'Balls of Steel'

Group 4

John Laurin
Joakim Åkerlund
Milan Ivanovic
Daniel Öberg
Christoffer Lundell Johansson

5.5 Detailed Design

This document describes the detailed design. It will list all public and protected methods (not the private ones). The marks are from UML and symbolise the following:

Mark	
+	Public
#	Protected
-	Private
~	Package

Menu

+ Constructor()

Parameters: none
Return Value: void
Description:
Initializes the menus User Interface (UI). A list is shown with the following elements: New Game, High Score, Settings, Instructions and Exit Game. Checks whether there is an earlier gaming session is available and if it is it also present the user with the ability to continue that game.
Pre-condition:
None
Post-conditions:
None
Called by:
Root
Calls:
isContinueAvailabe

+ ContinueGame()

Parameters: none
Return Value: void
Description:

Restores the screen to gameplay and passes the controlflow to Root which in turn creates GameLogic and starts the game.
Pre-condition:
isContinueAvailabe returns true.
Post-conditions:
If GraphicsData and GameData doesn't exist they get constructed.
Called by:
Called by delegates from keypad made by the java api.
Calls:
loadData

+ NewGame()

Parameters: none
Return Value: void
Description:
Restores the screen to gameplay and passes the controlflow to Root which in turn creates GameLogic and starts the game.
Pre-condition:
None
Post-conditions:
Graphics and GameData (leveldata and so on) will be loaded.
Called by:
Called by delegates from keypad made by the java api.
Calls:
loadData

+ Highscore()

Parameters: none
Return Value: void
Description:
Creates a new screen and lists the highscore on it.
Pre-condition:
None
Post-conditions:
None
Called by:
Called by delegates from keypad made by the java api.
Calls:
None.

+ Settings()

Parameters: none
Return Value: void
Description:

Creates a screen that present the user with the option of turning sound on or off.

Pre-condition:

None

Post-conditions:

That GameData will contain these settings.

Called by:

Called by delegates from keypad made by the java api.

Calls:

None

+ Instructions()

Parameters: none

Return Value: void

Description:

Creates a screen that present the user with information about the game on how to play it.

Pre-condition:

None

Post-conditions:

None

Called by:

Called by delegates from keypad made by the java api.

Calls:

None

+ Exit()

Parameters: none

Return Value: void

Description:

Optionally saves the data and exits.

Pre-condition:

None

Post-conditions:

The program shuts down

Called by:

Called by delegates from keypad made by the java api.

Calls:

None

- loadData(continue: boolean)

Parameters:

continue: boolean

Tells whether this function should try to load data from an earlier gaming

session, if the value is true then it will.
Return Value: void
Description:
Loads data from the phones memory card and creates or updates the GraphicsData and GameData structures.
Pre-condition:
Images should exist on the same path as the game excutable. If not the program throws an IOException which will lead to that the game shuts down.
Enough memory should exist to load the images into it otherwise throw out of memory exception.
Post-conditions:
The global structures GraphicsData and GameData will be created.
Called by:
NewGame
CreateGame
Calls:
FileWrapper.getFile();//FIX THIS!

- isContinueAvailabe(): boolean
Parameters: none
Return Value:
Returns true if there exists data to continue an earlier gaming session.
Description:
Checks if there exists data to continue an earlier gaming session.
Pre-condition:
None
Post-conditions:
None
Called by:
Constructor
Calls:
FileWrapper.fileExists(); //FIX THIS!

GameLogic
+ run() **Parameters:** none **Return Value:** void **Description:** An loop that calls AI, Physics and Renderings update function. Also updates the time and checks the time left for the player.

