Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G3: periodiC++

Class Design

Ahmet Kaan Demirci - 21050111031
Mehmet Emre Cebeci - 21050111037
Mustafa Özdemir - 21050111016
Teoman Güven - 23050151039

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp
Date: 11/12/2023

1.

Table of Contents

2. Introduction

The periodiC++ application is a comprehensive periodic table application. In this application, users can explore the periodic table in detail in many aspects and try fun quiz modes.

This report contains CRC cards and class diagram. In the CRC cards we wrote down all possible classes and tried to determine their resposibilities and collaborator. In the class diagram, we tried to determine the possible attributes and methods of the all classes that we determined before.

3. Class-responsibility-collaboration (CRC) cards

MainApplication

• Get main window • Get menu items • Display menu • Get user	• TableSection • SearchSection • LearnSection • QuizSection • SettingsSection

TableSection

• Get elements • Display the periodic table • Display legend (The section that shows what the colors used on a map represent is typically called the "legend") • Select an element • Get element properties • Select a element property • Get periodic properties • Select a periodic property • Change view	• MainApplication • LearnSection • SearchSection • Element • PeriodicProperty • ElementProperty

Element

• Generate element • Get element property • Get periodic property • Return Element informations • Return Element image	• TableSection • PeriodicProperty • ElementProperty

PeriodicProperty

• Represent different periodic properties • Return different color schema for each periodic property. • Return legend	• TableSection • Element

ElementProperty	
• Represent different element properties • Return different element representation for each element	• TableSection • Element

LearnSection	
• Get learn blocks • Select a learn block	• MainApplication • PeriodicTable • LearnBlock

LearnBlock	
• Return information of selected block	• Learn

SearchSection	
• Display elements in a listed form • Search an element • Select an element	• MainApplication • PeriodicTable

QuizSection	
• Get quiz modes	• MainApplication • Quiz

Abstract **QuizMode** LevelMode, ChallangeMode	
• Generate quiz mode • Start quiz	• MainApplication • LevelMode • ChallengeMode • Achievement

ChallengeMode QuizMode	
• Display time • Display score • Get questions • Display the questions • Set time limit • Get remaining time • Get random question • Get highest score • Get current score • Check score • Calculate score	• QuizMode • Question

LevelMode QuizMode	
• Display levels • Select level • Get target • Set target	• QuizMode • Level • Question

Level	
• Generate level • Check if unlocked • Get questions	• LevelMode • Question

Question	
• Generate question • Generate options • Display question • Check correct option	• Level • Challenge Mode

AchievementsSection	
• Display achievements as list • Get unlocked achievements • Get locked achievements • Check conditions	• MainApplication • QuizMode

