2018

Documentation for the Breakout project

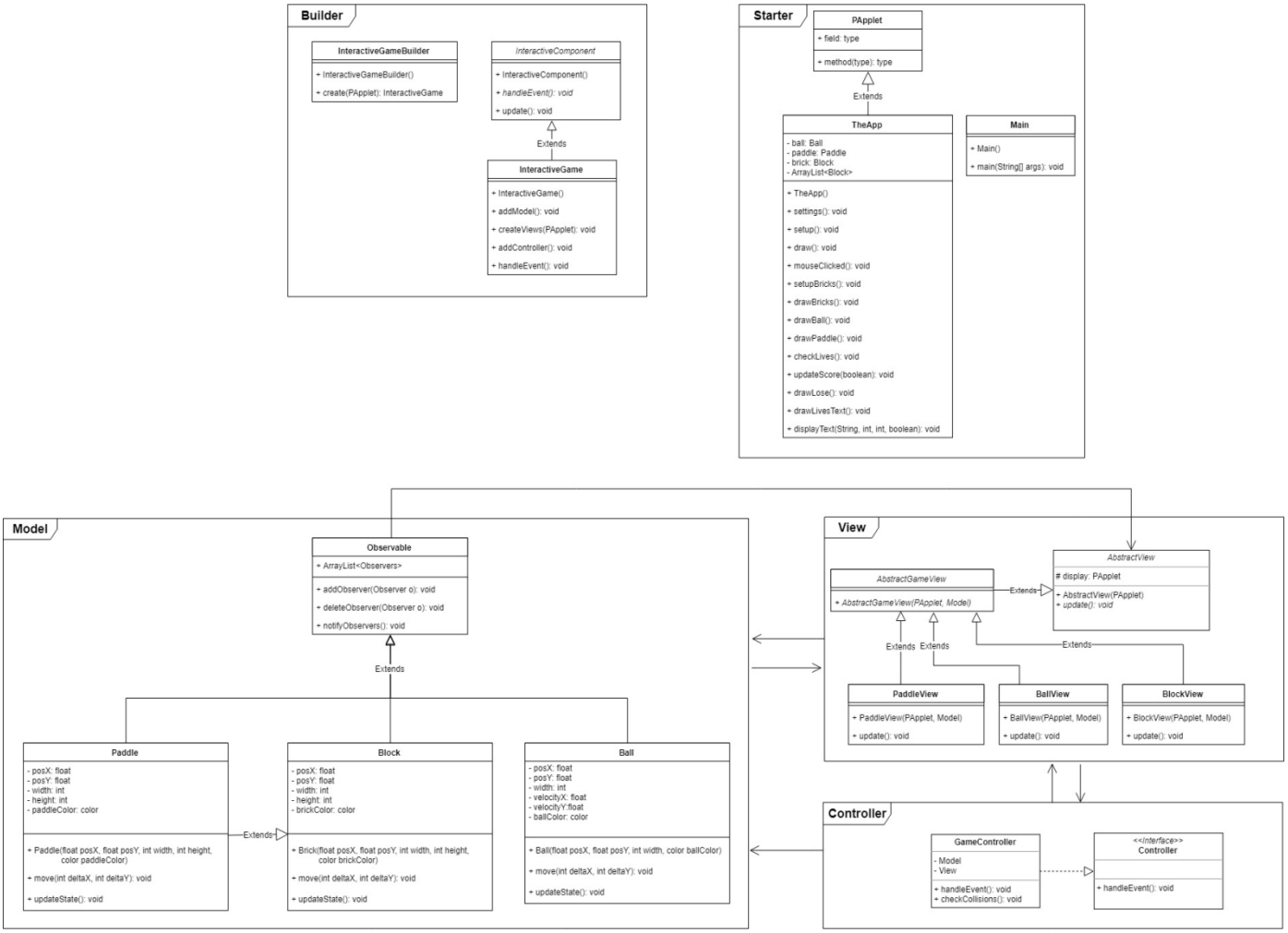
Team #42

The Breakout game is developed within the Java Capstone Series Pt. 1 course, held by openHPI in September, 2018, and is implemented in accordance with the given task.

Table of contents

UML diagram of the project

This is the UML class diagram of the project (picture 1). It shows the interfaces, abstract and concrete classes used in the project, as well as the relationships between them. The project was developed in accordance with the MVC architectural pattern.



Picture 1. UML class diagram.

Game description and gameplay

High-level description

In *Breakout*, a layer of bricks lines the top third of the screen. A ball travels across the screen, bouncing off the top and side walls of the screen. When a brick is hit, the ball bounces away and the brick is destroyed. The player loses a turn when the ball touches the bottom of the screen. To prevent this from happening, the player has a movable paddle to bounce the ball upward, keeping it in play[1].

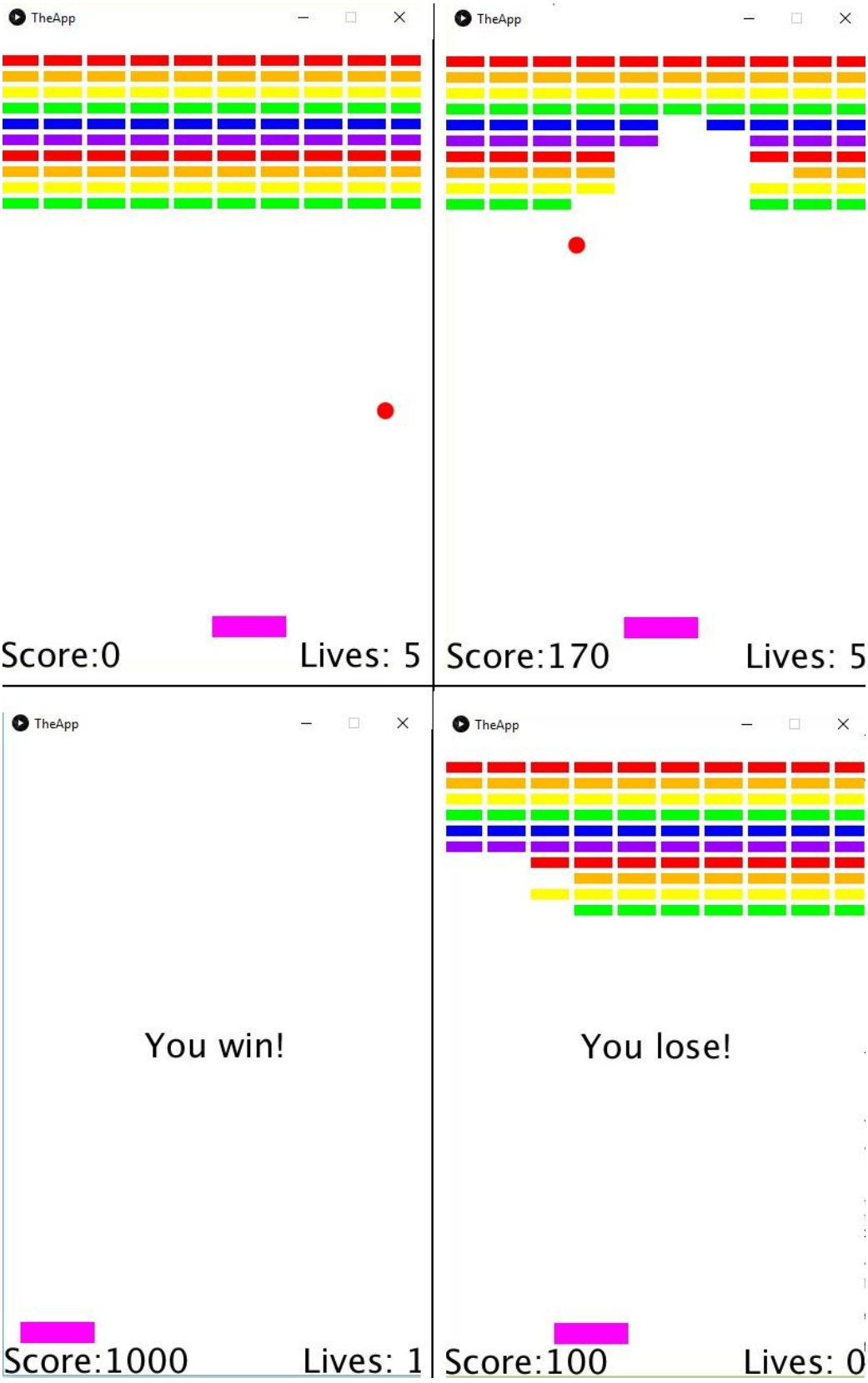
[1] https://en.wikipedia.org/wiki/Breakout_(video_game)

