactions to the Model and View components.

As previously stated, we want the user to be able to interact with the data collected and view it in a number of ways. We also don’t know all of the requirements in which the interaction and presentation of the data will be used as the application grows, with more games and tasks implemented. We want our application to be able to expand on the types of games available, so that the children using the application can learn new things as they develop new skills and continue to learn. We want our application to grow and offer new games/tasks to continue to teach the child essential skills. Thus, the future requirements are unknown to us at this time. Below we have broken down each of our classes with their attributes and methods into each of the three categories used in the Model-View-Controller Architectural Pattern.

**Models:**

The following UML diagrams show the classes and their attributes of Teaching Tasks which fall under the Model components in the Model-View-Controller (MVC) Architectural Pattern:

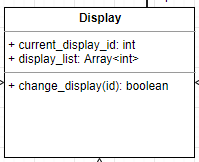


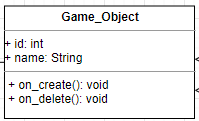


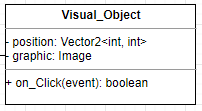
Model components show and represent knowledge. Mostly consist of data only. Methods that are used in Model components are those like Getters, Setters, etc. These are isolated and know nothing about the View components or Controller components.

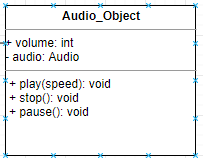
**Views:**

The following components are those that make up the View section of the Model-View-Controller (MVC) Architectural Pattern. In Teaching Tasks those components consist of:







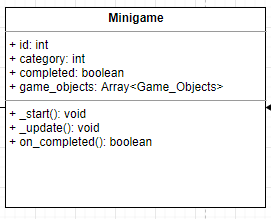


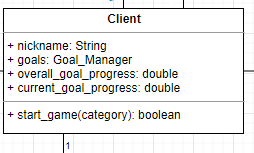
View components manage the display information. These are items much like the User Interface. They also only show you what you tell them to. They also lack the ability to transform, or validate logic. Thus, it only displays information through an event or some type of call back system. This part is also isolated, and knows nothing of the Model or Controller.

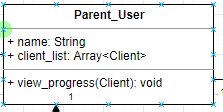
**Controllers:**

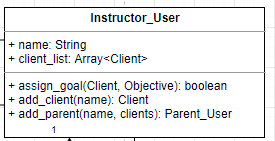
The following UML classes and attributes are those in the Teaching Task application that fall under the Controller portion of the Model-View-Controller (MVC) Architectural Pattern. A controller is the link between the user and system. The Controller places input in relevant views to present information in the appropriate places on the display. It presents the user with ways of giving commands and data. The Controller takes the output and turns it into appropriate messages and is able to pass said messages on to one or more views. It can be considered the brains of the application. This component takes user inputs, and decides how the model will change in regards to said input, and what the resulting view should look like. A controller sits between the Model and View, performs any type of data transformation which is necessary to get data from Model to the View. It also does data validation on input which comes back from the View component. A Controller knows about both the Model and View components.

