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

Table of Contents

1. Introduction.....2
2. Class-responsibility-collaboration (CRC) cards 2
3. Class Diagram 3
4. Conclusion.....3

1. Introduction

The PeriodiCpp application serves as a comprehensive tool for exploring the periodic table in detail, offering users the opportunity to delve into various aspects and engage in entertaining quiz modes. This report encompasses CRC (Class Responsibility Collaborator) cards and a class diagram. The CRC cards systematically outline potential classes, elucidating their respective responsibilities and collaborators. The class diagram further delineates the attributes and methods associated with each class identified earlier.

2. 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 information • 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

QuizMode ChallengeMode	
• 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

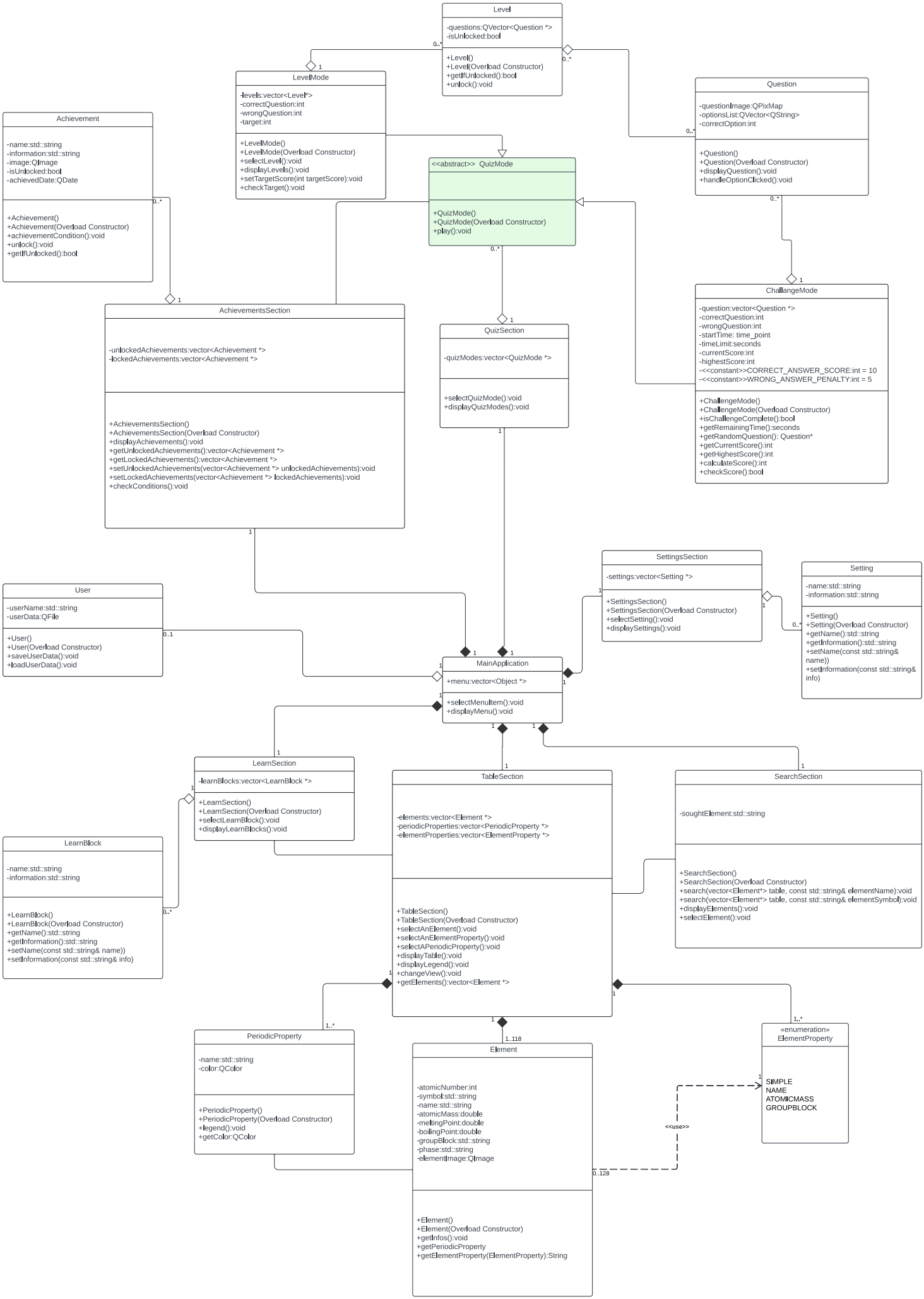
Achievement	
• Get achievement information • Generate achievement • Check the achievement situation	• AchievementsSection

SettingsSection	
• Display settings • Select setting	• MainApplication • Setting

Setting	
• Get setting information • Generate setting	• SettingsSection

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

3. Class Diagram


Level
-questions:QVector<Question *>
-isUnlocked:bool
+Level()
+Level(Overload Constructor)
+getIfUnlocked():bool
+unlock():void
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
Achievement
-name:std::string
-information:std::string
-image:QImage
-isUnlocked:bool
-achievedDate:QDate
+Achievement()
+Achievement(Overload Constructor)
+achievementCondition():void
+unlock():void
+getIfUnlocked():bool
<<abstract>> QuizMode
+QuizMode()
+QuizMode(Overload Constructor)
+play():void
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
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
QuizSection
-quizModes:vector<QuizMode *>
+selectQuizMode():void
+displayQuizModes():void
SettingsSection
-settings:vector<Setting *>
+SettingsSection()
+SettingsSection(Overload Constructor)
+selectSetting():void
+displaySettings():void
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)
User
-userName:std::string
-userData:QFile
+User()
+User(Overload Constructor)
+saveUserData():void
+loadUserData():void
MainApplication
+menu:vector<Object *>
+selectMenuItem():void
+displayMenu():void
LearnSection
-learnBlocks:vector<LearnBlock *>
+LearnSection()
+LearnSection(Overload Constructor)
+selectLearnBlock():void
+displayLearnBlocks():void
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 *>
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
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)
PeriodicProperty
-name:std::string
-color:QColor
+PeriodicProperty()
+PeriodicProperty(Overload Constructor)
+legend():void
+getColor:QColor
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
«enumeration»
ElementProperty
SIMPLE
NAME
ATOMICMASS
GROUPBLOCK
<<use>>

4. Conclusion

At this stage of the project, we started to sketch the design. We had difficulties designing the CRC card and class diagrams because we don't know enough about GUI yet, so there might 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. After that, Mehmet Emre Cebeci and Ahmet Kaan Demirci took the responsibility of detailing the diagrams.