Game elements and controls

Table 1

Element	Description	Keys
Ball	Once inserted to the playground, it moves to a certain set of rules: ▪ Ball bounces (changes its direction) when it hits the left, right or upper bound of the playground. ▪ Ball bounces when it hits the paddle. ▪ Ball leaves the playground when it "hits" lower bound of the playground. ▪ A new Ball is automatically inserted when the previous Ball has left the playground.	—
Paddle	The game element which is used to hit the ball. The Paddle can move to the right and to the left within the game window. The movement of the Paddle is controlled by the player.	Left arrow moves the Paddle to the left. Right arrow moves the Paddle to the right.
Wall of Bricks	The Wall of Bricks contains 10 rows of 10 Bricks in each. When the player destroys all bricks, he wins and gets the message “You win!”	—
Lives	At the beginning of the game the player has 5 lives. Each time the Ball leaves the playground, the player loses 1 life. When the player loses all lives, he loses the game and gets the message “You lose!”	The number of lives decreases automatically to 0.
Score	The player receives 10 points for each destroyed Brick. The maximum number of points is 1000. Points are counted throughout the game until the player loses or wins.	The score counts automatically.

Screenshots

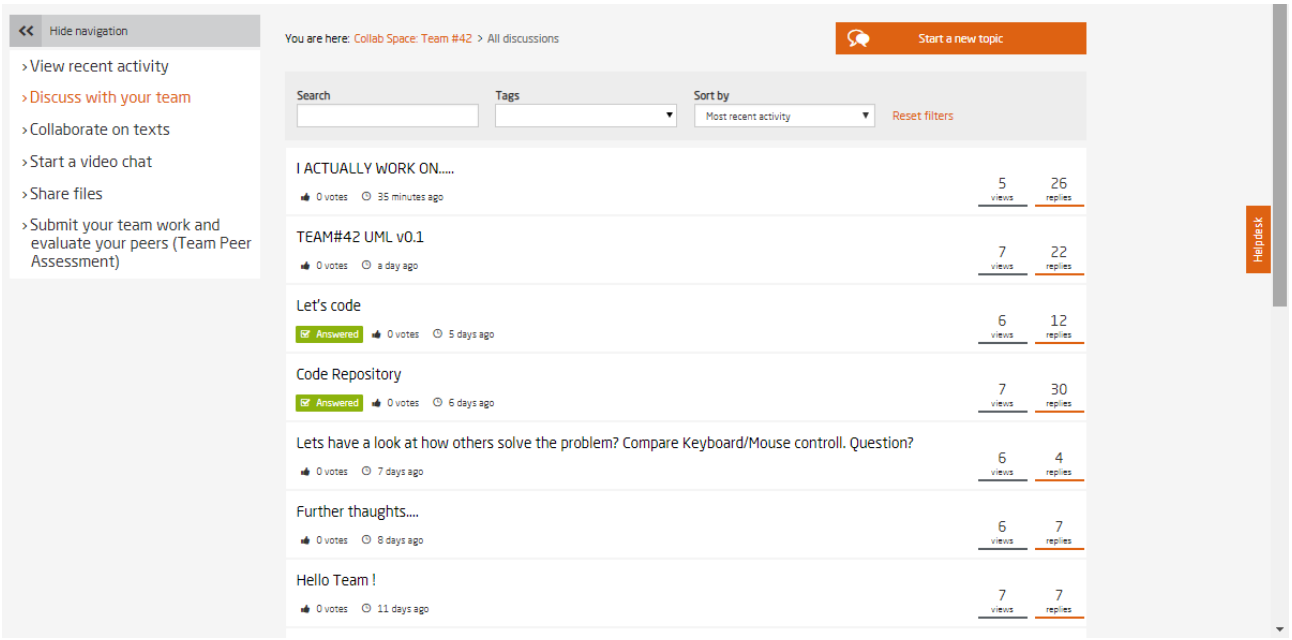


Picture 2. Screenshots of gameplay

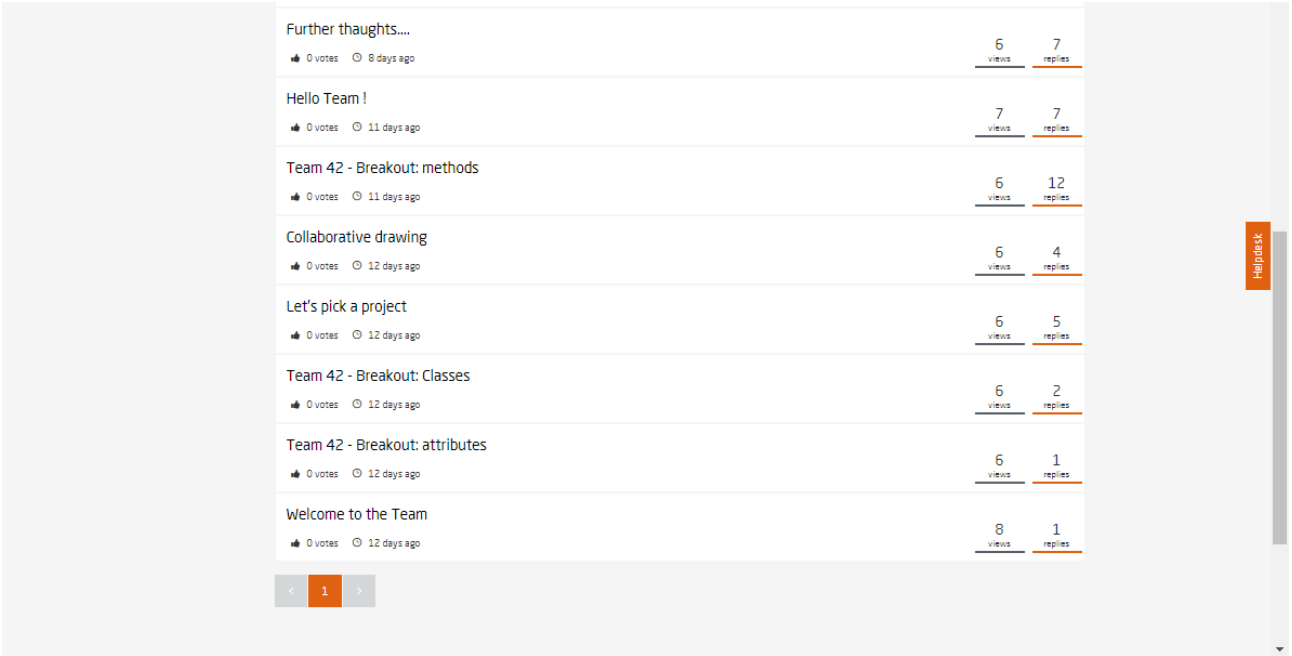
Lab Report

In our team work began from the very first day. Almost immediately, 7 out of 8 team members got in touch, but only 3 eventually started to work directly on the project.

Throughout the course, we actively communicated at the forum in all 13 topics that we've created (pict. 3 - 4).



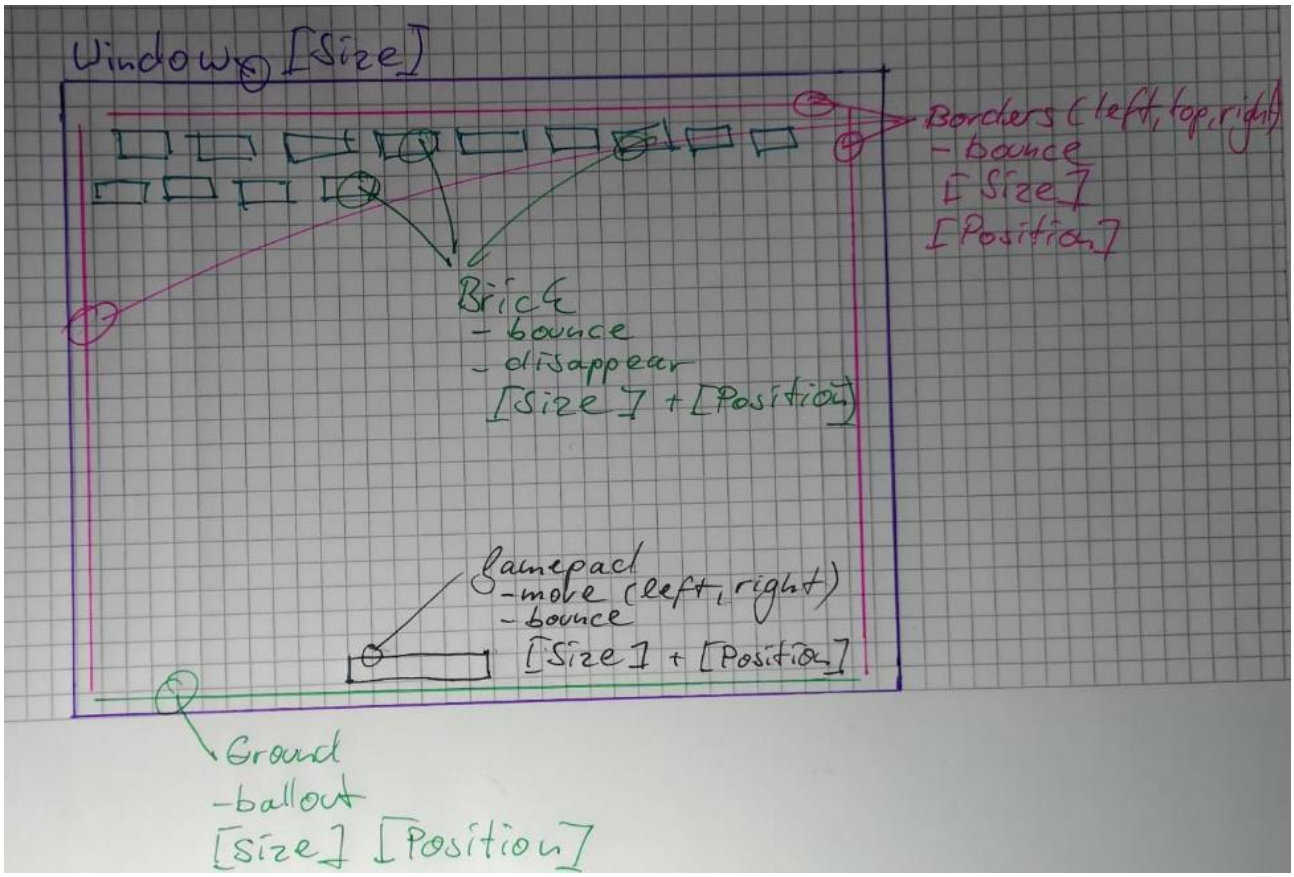
Picture 3. Team topics-1.



Picture 4. Team topics-2.

During the initial discussions, the following was done:

1. We selected a tool for drawing the UML class diagram, and it was Draw.io.
2. We forked Tom's repository with the starter code.
3. We sketched out a list of necessary classes, attributes and methods.
4. We made a sketch of the game field (pict. 5).



Picture 5. Initial sketch of the game field.

We argued a lot about the ways of collision detection, as well as how to implement a Wall of Bricks. In this regard, there were 2 main positions:

1. create a 2d array [row, column] of Brick-Objects. The bricks (rect) themselves have attributes: x = start position x, y = startposition y, h = height, w = width.
2. create a 1d array where the two-dimensional location of the bricks is defined by the coordinates stored in the objects.

As a result, the first option was chosen.