cfdvz



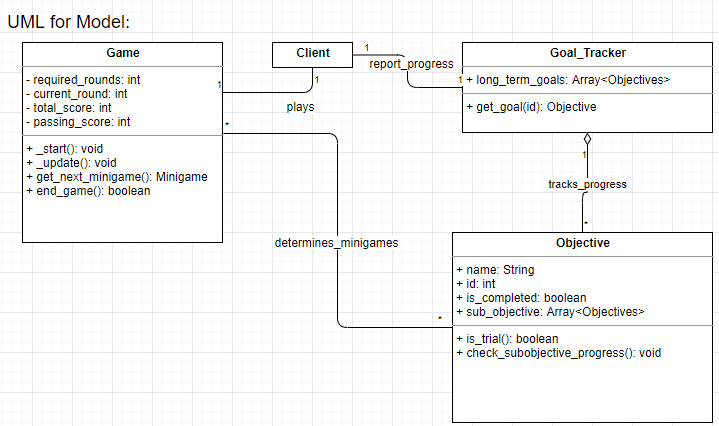




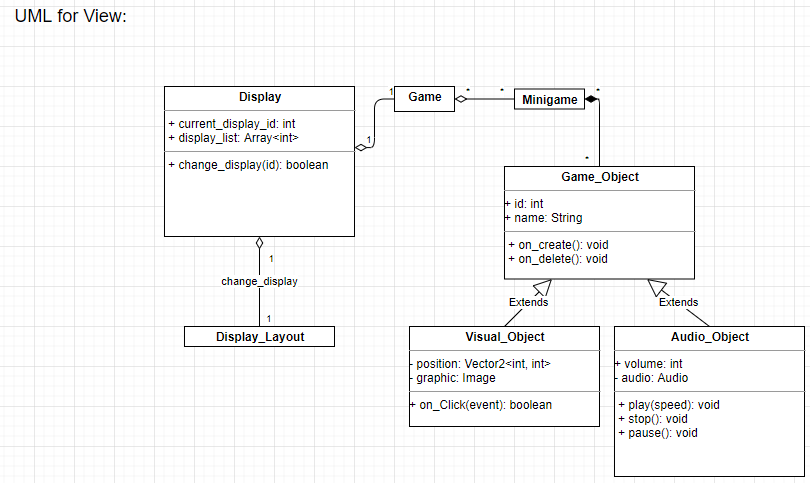


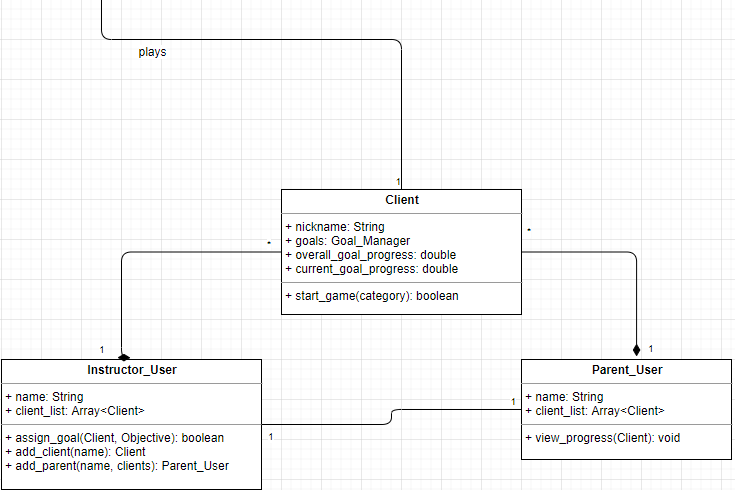
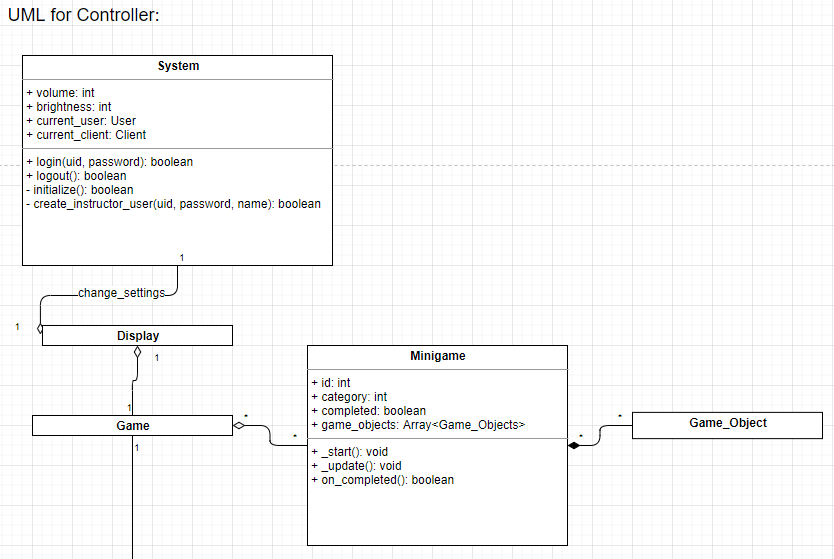
**UML for Model:**

The following UML diagrams show the complete path/connections that link the specific parts of the Model components which consist of the following: Game, Goal Tracker\*, and Objective.



The View Components consist of the following: Display, Display Layout, Game Object, Visual Object, and Audio Object. The UML diagram below displays how these are connected to each other and other components.



And Finally, the Controller components consist of the following: System, Minigame\*, Client, Parent User, and Instructor. The UML diagram displays how these components are connected to each other and the path they take which connects them to the other components of the Teaching Task application..