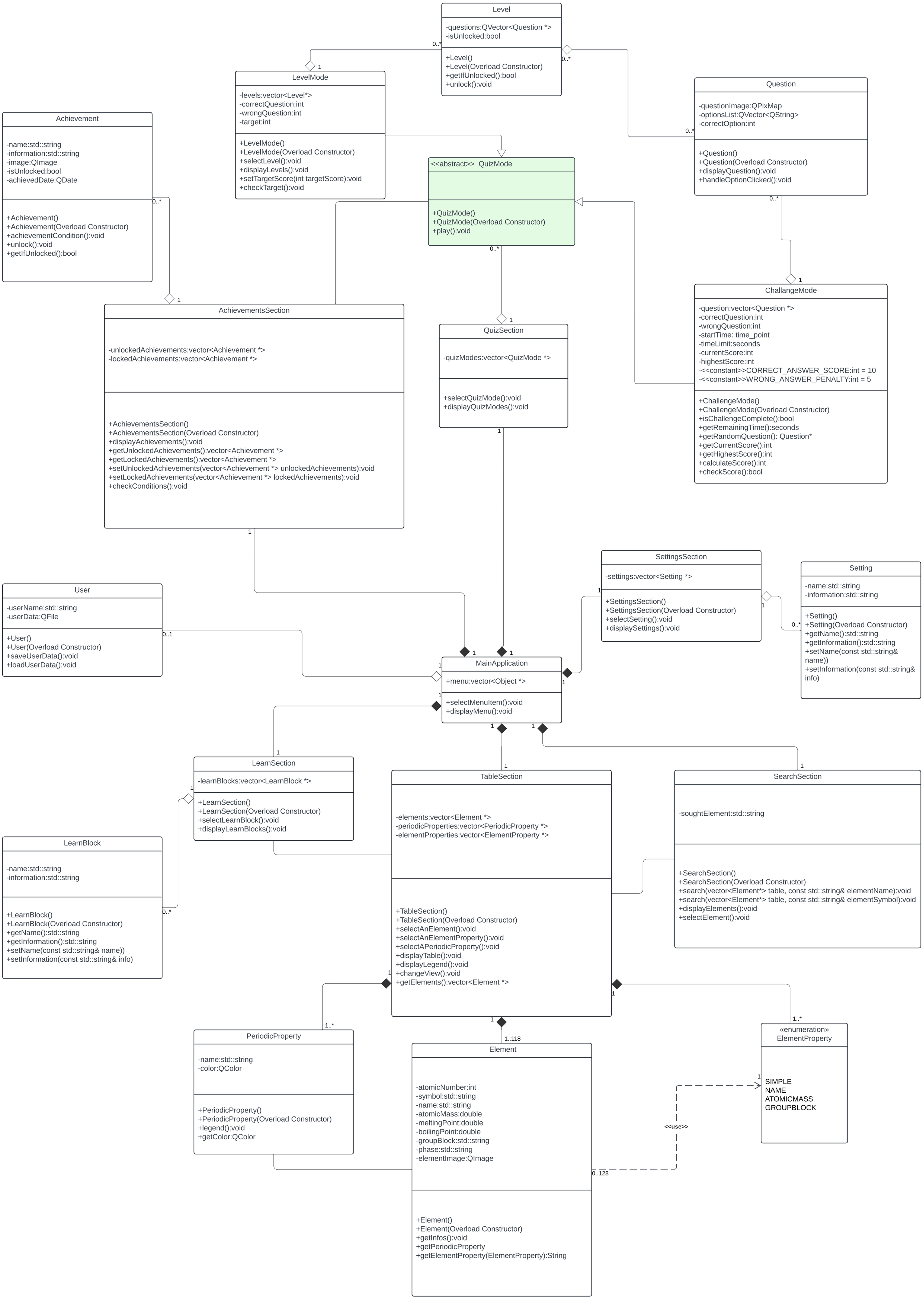
Achievement	
• Get achievement informations • Generate achievement • Check the achievement situation	• AchievementsSection