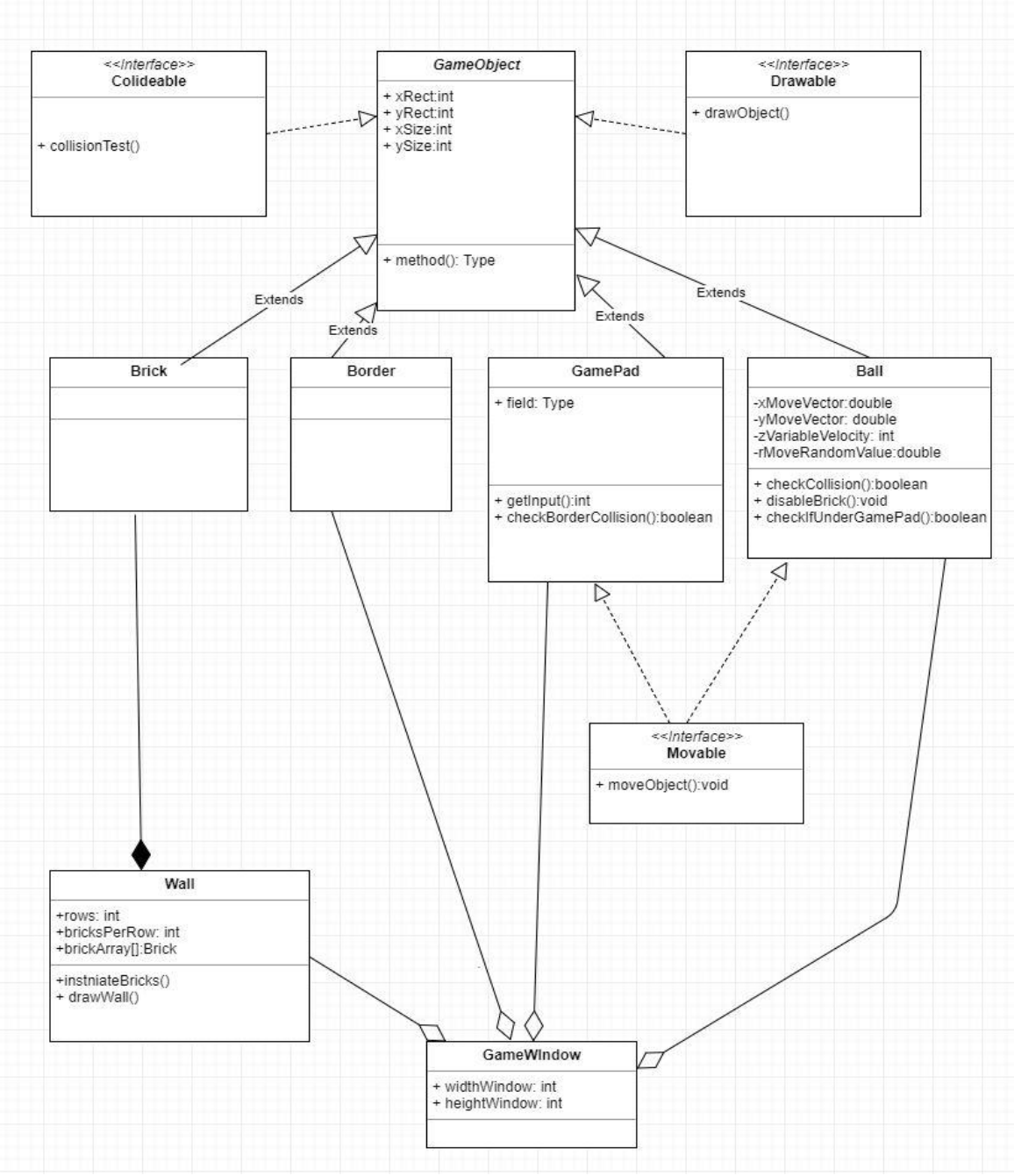
Also, we did not immediately manage to agree on the choice of controlling the paddle between the keyboard and the mouse, but eventually chose control from the keyboard.

Due to the fact that not all the members of the team were very familiar with the Git and the patterns, after a few days the work stopped - everyone was immersed in the study.

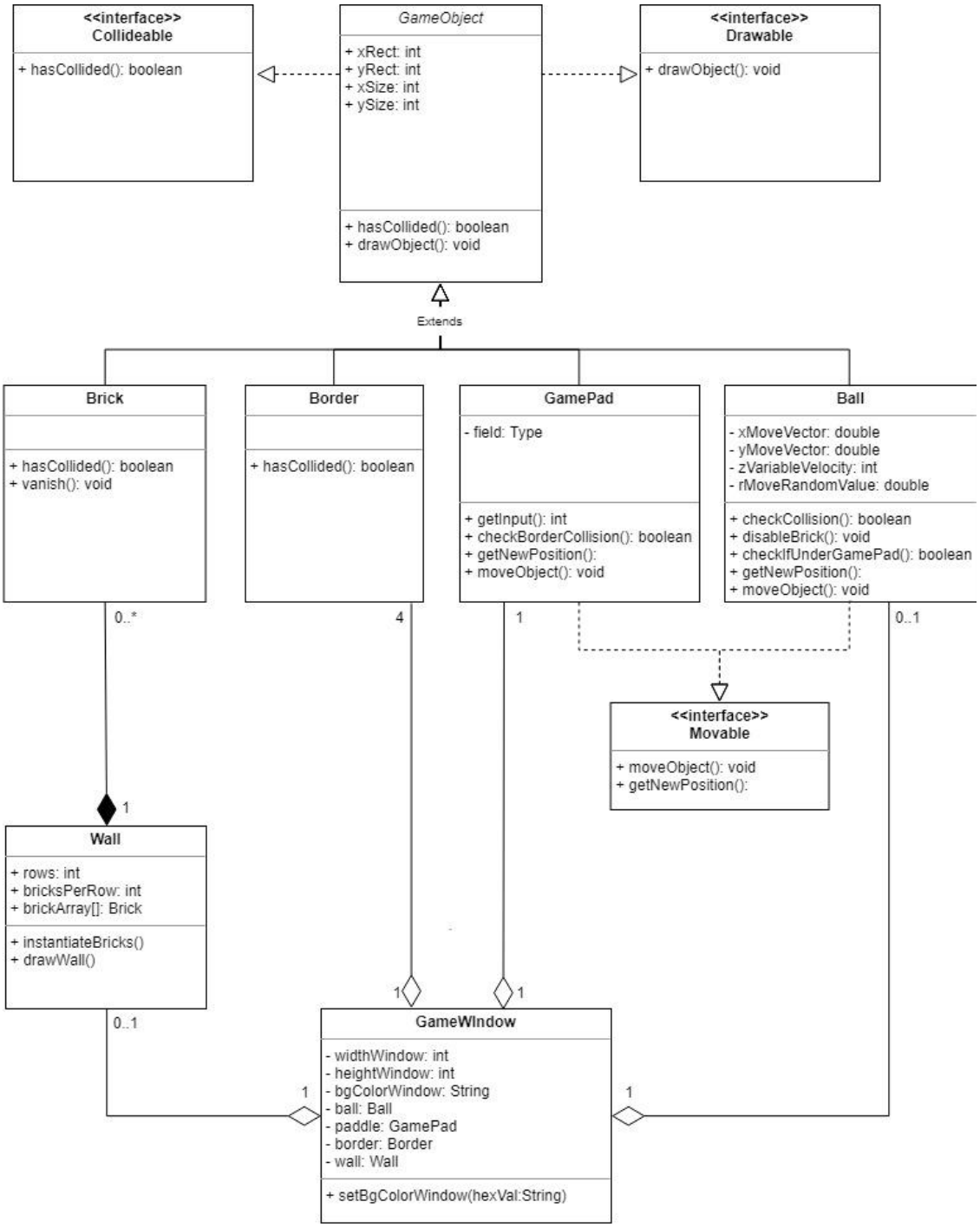
By the time the work was resumed and we had returned to the creation of the UML diagram, almost half the project time had passed.

