Politechnika Śląska

Wydział Automatyki, Elektroniki i Informatyki

Computer Programming

«Panda bears»

author	Katarzyna Sęk
instructor	Mgr inż. Wojciech Dudzik
year	2020/2021
lab group	----------
deadline	2021-05-20

1. Panda bears

Panda bears is a clicker game. It is programmed with usage of the SFML library with the C++ programming language.

At the beginning of the game players have the possibility to input their name in the main menu options. If there is no such input the player name will be setted as NoName.

In this game it is necessary to keep the three pandas alive by feeding them, playing with them and giving them enough sleep.

The goal of the game is to survive as long as possible without any of the panda bears dying. The longer you live, the higher your score is. The game ends with the death of any panda.

After losing the score is displayed and a player can go to the leaderboard located in the main menu to see which place he has.

2. Analysis of the project

2.1. Classes

Game - it initializes SFML Window, initializes supported keys, sets framerate, updates SFML events and runs the whole game.

State - it is the base state of the game, all states derive from it. It sets mouseposition and also the end state.

GameState - it is the state of the game, which displays the game screen. It creates pandas, bars and buttons and updates them, checking which button or panda is in which state (active, pressed, idle).

GameOverState - it is the state of the game, which displays the game over screen. It also saves the score of the player into the .txt file.

HowToPlayState - it is the state of the game, which displays the how to play screen, with some basic instructions on how to play the game.

LeaderboardState - it is the state of the game, which displays the Leaderboard screen. It reads the saved scores and players names from the file and sorts it.

MainMenuState -it is the state of the game, which displays the main menu screen. It creates buttons and updates them, if they are pressed the next state is pushed.

OptionsState - it is the state of the game, which displays the options screen. It allows input player names with correct syntax.

Panda - it is responsible for settings of visualization of Panda bears, it also sets the current state of pandas and sets/checks/decreases/increases their stats.

Button - it is responsible for settings of visualization of buttons, it also sets the current state of buttons.

Bar - it is responsible for settings of visualization of bars and their texts, it also updates them.

AnimationComponent - it adds an animation with specific parameters and plays it.

Animation - it creates an animation from a texture sheet with the usage of specific parameters passed to it.

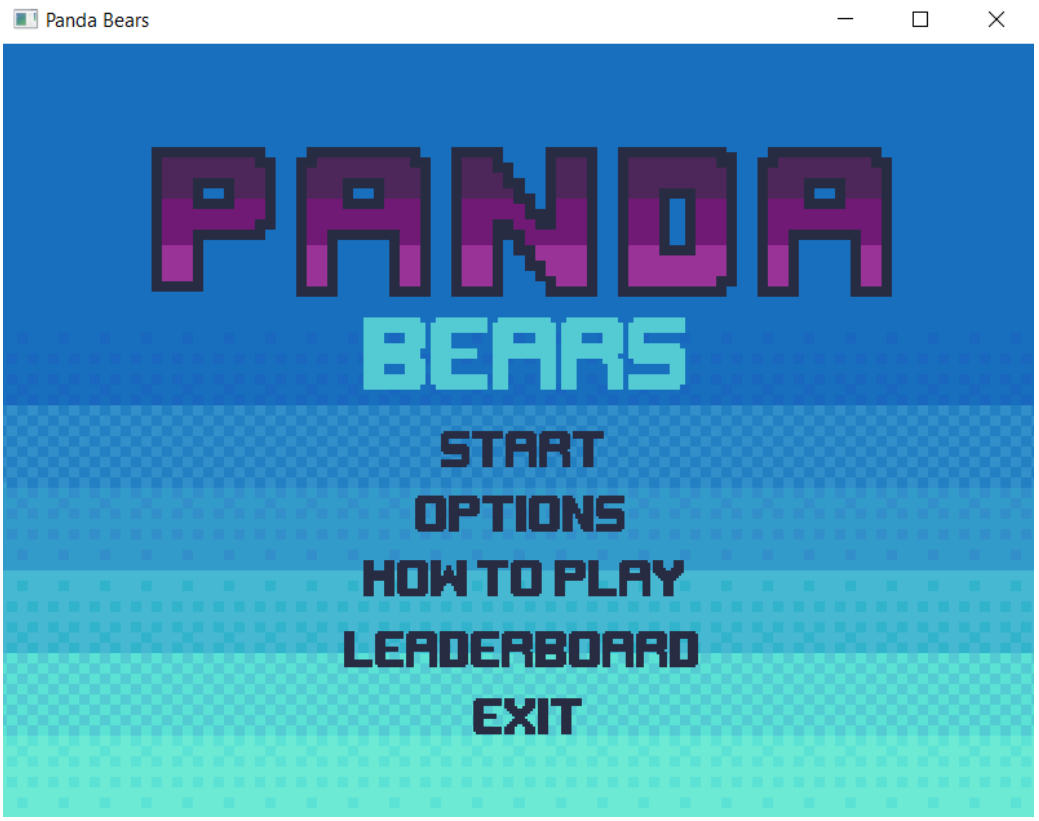
TextureManager - derives from ResourceManager, loads new textures.