Pre-condition:
None
Post-conditions:
None
Called by:
Root
Calls:
AI.update
Physics.update
Rendering.update

AI

+ update()

Parameters: none
Return Value: void
Description:
An loop that calls AI, Physics and Renderings update function.
Pre-condition:
None
Post-conditions:
None
Called by:
GameLogic.run
Calls:
moveEnemies

- moveEnemies()

Parameters: none
Return Value: void
Description:
Moves the enemies according to a pattern. Changes the enemies coordinates.
Pre-condition:
None
Post-conditions:
GameData is updated with the new coordinates for the enemies.
Called by:
update
Calls:

None

- attackPlayer()

Parameters: none
Return Value: void
Description:
Removes some of the players time left.
Pre-condition:
None
Post-conditions:
GameData.timeleft is updated.
Called by:
update
Calls:
Audio.playSound

Physics

+ update()

Parameters: none
Return Value: void
Description:
Checks if the player bumped into an object or an enemy. If the player bumped into an object it changes the angle the ball is traveling. If it bumps into an enemy it calls AI to tell it to do something. It also moves the players ball.
Pre-condition:
None
Post-conditions:
None
Called by:
GameLogic.run
Calls:
isPlayerCloseToEnemy
isPlayerCloseToObject

- isPlayerCloseToObject(): boolean

Parameters: none
Return Value:
returns true if an object and the players ball have overlapped.
Description:

Checks if an object and the players ball have overlapped. If so it checks for what kind of object it is. If it is a wall it changes the angle the ball is traveling. If it is a powerup it changes the GameData depending on what kind of powerup it is.

Pre-condition:

None

Post-conditions:

None

Called by:

update

Calls:

none

- isPlayerCloseToEnemy(): boolean

Parameters: none

Return Value:

returns true if an enemy and the players ball have overlapped.

Description:

Checks if an enemy and the players ball have overlapped. If so it calls AI.attackPlayer.

Pre-condition:

None

Post-conditions:

None

Called by:

update

Calls:

AI.attackPlayer

- movePlayer()

Parameters: none

Return Value: none

Description:

Moves the players ball according to the angle and speed specified in GameData.

Pre-condition:

None

Post-conditions:

GameData is updated with the new coordinate for the player.

Called by:

update

Calls:

None

Filewrapper

+ getData()

Parametes: Defines what data type is to be loaded (level, save, graphic etc)
Return Value: Data
Description:
Fetches specific data.
Pre-condition:
That the data files are not corrupted.
Post-conditions:
None
Called by:
Called by GameLogic when loading the game.
Calls:
none

+ saveGameData()

Parametes: none
Return Value: void
Description:
Saves the game data.
Pre-condition:
The user shuts down a game in progress.
Post-conditions:
None
Called by:
Called by GameLogic when saving the game.
Calls:
none

+ sendGraphicsData()

Parametes: none
Return Value: Graphical Data
Description:
Sends the graphical data that was loaded from the filesystem to the rendering class.
Pre-condition:
That the data in GraphicsData is intact and available. If this fails exit the game.
Post-conditions:
None

Called by:
Called by GameLogic when loading the game.
Calls:
Filewrapper.getData()

+ sendGameData()

Parametes: none
Return Value: void
Description:
Sends the game data that was loaded from the filesystem to the rendering class.
Pre-condition:
That the data in GameData is intact and available. If this fails exit the game.
Post-conditions:
none
Called by:
Called by GameLogic when loading the game.
Calls:
Filewrapper.getData()

Input

+ pollInput()

Parametes: none
Return Value: Data regarding for which keycommands are issued
Description:
The method listens for any keypresses to occur
Pre-condition:
The game must be loaded.
Post-conditions:
None
Called by:
Called by GameLogic when loading the game.
Calls:
none

+ sendInput()