Working on the diagram took some of the remaining time.

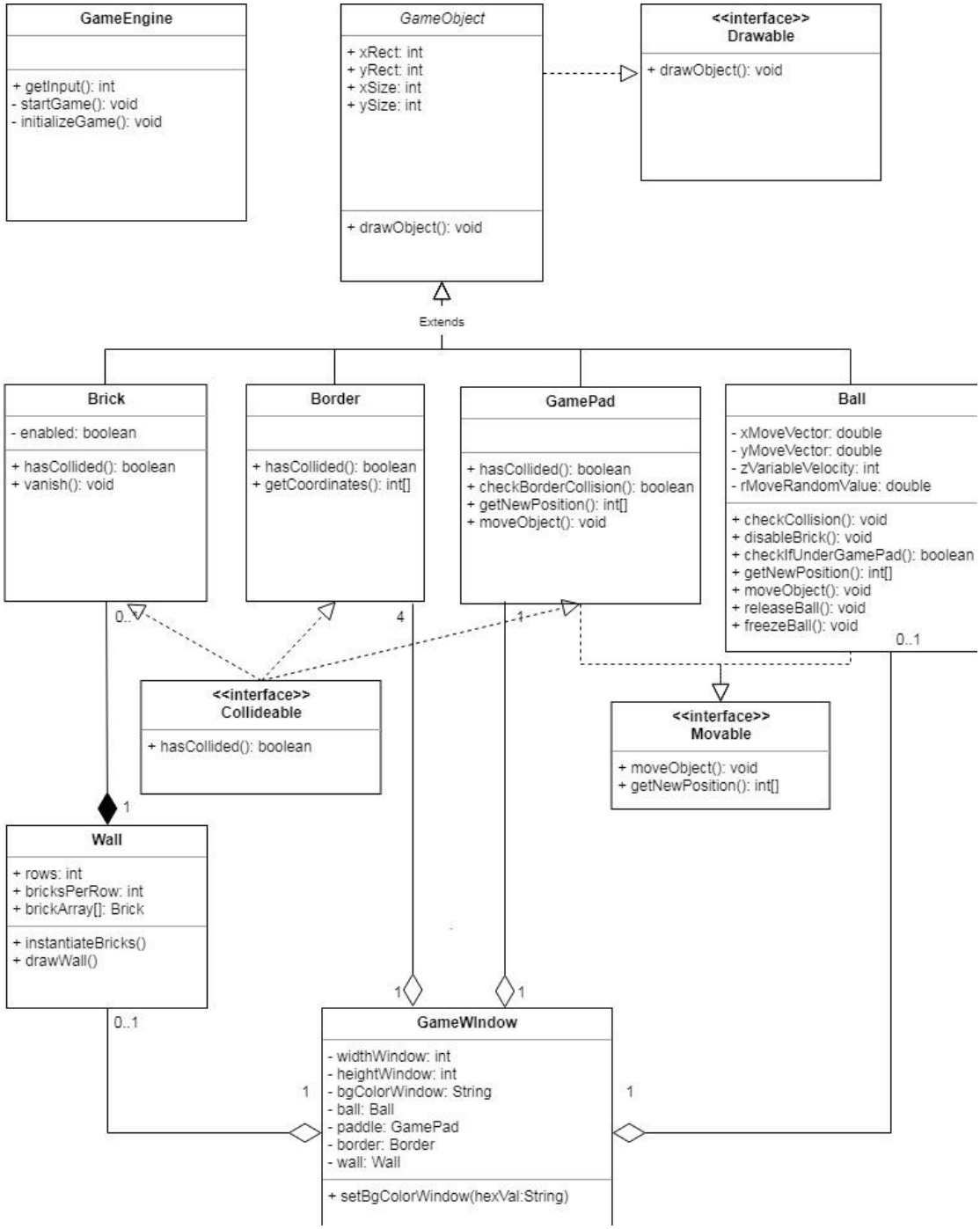
5 variants of the diagram were prepared. Its evolution is reflected in pictures 6 - 10.