SettingsSection	
• Display settings • Select setting	• MainApplication • Setting

Setting	
• Get setting informations • Generate setting	• SettingsSection

User	
• Generate user • Save user data • Load user data	• MainApplication

4.Class Diagram


Level
-questions:QVector<Question *>
-isUnlocked:bool
+Level()
+Level(Overload Constructor)
+getIfUnlocked():bool
+unlock():void
0..*
1
0..*
LevelMode
-levels:vector<Level*>
-correctQuestion:int
-wrongQuestion:int
-target:int
+LevelMode()
+LevelMode(Overload Constructor)
+selectLevel():void
+displayLevels():void
+setTargetScore(int targetScore):void
+checkTarget():void
Question
-questionImage:QPixMap
-optionsList:QVector<QString>
-correctOption:int
+Question()
+Question(Overload Constructor)
+displayQuestion():void
+handleOptionClicked():void
0..*
Achievement
-name:std::string
-information:std::string
-image:QImage
-isUnlocked:bool
-achievedDate:QDate
+Achievement()
+Achievement(Overload Constructor)
+achievementCondition():void
+unlock():void
+getIfUnlocked():bool
0..*
<<abstract>> QuizMode
+QuizMode()
+QuizMode(Overload Constructor)
+play():void
0..*
1
ChallangeMode
-question:vector<Question *>
-correctQuestion:int
-wrongQuestion:int
-startTime: time_point
-timeLimit:seconds
-currentScore:int
-highestScore:int
-<<constant>>CORRECT_ANSWER_SCORE:int = 10
-<<constant>>WRONG_ANSWER_PENALTY:int = 5
+ChallengeMode()
+ChallengeMode(Overload Constructor)
+isChallengeComplete():bool
+getRemainingTime():seconds
+getRandomQuestion(): Question*
+getCurrentScore():int
+getHighestScore():int
+calculateScore():int
+checkScore():bool
1
AchievementsSection
-unlockedAchievements:vector<Achievement *>
-lockedAchievements:vector<Achievement *>
+AchievementsSection()
+AchievementsSection(Overload Constructor)
+displayAchievements():void
+getUnlockedAchievements():vector<Achievement *>
+getLockedAchievements():vector<Achievement *>
+setUnlockedAchievements(vector<Achievement *> unlockedAchievements):void
+setLockedAchievements(vector<Achievement *> lockedAchievements):void
+checkConditions():void
1
QuizSection
-quizModes:vector<QuizMode *>
+selectQuizMode():void
+displayQuizModes():void
1
SettingsSection
-settings:vector<Setting *>
+SettingsSection()
+SettingsSection(Overload Constructor)
+selectSetting():void
+displaySettings():void
1
1
Setting
-name:std::string
-information:std::string
+Setting()
+Setting(Overload Constructor)
+getName():std::string
+getInformation():std::string
+setName(const std::string& name))
+setInformation(const std::string& info)
0..*
User
-userName:std::string
-userData:QFile
+User()
+User(Overload Constructor)
+saveUserData():void
+loadUserData():void
0..1
1
1
1
MainApplication
+menu:vector<Object *>
+selectMenuItem():void
+displayMenu():void
1
1
1
1
1
LearnSection
-learnBlocks:vector<LearnBlock *>
+LearnSection()
+LearnSection(Overload Constructor)
+selectLearnBlock():void
+displayLearnBlocks():void
1
1
TableSection
-elements:vector<Element *>
-periodicProperties:vector<PeriodicProperty *>
-elementProperties:vector<ElementProperty *>
+TableSection()
+TableSection(Overload Constructor)
+selectAnElement():void
+selectAnElementProperty():void
+selectAPeriodicProperty():void
+displayTable():void
+displayLegend():void
+changeView():void
+getElements():vector<Element *>
1
1
1
SearchSection
-soughtElement:std::string
+SearchSection()
+SearchSection(Overload Constructor)
+search(vector<Element*> table, const std::string& elementName):void
+search(vector<Element*> table, const std::string& elementSymbol):void
+displayElements():void
+selectElement():void
1
LearnBlock
-name:std::string
-information:std::string
+LearnBlock()
+LearnBlock(Overload Constructor)
+getName():std::string
+getInformation():std::string
+setName(const std::string& name))
+setInformation(const std::string& info)
0..*
1..*
1..*
PeriodicProperty
-name:std::string
-color:QColor
+PeriodicProperty()
+PeriodicProperty(Overload Constructor)
+legend():void
+getColor:QColor
1
1..118
Element
-atomicNumber:int
-symbol:std::string
-name:std::string
-atomicMass:double
-meltingPoint:double
-boilingPoint:double
-groupBlock:std::string
-phase:std::string
-elementImage:QImage
+Element()
+Element(Overload Constructor)
+getInfos():void
+getPeriodicProperty
+getElementProperty(ElementProperty):String
<<use>>
0..128
1
«enumeration»
ElementProperty
SIMPLE
NAME
ATOMICMASS
GROUPBLOCK

4.Conclusion

At this stage of the Project, the design started to be outlined. While designing CRC card and class diagrams, we had difficulties because we don't know enough GUI yet, so there may be some mistakes. For the next phase we focus on learning Qt.

We all thought together about the implementation of the application, CRC cards and some of the UML diagrams were prepared by the group members together. Afterwards, Mehmet Emre Cebeci and Ahmet Kaan Demirci took the responsibility of detailing the diagrams.