ResourceManager - stores the resource name and a counter, so they can be reused and accessed with a string.

FileMissingException - exception class.

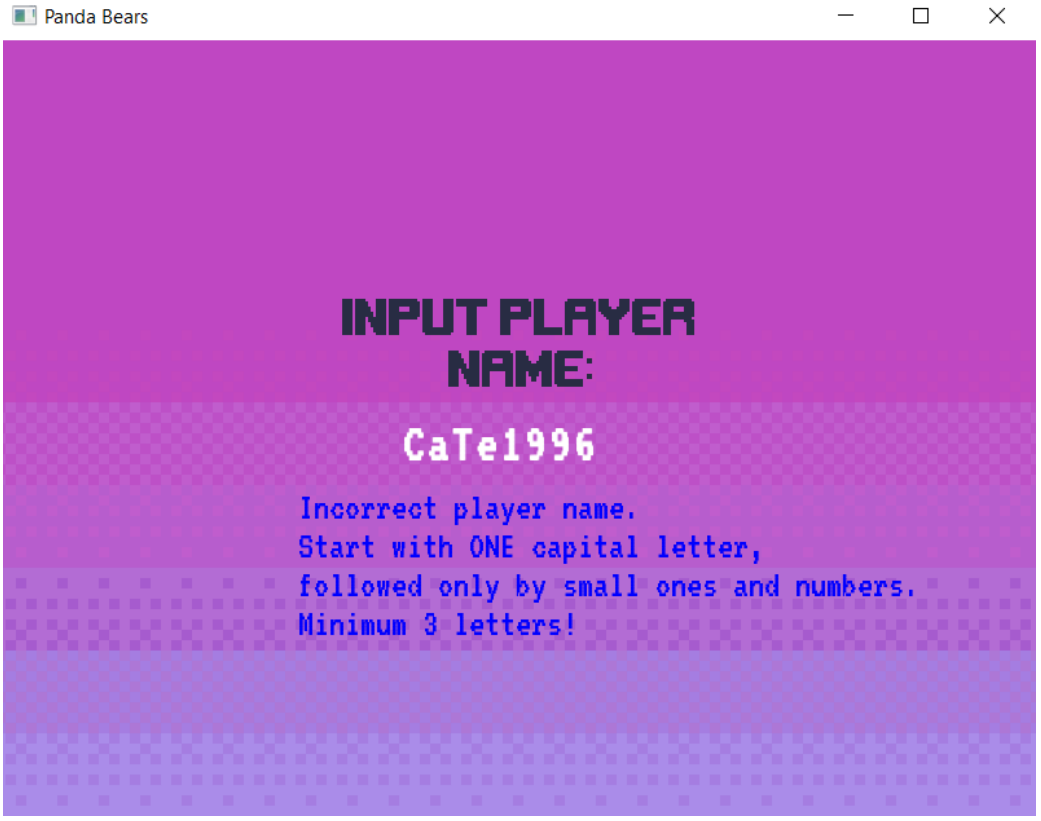
2.2. Program interface

After launching the Panda bears game the main menu is visible. From there the player has several options to choose. If the player wants to play the game under some nickname it is necessary to select the options button.