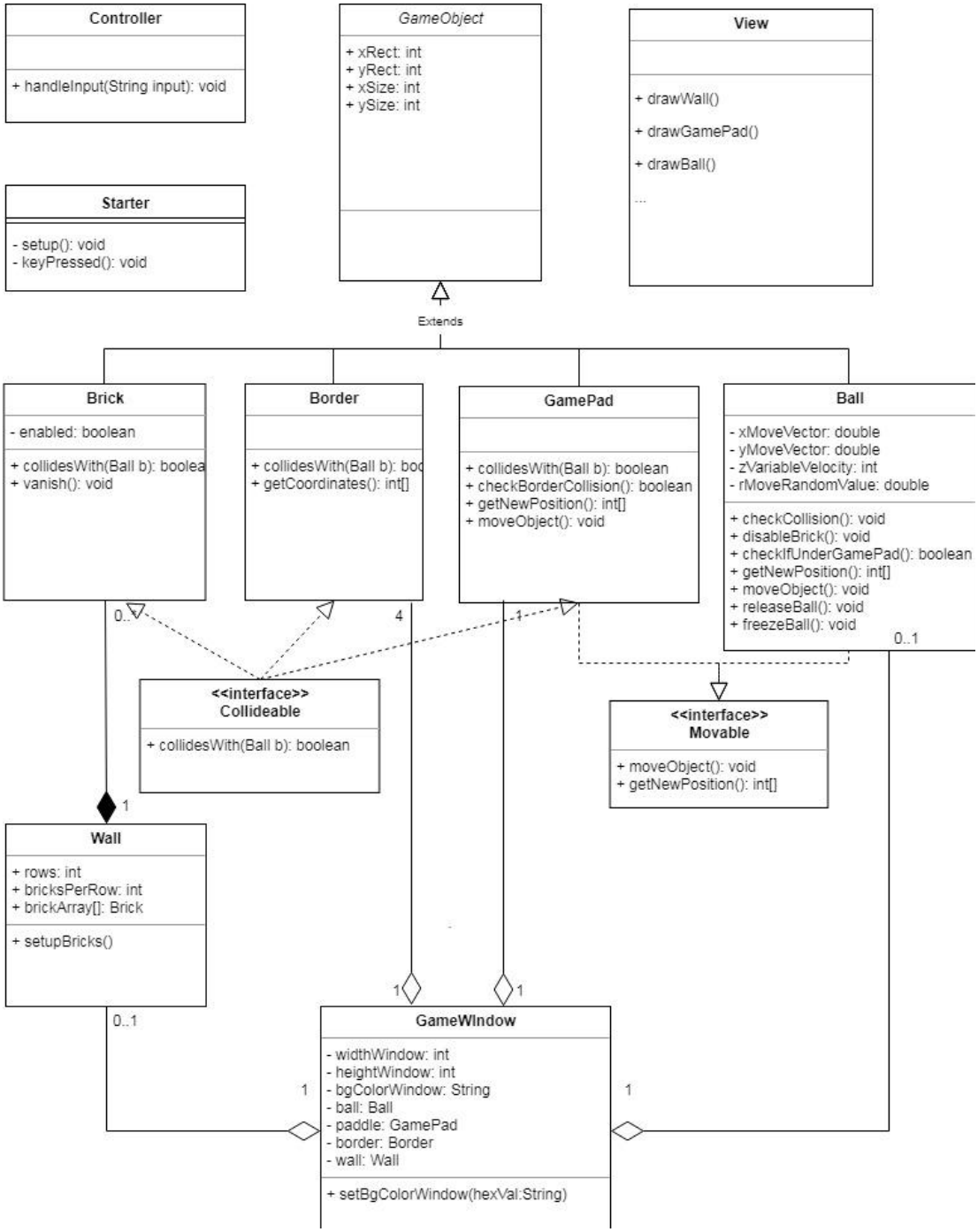
Picture 6. UML diagram - v1.



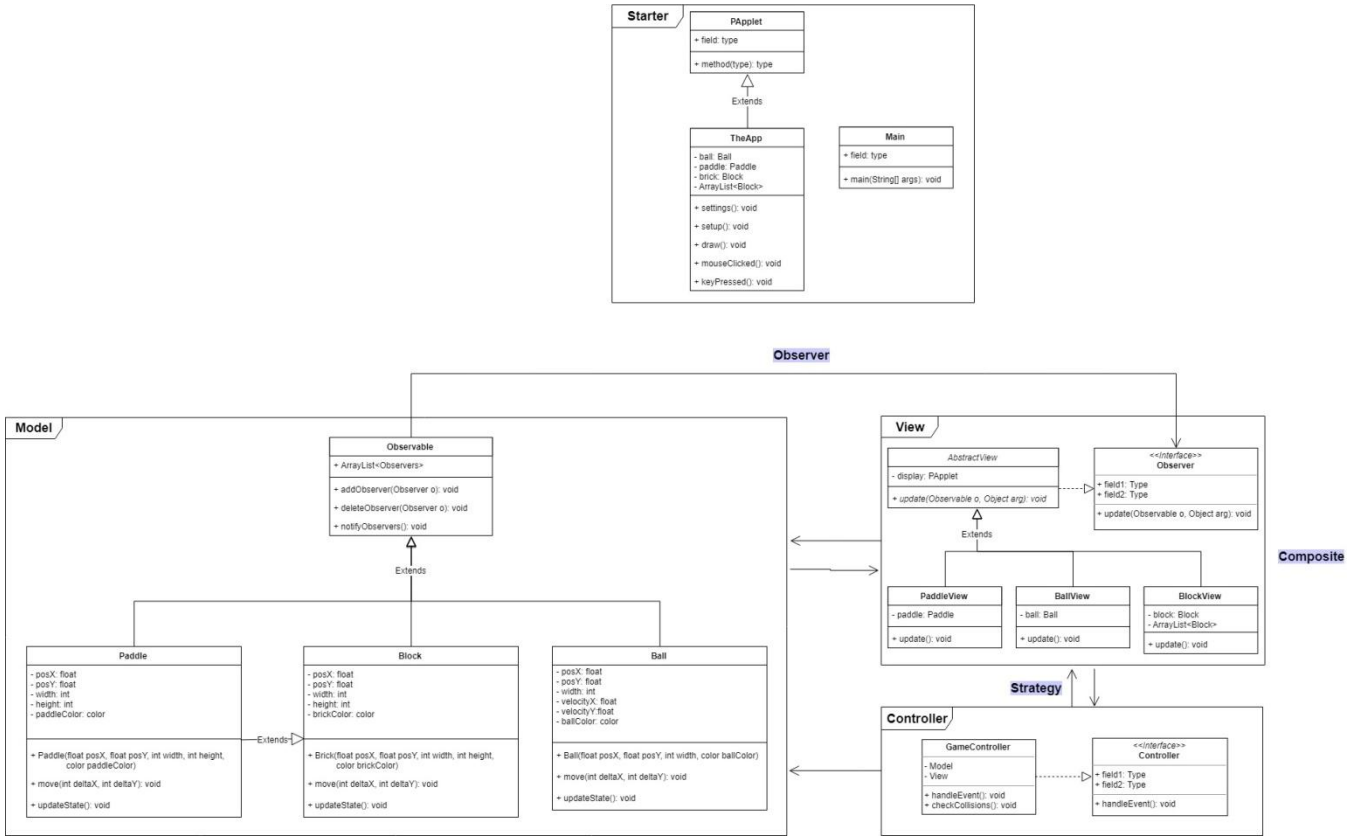
Picture 7. UML diagram – v2.



Picture 8. UML diagram – v3.



