

**Assessment 2**

**Lab 2**

Author: Bertocco Thomas-Killian - #13758322

Lecturer: Da Sheng Liu

CO5609 Intelligent Agents and Simulation | SP53 | 2019

5 January 2020

Table of Contents

[Changelog 4](#_Toc29135183)

[All files 4](#_Toc29135184)

[VladPackManGame.cpp 4](#_Toc29135185)

[main() 4](#_Toc29135186)

[BoardH.h 4](#_Toc29135187)

[Board() 4](#_Toc29135188)

[setBoardOk() 5](#_Toc29135189)

[IamAlive() 5](#_Toc29135190)

[Board.ccp 5](#_Toc29135191)

[touch\_all() 5](#_Toc29135192)

[game\_all() 5](#_Toc29135193)

[move() 5](#_Toc29135194)

[paint() 5](#_Toc29135195)

[ListBoardObjects.cpp 6](#_Toc29135196)

[List() 6](#_Toc29135197)

[the first() 6](#_Toc29135198)

[theNext() 6](#_Toc29135199)

[addOne() 6](#_Toc29135200)

[isCookieValue() 6](#_Toc29135201)

[BoardObjectH.h 6](#_Toc29135202)

[boardObject() 6](#_Toc29135203)

[Touch() 6](#_Toc29135204)

[set\_color() 7](#_Toc29135205)

[get\_Type() 7](#_Toc29135206)

[get\_Color() 7](#_Toc29135207)

[get\_x() 7](#_Toc29135208)

[get\_y() 7](#_Toc29135209)

[set\_x() 7](#_Toc29135210)

[set\_y() 7](#_Toc29135211)

[CookieH.h 7](#_Toc29135212)

[Cookie() 7](#_Toc29135213)

[Cookie.cpp 7](#_Toc29135214)

[Touch() 7](#_Toc29135215)

[get\_value() 7](#_Toc29135216)

[PlayerH.h 8](#_Toc29135217)

[Player() 8](#_Toc29135218)

[Player.cpp 8](#_Toc29135219)

[Touch() 8](#_Toc29135220)

[WallH.h 8](#_Toc29135221)

[Wall() 8](#_Toc29135222)

[Review Façade and Iterator patterns 9](#_Toc29135223)

[Facade design pattern 9](#_Toc29135224)

[Iterator pattern 9](#_Toc29135225)

[Class diagram 10](#_Toc29135226)

[Sequence diagram 11](#_Toc29135227)

# Changelog

*To see all the changes in the code, please visit my Github repository: https://github.com/Yunori/Lab02*

*The first commit is the original code, and the second one is my modifications.*

# All files

* Corrected indentation and spaces, code style
* Corrected many bugs… because the structure sucks 😊

# VladPackManGame.cpp

## main()

* Changed the control mapping to "wasd" keys
* Changed the check of user input
* Changed the input method to "\_getch" which doesn't use a buffer, and prevent user from typing multiple characters
* Changed the "aMove" variable name to "userInput"

# BoardH.h

* Renamed the alias "p\_ListBoardObjects" to "PointerListBoardObjects"
* Initialized all variables to fix warning
* changed the "player\_id\_X" variable name to "\_playerIdX"
* changed the "player\_id\_y" variable name to "\_playerIdY"
* changed the "the\_rows" variable name to "\_rowsAmount"
* changed the "the\_cols" variable name to "\_columnsAmount"
* changed the "gameOn" variable name to "\_gameStarted"
* Changed the "mySpace" variable name to "\_objectsList"
* changed the "isOk" variable name to "\_isMapSizeOk"

## Board()

* Changed the parameter name "rows" to "rowsAmount"
* Changed the parameter name "cols" to "columnsAmount"
* Passed the "rows" parameter by reference ("&rows")
* changed the "rows" parameter type from "int" to "const int"
* Passed the "cols" parameter by reference ("&cols")
* changed the "cols" parameter type from "int" to "const int"
* Replaced "setOK - true;" by "setOK = true;"
* Merge the "add wall" and "add cookie" loop
* Added a verification of the "the\_cols" variable size to fix a warning
* changed the "aBackground" variable name to "tileBackground"
* Removed the useless "p\_ListBoardObjects aList = new ListBoardObjects(tileBackground);" line, and assigned it directly to map[i][j]
* changed the "aWall" variable name to "newWall"
* changed the "aCookie" variable name to "newCookie"

