Name
Jakub Dunajko

Technical project documentation

The Greatest City

Subject: Object-Oriented Programming

1 General information

1. *Type of the project - game, city management simulation*
   This project is created to give an opportunity to learn some essential skills such as: resource management, decision making and it's impact on future actions.

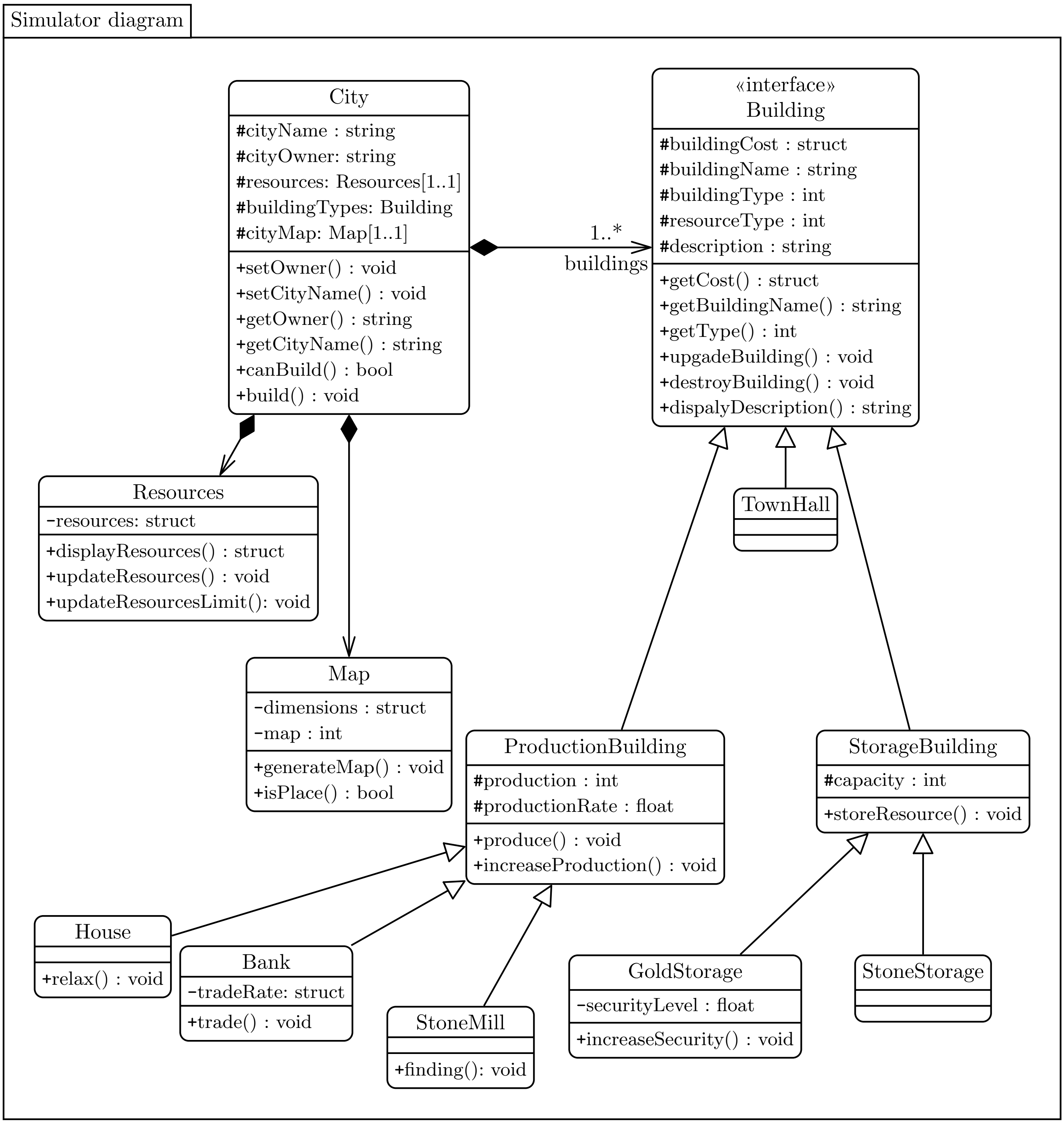
2. *Game plot*
   You are the city owner and your goal is to lead your city to it's greatness. You have to manage your resources: gold, population, stone and happiness. Each of this factor affects your city development in a certain way.

3. *Project target*
   Simulator is dedicated to those who want to fill their free time and learn some vital skills in pleasurable way.

4. *Project language*
   Project is written in C++ language

1. *Do czego system będzie wykorzystywany?*
2. *Jakie są granice używalności tworzonego systemu?*
3. *Jaki jest target systemu? (Do kogo system jest skierowany?)*
4. *Z czego wynika potrzeba utworzenia systemu?*

2 Class diagram



3 Requirements

3.1 Functional Requirements

1. canBuild()
   In this function we check resources and space on the map and decide whether we can build or not. Process in steps:

   - Get building type
   - Check for resources
   - Check for place on the map
   - return result

2. isSpace()
   In this function we check for space on the map for a new building. Process in steps:

- Check subsoil type(ground or water)
- Check if place is not taken
- return result

3. trade()
   In this function we process trading resources. Process in steps:
   - display tradeRate of resources
   - trade or leave
   - (trade) select resources to trade
   - (trade) select amount to trade
   - (trade) confirm trade

4. build()
   In this function we assign building to the place on the map and create its visual instance and consume resources

5. destroyBuiliding()
   In this function we remove buiding from the map and get some resources back.

6. updateResources()
   In this function we update current state of resources. Function is called whenever we build function or at the end of a day.

3.2 Non-functional requirements

1. *Easy to use*
   Any user can understand mechanics after few attempts

2. *Compatible with Windows system*

3. *Open to further development*

4 Project realisation

Libraries:

- *basic 3D library*
- *iostream*
- *vector*
- *chrono*

Milestones:

- *creating city map*
- *place first building on the map*
- *integration of classes Resources, Map and Building*
- *full integration of processes*

5 Criteria

- *Correct map generation*
- *Resources bar visibility*
- *Correct building placement*
- *Functions compatibility*
- *Intuitive interface*