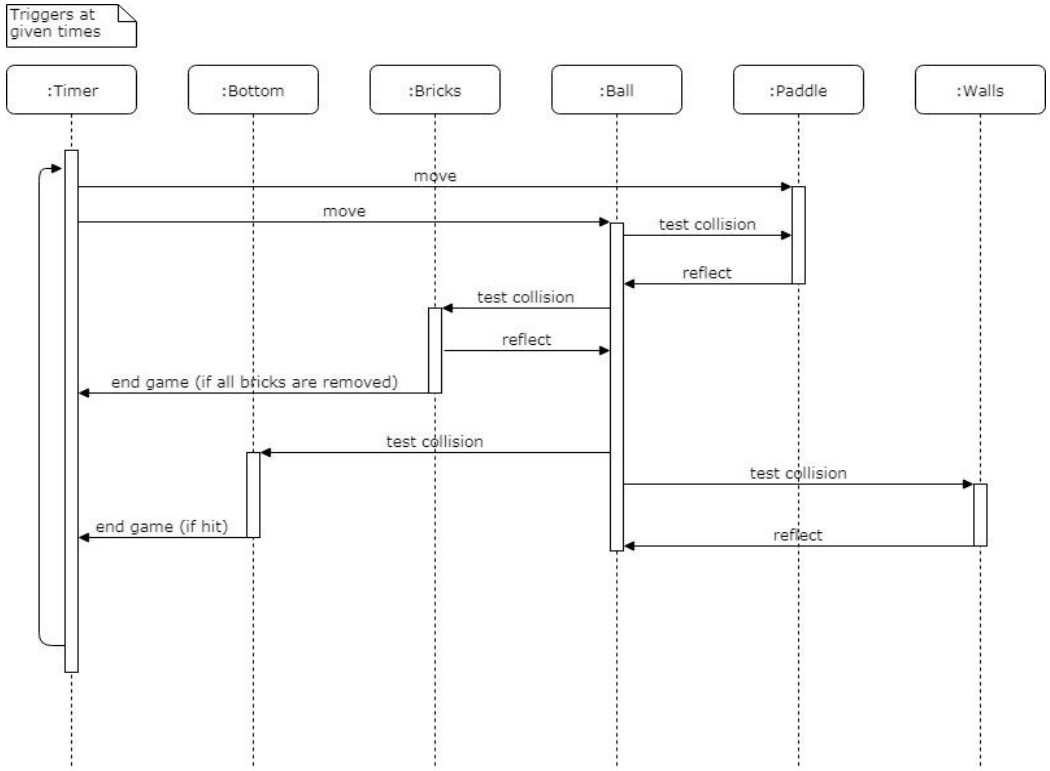
Picture 9. UML diagram – v4.



Picture 10. UML diagram – v5.

This version of the diagram was also published on the general forum.

In addition, at one stage of the work on the class diagram, a Sequence diagram was also developed (pict. 11).



Picture 11. UML sequence diagram.

Finally, we came to the conclusion that time had almost come to an end, and almost all of it was spent studying new material, preparing the diagram and planning the game. And we had a very little time left for coding.

That is why, it was decided to take the existing code[2] as a basis and refactor it on our own according to MVC and other patterns introduced in the course material.

Based on the screenshots, you might think that our project is identical to the borrowed by us original one, but it's not! We are not plagiating! Though the processing source is not developed by us, we decided to focus on the MVC-part and did our best to refactor it accordingly.

Two of us worked on refactoring the code, and the third was preparing documentation for the project.

The result of the joint work we are sharing with you today.

Thank you for attention!

[2] https://www.openprocessing.org/sketch/104480

Appendix:

Illustrations:

I) Current Version History Master Branch GIT

II) Currents structure of classic part (waiting for conversion to MVC)

III+IV) Current MVC-Structure

I)

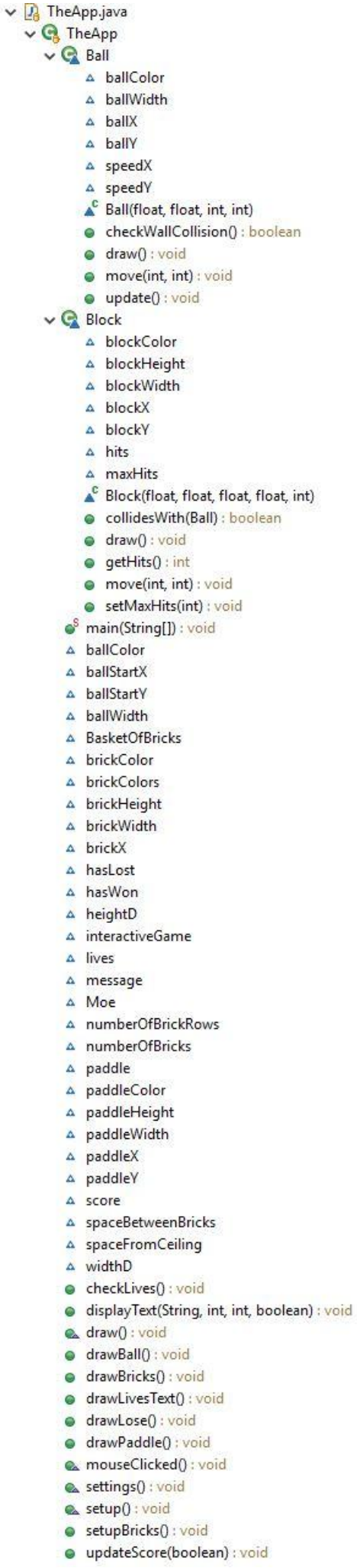
History

Project: capstone1-TheProject [capstone1-Team42.2]