## setBoardOk()

* Changed the setter to a getter (getBoardOk), because I think it's a getter (only return a value)
* changed the return type from "bool" to "const bool"

## IamAlive()

* Changed the method name to isPlayerAlive
* changed the return type from "bool" to "const bool"

# Board.ccp

## touch\_all()

* Changed the method name to "changeColorAllObjects"
* renamed the "aList" parameter to "objectsList"
* Passed the "objectsList" parameter by reference ("&objectsList")
* changed the "objectsList" parameter type from "p\_ListBoardObjects" to "const p\_ListBoardObjects"
* changed the "toIterate" variable name to "currentObject"

## game\_all()

* Changed the method name to convertObjectToLetter
* Rewrote this stupid function (takes a list of objects but can only return 1 char...)
* renamed the "aList" parameter to "objectsList"
* Passed the "objectsList" parameter by reference ("&objectsList")
* changed the "objectsList" parameter type from "p\_ListBoardObjects" to "const p\_ListBoardObjects"
* Added a return before the end of the method to fix a warning
* change the return type from "char" to "const char"
* changed the "toIterate" variable name to "currentObject"

## move()

* Totally rewrote the method because it was disgusting
* Passed the "command" parameter by reference ("&command")
* changed the "command" parameter type from "char" to "const char"

## paint()

* renamed to render()

# ListBoardObjects.cpp

Renamed the alias "p\_BoardObject" to "PointerBoardObject"

## List()

* changed the "newOne" parameter type from "p\_BoardObject" to "const p\_BoardObject"
* Changed the "newOne" variable name to "newObject"
* Passed the "newObject" parameter by reference ("&newObject")

## the first()

* Changed the method name to firstObject
* Now return a constant

## theNext()

* Changed the method name to nextObject
* Now return a constant

## addOne()

* changed the "newOne" parameter type from "p\_BoardObject" to "const p\_BoardObject"
* Changed the method name to addObject
* Changed the "newOne" variable name to "newObject"

## isCookieValue()

* Changed the method name to getCookieValue
* changed the "temp" variable name to "cookieValue"

ListBoardObjectsH.h

* Prepend "\_" to the attributes name.

# BoardObjectH.h

* Set all the attributes to private
* Prepend "\_" to the attributes name
* Changed enum ObjectColours to enum class to fix a warning
* Changed enum ObjectTypes to enum class to fix a warning
* Changed the "myClass" variable name to "objectType"
* Changed the values of the ObjectTypes enum class: BACKGROUND, COOKIE, PLAYER, WALL

## boardObject()

* Changed the name of parameters start\_x and start\_y to startX and startY

## Touch()

* Changed the name of the method "touch()" to "changeColor()"

## set\_color()

* Changed the method name to setColor
* Changed the name of the parameter "new\_color" to "newColor"
* Passed the "newColor" parameter by reference ("&newColor")

## get\_Type()

* Changed the method name to getType
* Now return a constant reference

## get\_Color()

* Changed the method name to getColor
* Now return a constant reference

## get\_x()

* Changed the method name to getX

## get\_y()

* Changed the method name to getY

## set\_x()

* Changed the method name to setX
* Changed the name of the parameter "new\_x" to "newX"

## set\_y()

* Changed the method name to setY
* Changed the name of the parameter "new\_y" to "newY"

# CookieH.h

* Prepend "\_" to the attributes name.
* Changed enum CookieState to enum class to fix a warning
* Changed the "PrizeValueIndex" variable name to "CookieValueIndex"
* Changed enum PrizeValueIndex to enum class to fix a warning

## Cookie()

* Changed the name of parameters start\_x and start\_y to startX and startY

# Cookie.cpp

* Changed the "PrizeValues" variable name to "CookieValues"

## Touch()

* Changed the name of the method name to "changeColor()"

## get\_value()

* Changed the method name to getValue

# PlayerH.h

* Prepend "\_" to the attributes name.
* Changed enum PlayerState to enum class to fix a warning

## Player()

* Changed the name of parameters start\_x and start\_y to startX and startY

Touch()

Changed the type of the method from "virtual void" to "void"

# Player.cpp

## Touch()

* Changed the name of the method name to "changeColor()"

# WallH.h

Changed the “color1” attribute name to "\_objectColor"

## Wall()

* Changed the name of parameters start\_x and start\_y to startX and startY

# Wall.cpp

## Touch()

* Changed the name of the method name to "changeColor()"

# Review Façade and Iterator patterns

## Facade design pattern

The principle of the façade pattern is to create an interface between the user and the code, that will hide the code complexity.