Parametes: none

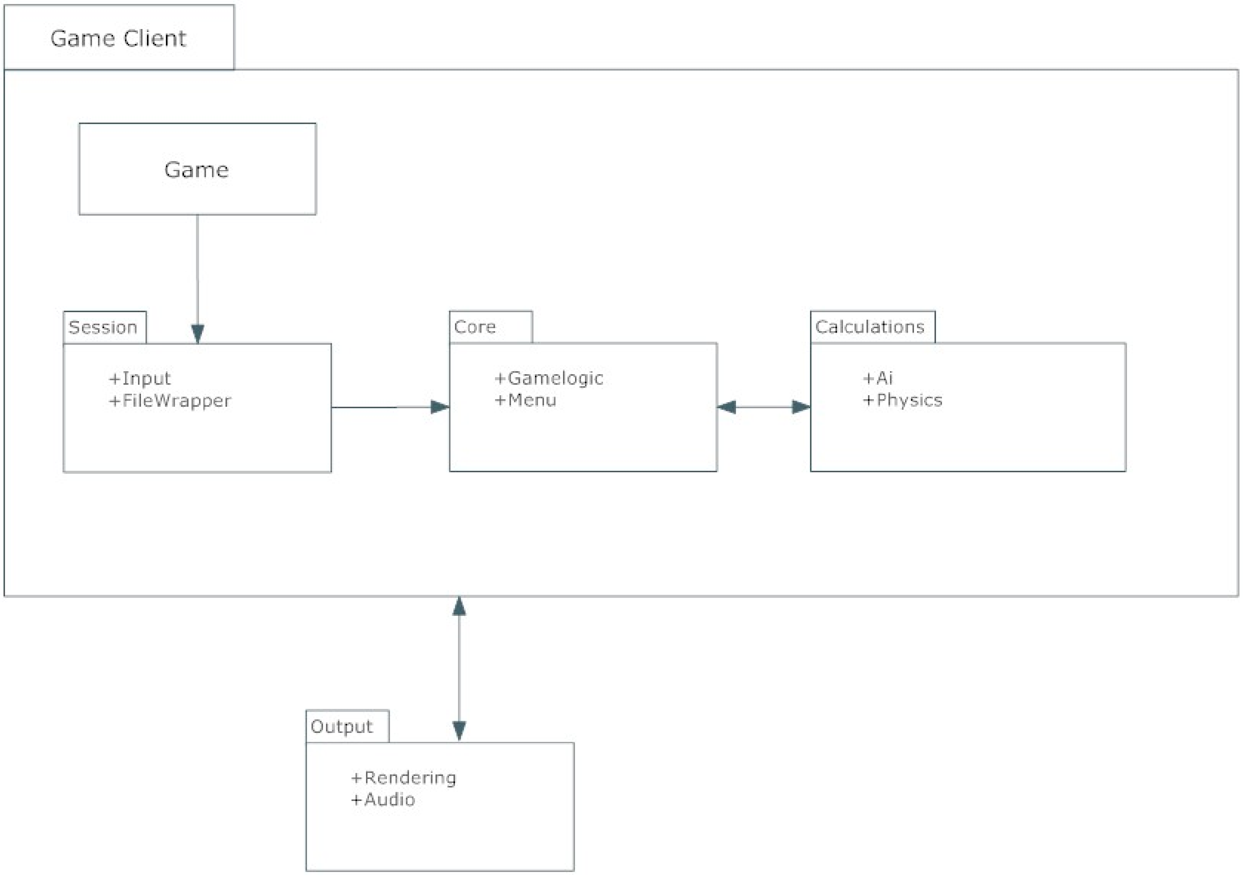
Return Value: void **Description:** Sends data regarding the keypresses to the GameLogic. **Pre-condition:** The game must be loaded and pollInput() must be running. **Post-conditions:** None **Called by:** Called by GameLogic when pollInput() has received data to send. **Calls:** none

Rendering
+ updateGraphics() **Parametes:** Data regarding graphics (updated position etc) **Return Value:** void **Description:** Updates the graphical output on the mobile phone. **Pre-condition:** The game must running. **Post-conditions:** None **Called by:** Called by GameLogic while running the game. **Calls:** none

Audio
+ playSound()

Parametes: Data regarding the audio (what sounds to play etc)
Return Value: void
Description:
Plays the sound.
Pre-condition:
The game must running.
Post-conditions:
None
Called by:
Called by GameLogic while running the game.
Calls:
none

5.6 Package Diagram


Game Client
Game
Session
+Input
+FileWrapper
Core
+Gamelogic
+Menu
Calculations
+Ai
+Physics
Output
+Rendering
+Audio