Id	Message	A...	Authored Date	C.	Committed Da
3952292	master origin/master HEAD Names changed + extension	Buecl	20 minutes ago	Buec	20 minutes ago
e91803a	Merge branch 'master' of https://github.com/JHPeters/capstone1-Team42.git	Buecl	22 minutes ago	Buec	22 minutes ago
6cc4cb5	Try to synch	Buecl	23 minutes ago	Buec	23 minutes ago
c13e4e7	Implementing models into interactive game	Buecl	25 minutes ago	Buec	25 minutes ago
f3c6e23	Screenshots and UML v6	Capri	54 minutes ago	Capi	54 minutes ago
28348aa	deleted Papyrus test-files reorganised Folders	Buecl	2 hours ago	Buec	2 hours ago
92ddcbC	Just to get things synched	Buecl	2 hours ago	Buec	2 hours ago
f905eb6	Added model classes	Buecl	2 hours ago	Buec	2 hours ago
06ba20d	BallView PaddleView BlockView	Buecl	11 hours ago	Buec	11 hours ago
d160f92	What the Hell, Git is freaking me....	Buecl	11 hours ago	Buec	11 hours ago
2cb2b7c	Merge remote-tracking branch 'origin/master'	Buecl	12 hours ago	Buec	12 hours ago
72be4f3	Added missing classes	Buecl	12 hours ago	Buec	12 hours ago
38a30a4	Added missing classes (views and abstract classes)	Buecl	12 hours ago	Buec	12 hours ago
bbf263b	Removed duplicate main method from TheApp	Danie	23 hours ago	Dani	23 hours ago
1976852	Added Views	Danie	25 hours ago	Dani	24 hours ago
20d69ab	Commented	Buecl	2 days ago	Buec	2 days ago
0eecab6	Upload core.jar	Buecl	3 days ago	Buec	3 days ago
985a64c	Fixed Path to core.jar	Buecl	3 days ago	Buec	3 days ago
08666e1	More testing ShitHub	Buecl	3 days ago	Buec	3 days ago
74b614c	Staging core.jar	Buecl	3 days ago	Buec	3 days ago
ec39518	Edited gitignore, try to upload core.jar	Buecl	3 days ago	Buec	3 days ago
57765f9	One more try to include core.jar	Buecl	3 days ago	Buec	3 days ago
55aef0c	Also cleaning up my git mess and check if still able to commit :-(((	Buecl	3 days ago	Buec	3 days ago
8fd3c97	Mostly practising commit push. Also added some debug messages.	Danie	3 days ago	Dani	3 days ago
bd74fb5	Test of push	Danie	3 days ago	Dani	3 days ago
2c6d7f9	Added code to build and instantiate the MVC components. Added GameController.	Danie	3 days ago	Dani	3 days ago
3333bc0	Changed names of UML, to bind draw io as external editor permanently	Buecl	4 days ago	Buec	4 days ago
4a92df4	Changed number of lives (testing if upload is still possible, struggling with git)	Buecl	4 days ago	Buec	4 days ago
47dbe93	Added Breakout UML 0.5	Capri	5 days ago	Capi	5 days ago
775fbf8	Just a few outcommented lines in "Setup" to be used for building the game objects.	Danie	5 days ago	Dani	5 days ago
0adb1dk	Minor scaffold changes	Danie	5 days ago	Dani	2018-09-22 15:5
8743e64	Changed controll habits and added out of bounds check	j.schi	6 days ago	j.sch	6 days ago
d29e653	Update TheApp.java	dcsdk	7 days ago	GitH	7 days ago
de7097a	Merge pull request #2 from dcsdk/patch-2	Capri	7 days ago	GitH	7 days ago
edd33a7	Create Breakout UML 0.4	dcsdk	7 days ago	GitH	7 days ago
fc6686b	Compare useabilty Mouse or Keyboard steering	Buecl	8 days ago	GitH	8 days ago
90aae56	Added a Sequence Diagram	Jan H	9 days ago	Jan H	9 days ago
8d07985	Added a note to revision history	Capri	10 days ago	Capi	10 days ago
fcbbf42	Merge pull request #1 from dcsdk/patch-1	Capri	10 days ago	GitH	10 days ago
9625a2b	Create Breakout UML 0.3	dcsdk	10 days ago	GitH	10 days ago
b9fca19	Created Docs and materials folder	Capri	11 days ago	Capi	11 days ago
8f85c31	Upload working materials	Buecl	11 days ago	GitH	11 days ago
7770e90	processing.core.PApplet (fixed)	Buecl	13 days ago	Buec	13 days ago
787c243	Fixed the Missing Buildpath to core3.java and imported core.jar into project	Buecl	13 days ago	Buec	13 days ago
f16abc5	Create README.md	JavaC	3 weeks ago	GitH	3 weeks ago
129c38c	Update .classpath	JavaC	3 weeks ago	GitH	3 weeks ago
ef5ed07	Initial Commit	Tom	3 weeks ago	Tom	3 weeks ago

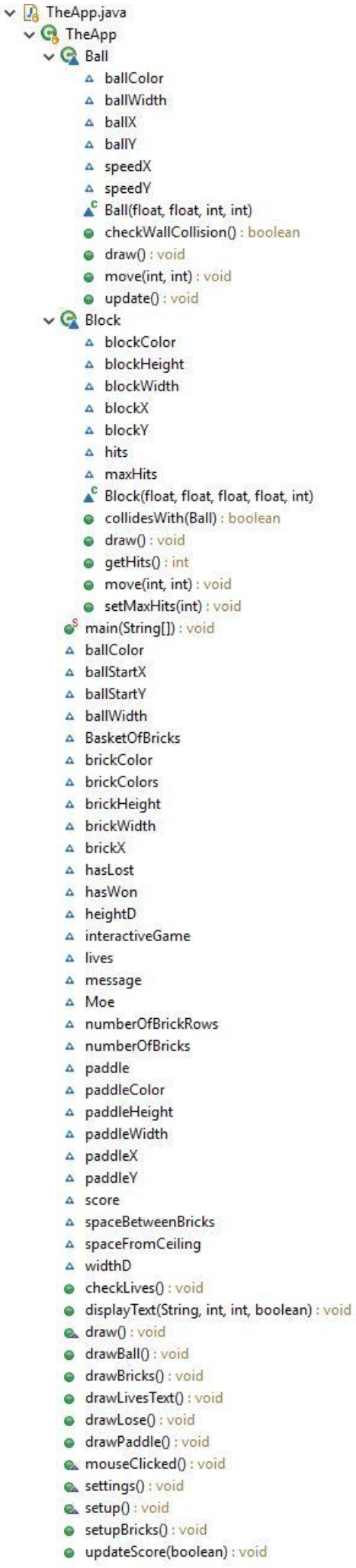
Current Version History Master Branch GIT

II)



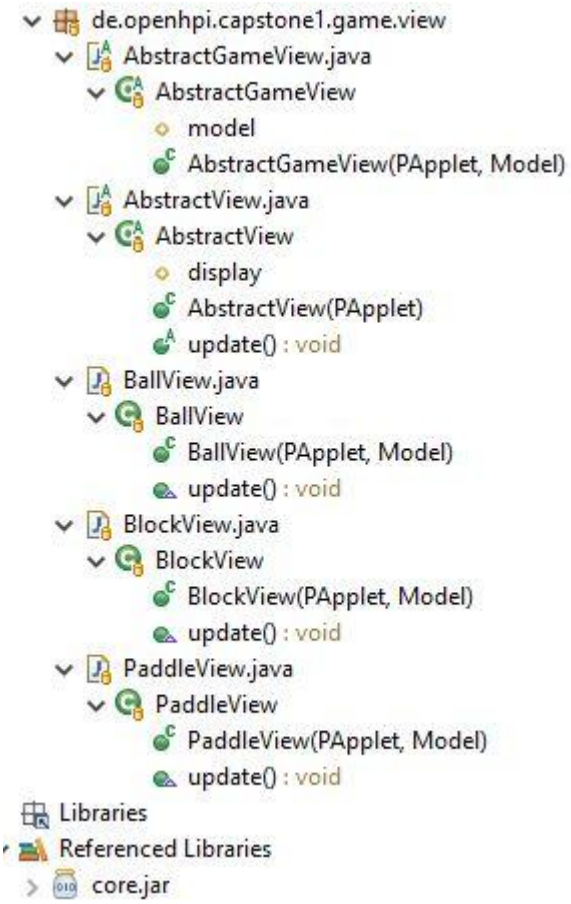
Currents structure of classic part (waiting for conversion to MVC)

III)



Current MVC-Structure

IV)



Current MVC-Structure