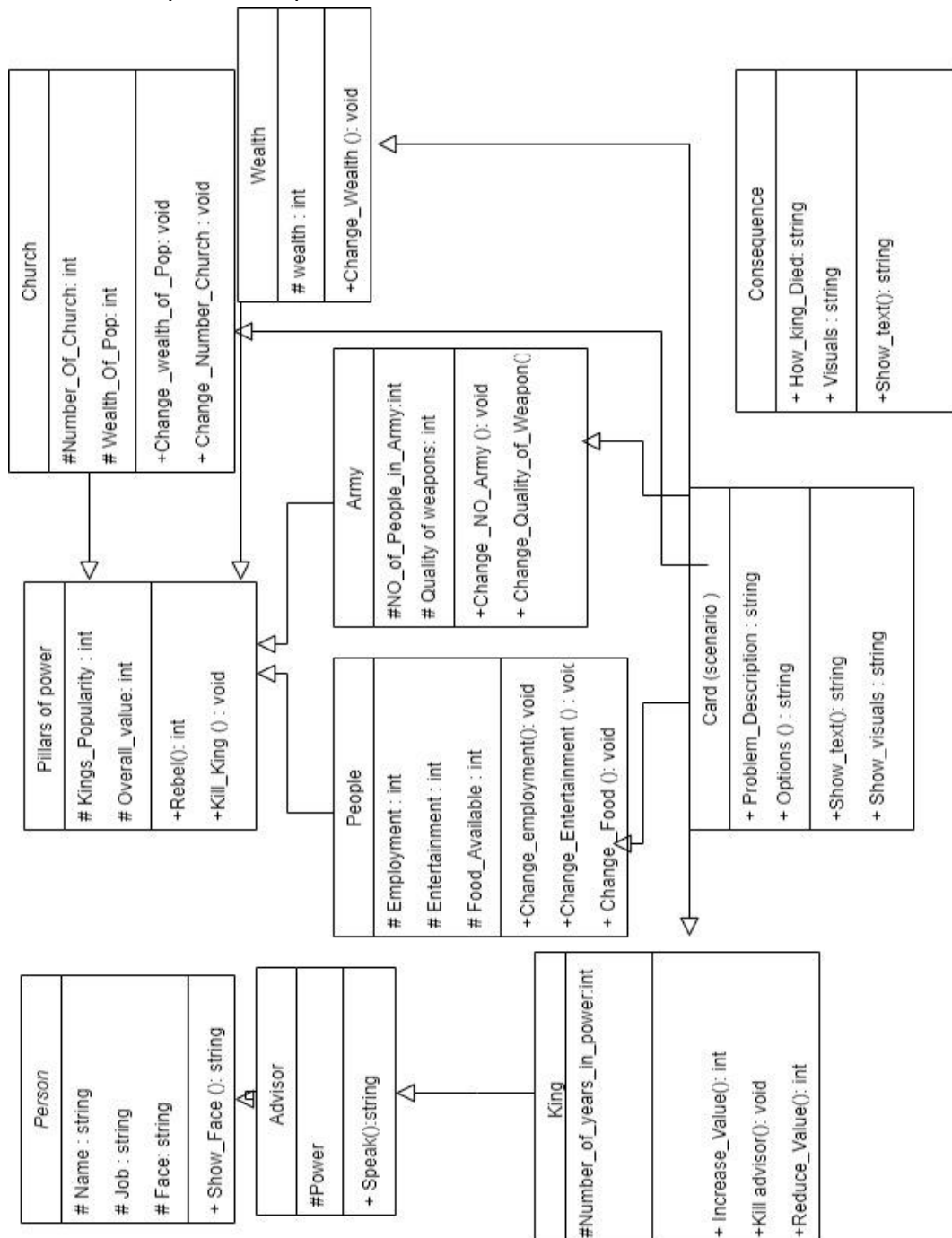


Use Case Description

The Monarch is a roleplaying game where the user is the king who is presented with problems by his advisors. The user can type 'y' or 'n' in the terminal to either accept or reject the advisors' suggestions (or click using graphical interface). The users' answers will affect different pillars of society such as the army, church, wealth, and people. The pillars must be kept at a balance as when the values of a pillar near zero or maximum, the user loses. The user is then required to enter a name for the new king from them to continue as.

After each decision, the value for each pillar is updates and presented in text (or ASCII or higher quality graphic) format to the user.

Each advisor is depicted with ASCII art which include a General, Queen, Treasurer, Excursionist, Spirit, and Priest. In the terminal below the ASCII will be the text containing their problems with the yes or no options.



Description of Classes

Person - Should be used as an abstract class where advisor and other classes are inherited from Person. A person in the game can only show_Face.

Advisor – Could be banker, general, king's guard and queen. They can speak to the king.

King – which is the user, can decide, and decisions will have impact on pillars of power. Once king is dead, a new king object is created by the user. User can set the king's name. The name must be different than the previous king's name.

Reduce_value and increase_value () are used, to increase or decrease a certain variables of powers such as church or Army. The power of king matters, because the impact of a decision is calculated from $\text{Power} * \text{Reduce_value}$, $\text{Power} * \text{increase_value}$. This is to make the power of king and advisors dynamic and dependant on the game progress.

Spirit – King will have random encounters with the spirit. Spirit can kill the king. It can also cast a spell, which can impact the satisfaction values.

Pillars of power - In each power, the satisfaction_value is calculated from other variables. The values are identified by the below symbols.



Church – satisfaction value is set at max 100, it is calculated from number of churches and wealth of pop. Each of those variables are set at 50.

Wealth – Similar to church

Army – Similar to church

People – Similar to church

Card(scenario) - In each scenario a text and options are presented to the user. There will be an if statement, if a certain answer is chosen certain values are deducted or increased.

Consequence class - This is intended to be activated when one of the satisfaction values are reached to zero and the king is killed or died. This class will show text of how king died followed by a visual.

Time. cpp – This is like the main.cpp. It is the storyline, where different classes and functions are called to progress the game.

TO DO

- Story line, content of up to 7 mins of play time
- Visualisations, how will the characters look, how will the user choose options
- Refine class design

CODING

- Person
- Advisor
- King
- Spirit
- Pillars of power
- Church
- Wealth
- Army
- People
- Card(scenario)
- Consequences
- Main.cpp

OTHER

- Graphical Interface
- Double checking the project requirements are met
- Testing of each file and overall game
 - As there are limited inputs user error should be easily limitable
 - Testing of each file can be conducted through altered, simple versions of the main function and checking that each class can interact as intended
 - Completing playthroughs
 - Achieving different deaths e.g., max or min values

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Week 8				Research and idea time/Timeline creation			
Uni Break 1	User description		Class creation				
Uni Break 2	Class creation				Planning for testing		
Week 9	Compilation and checking			Coding	MARKING	Coding/Testing	
Week 10	Coding/Testing						
Week 11	Debugging and Checking				MARKING		