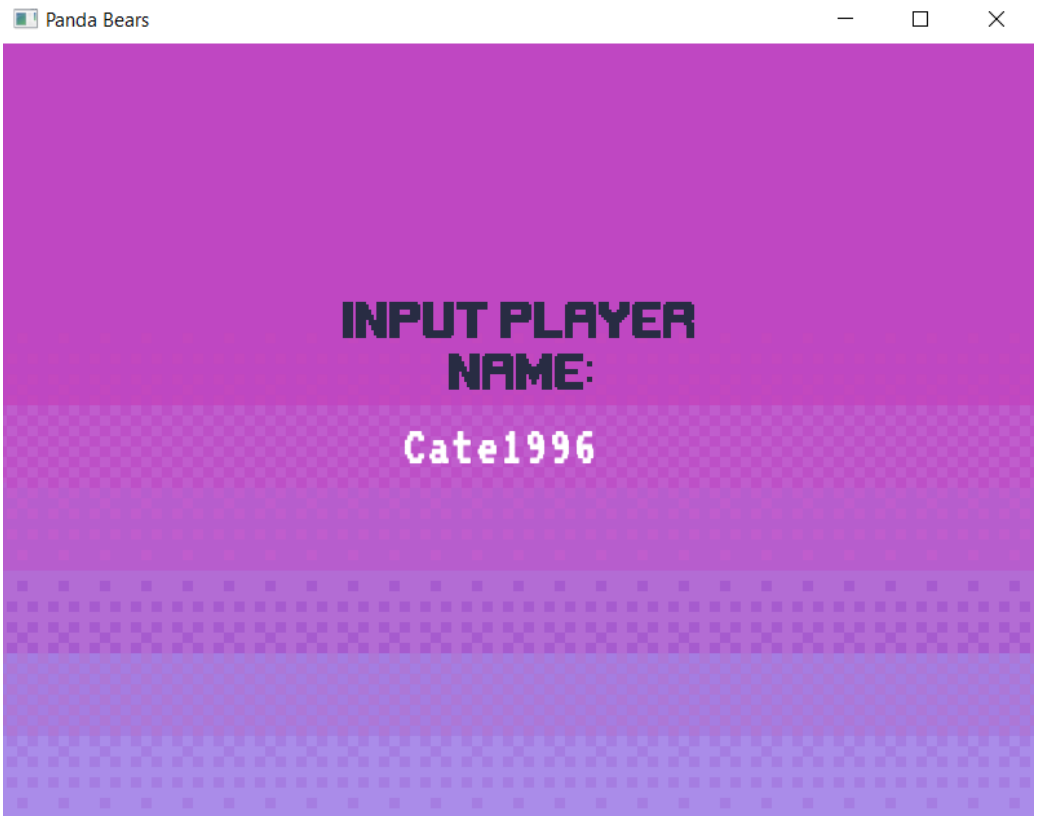


Picture 1. Main menu

Next, selecting the options button will give the opportunity to write a player name. There are some rules regarding this. If the player name is inputted in the wrong way, the following message will appear.