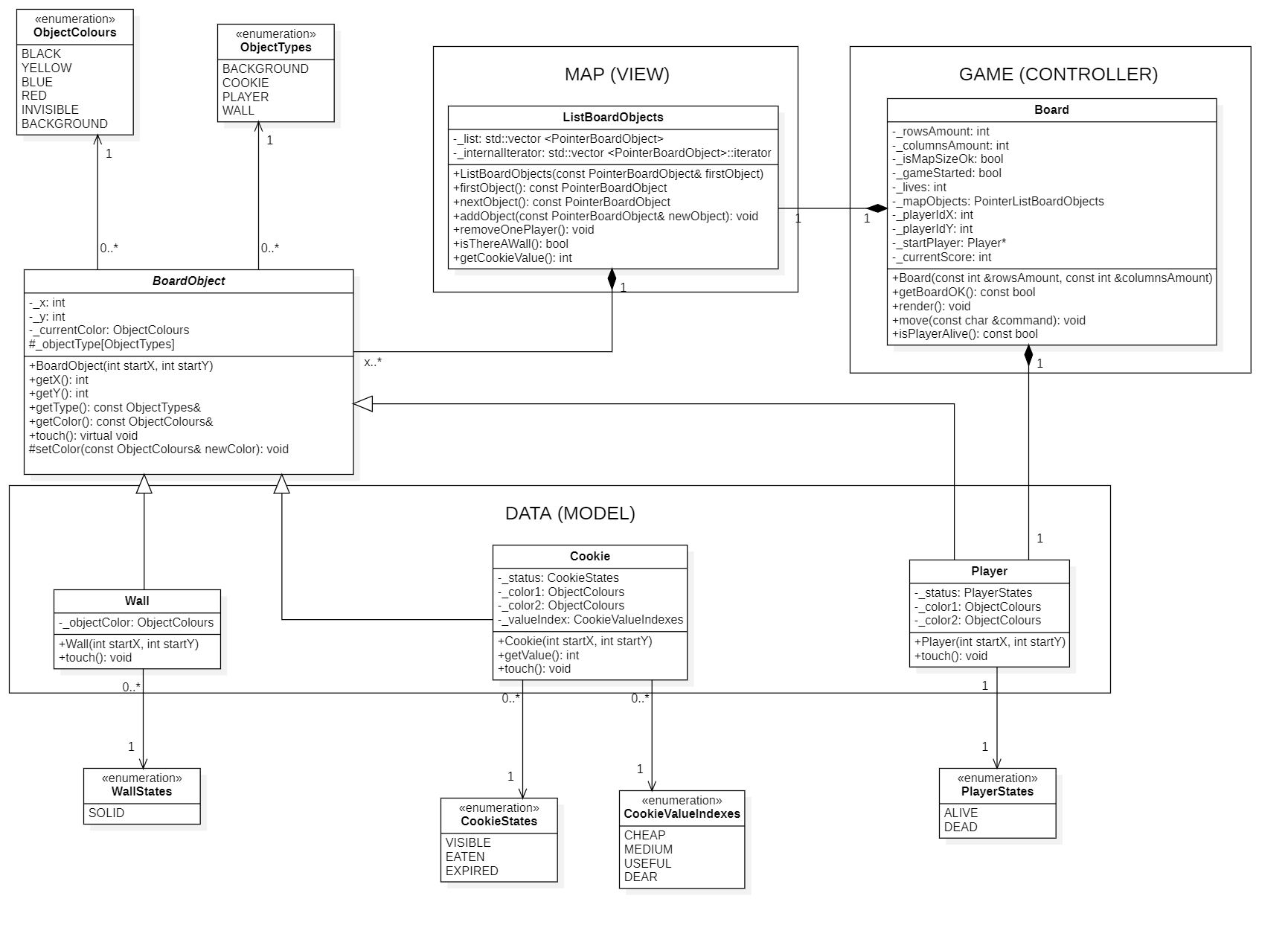
In our case, the class "ListBoardObject" is a façade because it regroups every objects of the map, from different classes (cookies, player, empty zones, walls). It can be considered as a view, a map.

## Iterator pattern

The iterator pattern consists of creating an iterator that will allows us to review/modify every object in a data structure.

In our case, we use an iterator on the ListBoardObject class to review every objects of the map, changed their color and look at their type to display the correct letter on the map.

# Class diagram



# Sequence diagram

