

Most of the Game Client was made based on the Class diagram, including model, network, and controller part. Only some names and functions were changed/added additionally. The most changes were done for creation of HalfMap, MapBrain to find treasure or castle, and path finder. Initially I wanted to have MapBrain to analyze the map and based on that analysis then to use MovementBrain to find next move and important items on map. In reality I created only MapBrain that fills the map with values and based on them finds those items inside one class. MapCreator has more additional functions apart from single "createHalfMap", we also need to check map for confirming to the rules.

Implemented MapBrain should have been sliced at least into 2 classes for better readability and to be more scalable.

In the context of making single MapBrain class, given complexity of implementing an algorithm, we chose making only one class and neglected splitting it in 2 to achieve path finding, at the cost of overcomplexity of MapBrain class because of time limit.

In the context of making map creating class, given it's complexity, we chose adding additional components and neglected planned version based on class diagram to achieve desired functionality, at the cost of putting everything only in one function inside the class because of overcomplexity issues.

In the context of leaving PathBrain class behind, given troubles with creating functioning AI, we chose adding it's functionality to MapBrain class and neglected splitting it in two to achieve fully functioning AI, at the cost of overcomplicated code because of time limits.