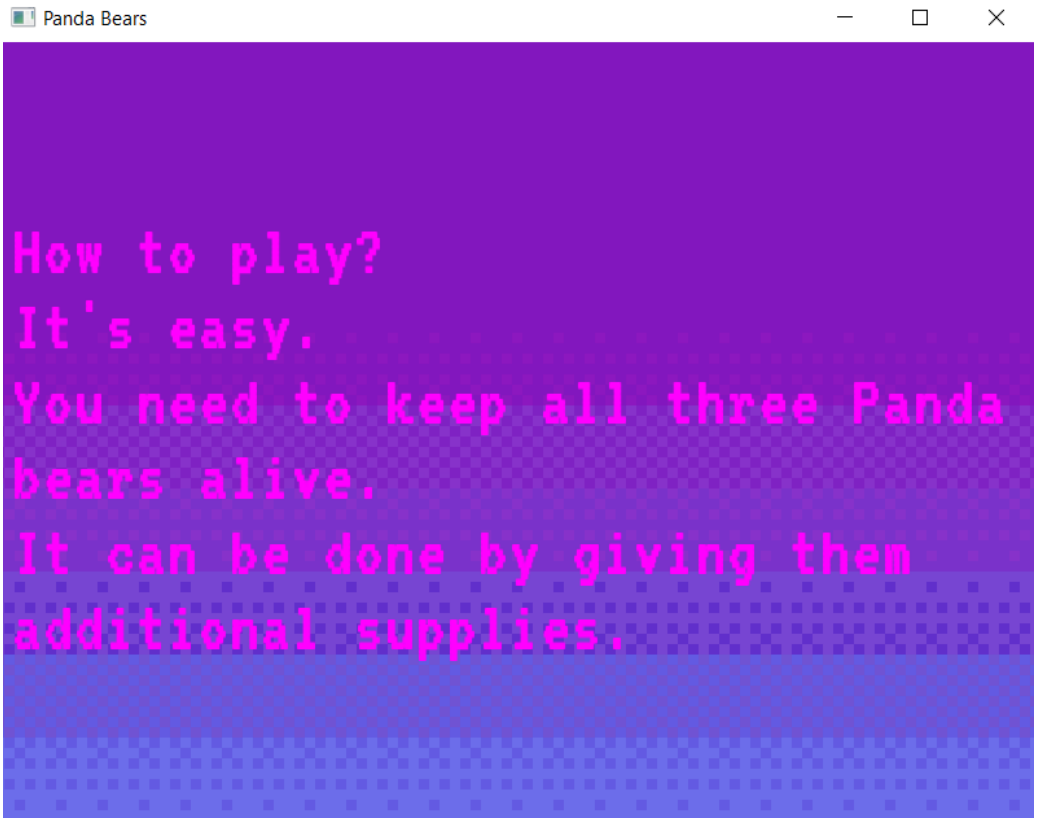


Picture 2. Options with wrong input



Picture 3. Options with correct input

After inputting the correct name and pressing enter, again, the main menu is visible. Next button which may be helpful is how to play. After clicking it a short instruction appears.



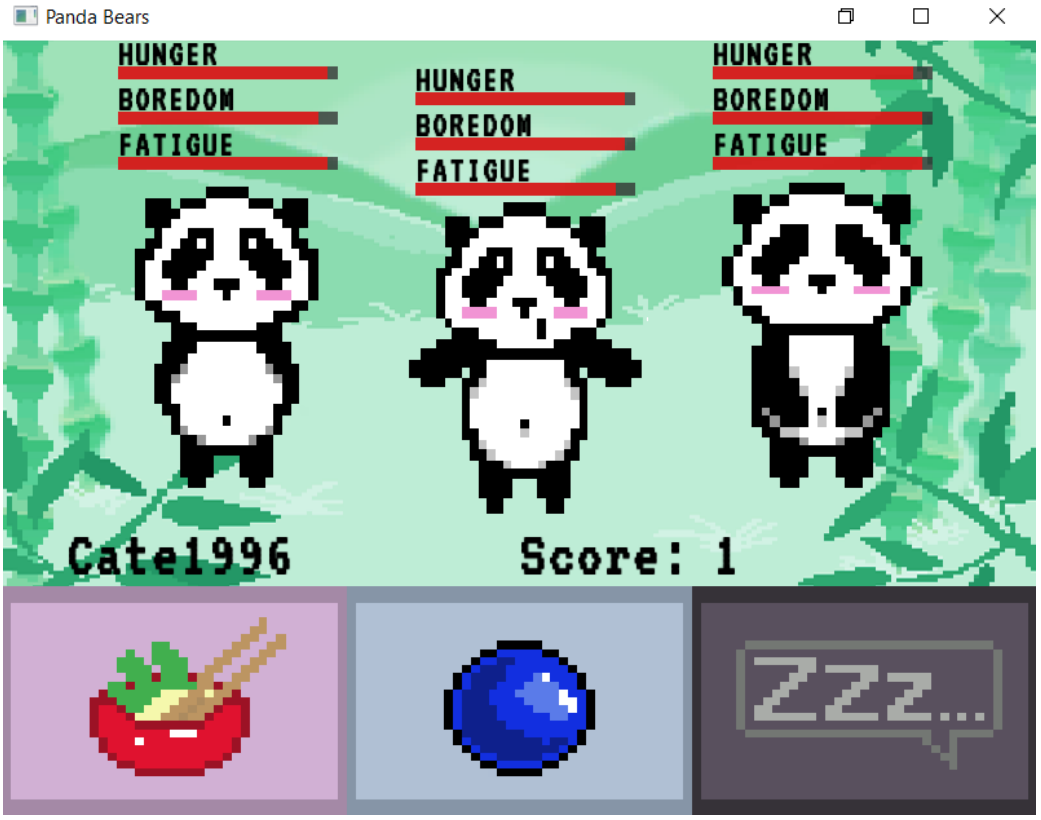
Picture 4. How to play

Next, finally, the player can start the game by choosing the start button in the main menu.

The game immediately starts and panda bears with bars and buttons are visible. Above the food, fun and sleep buttons player can see inputed earlier name and current score.

Score rises with the time that panda bears stay alive. To make this work players need to choose the correct panda by clicking on it and raise her stats by clicking the correct button.

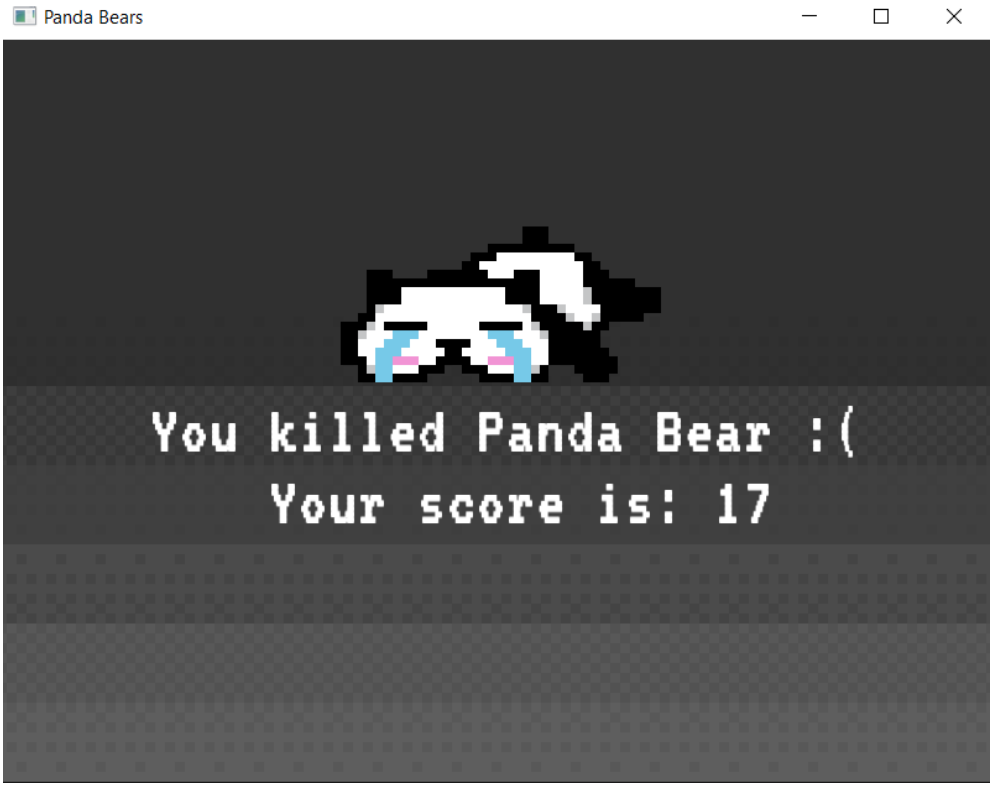
If in the options the player name won't be inputed automatically: NoName will be set.



Picture 5. Game started

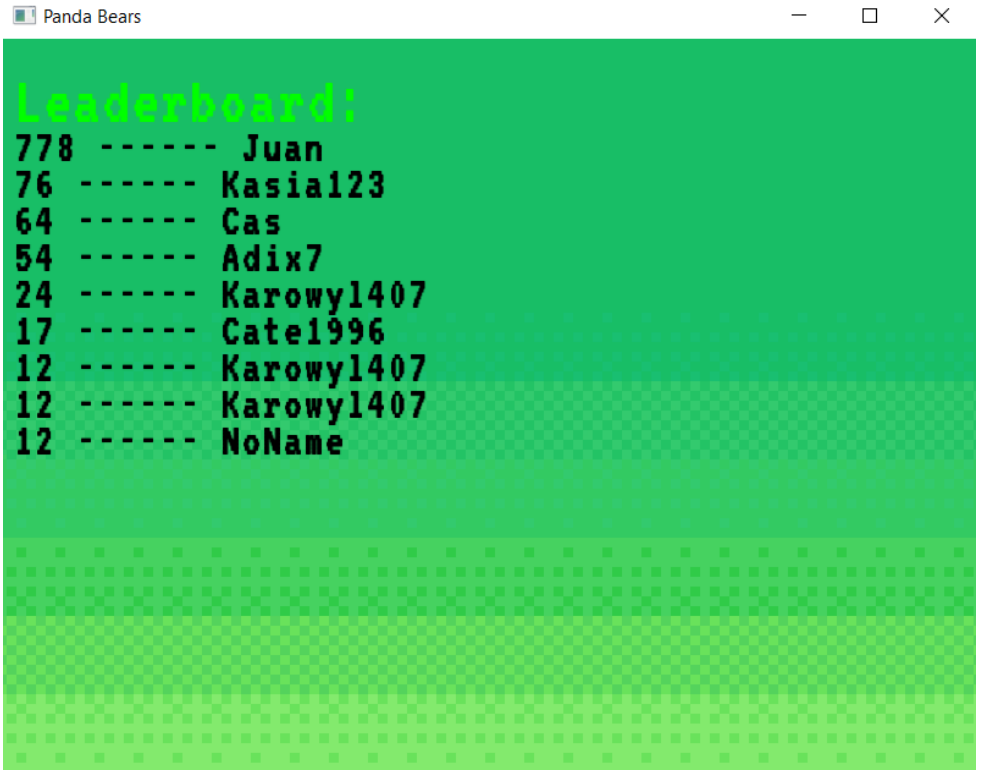
Picture 6. Game started with no input name

If any of the panda bears' bars drop to zero the game ends and a screen with some message and gained score will appear.



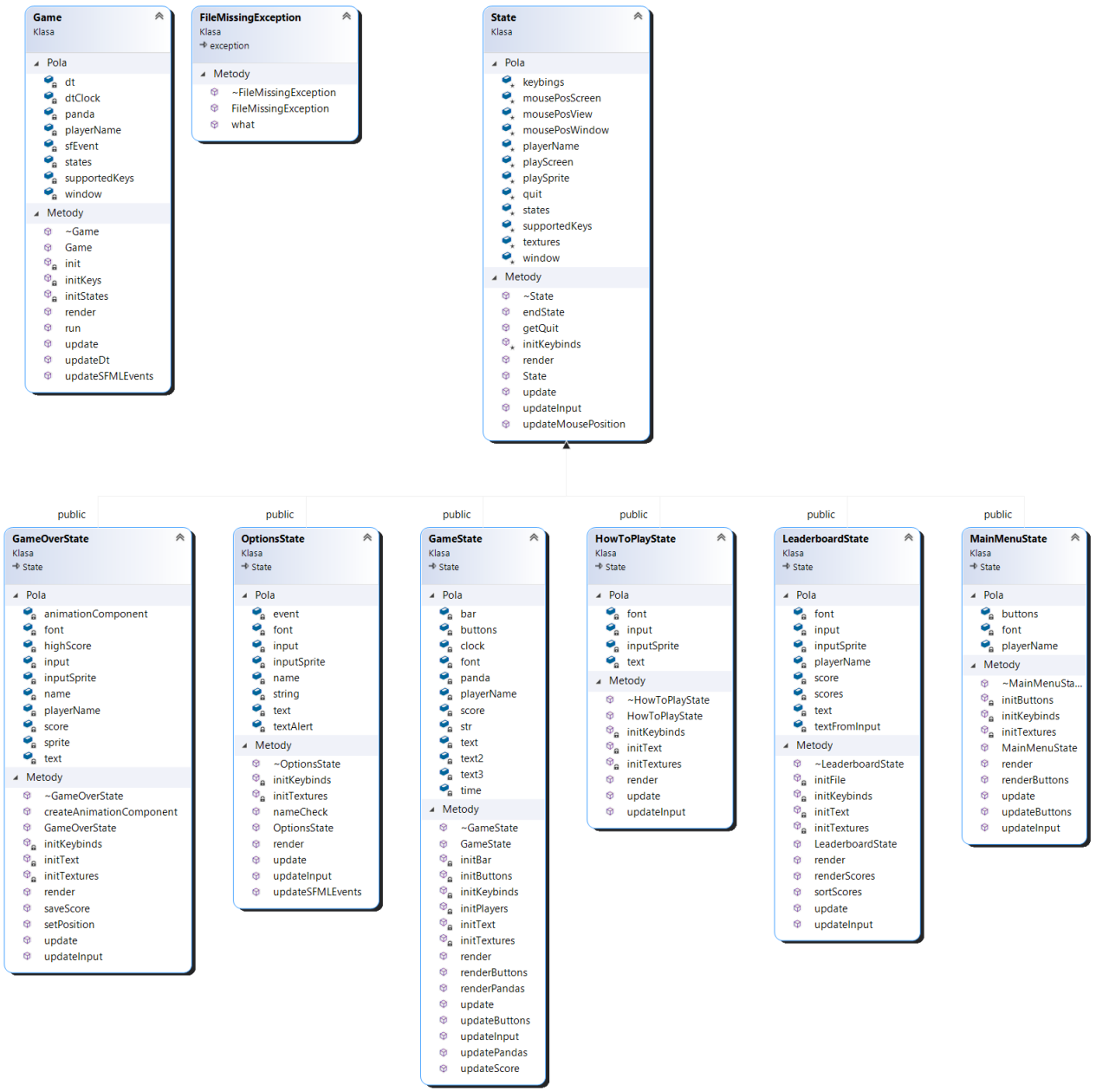
Picture 7. End game

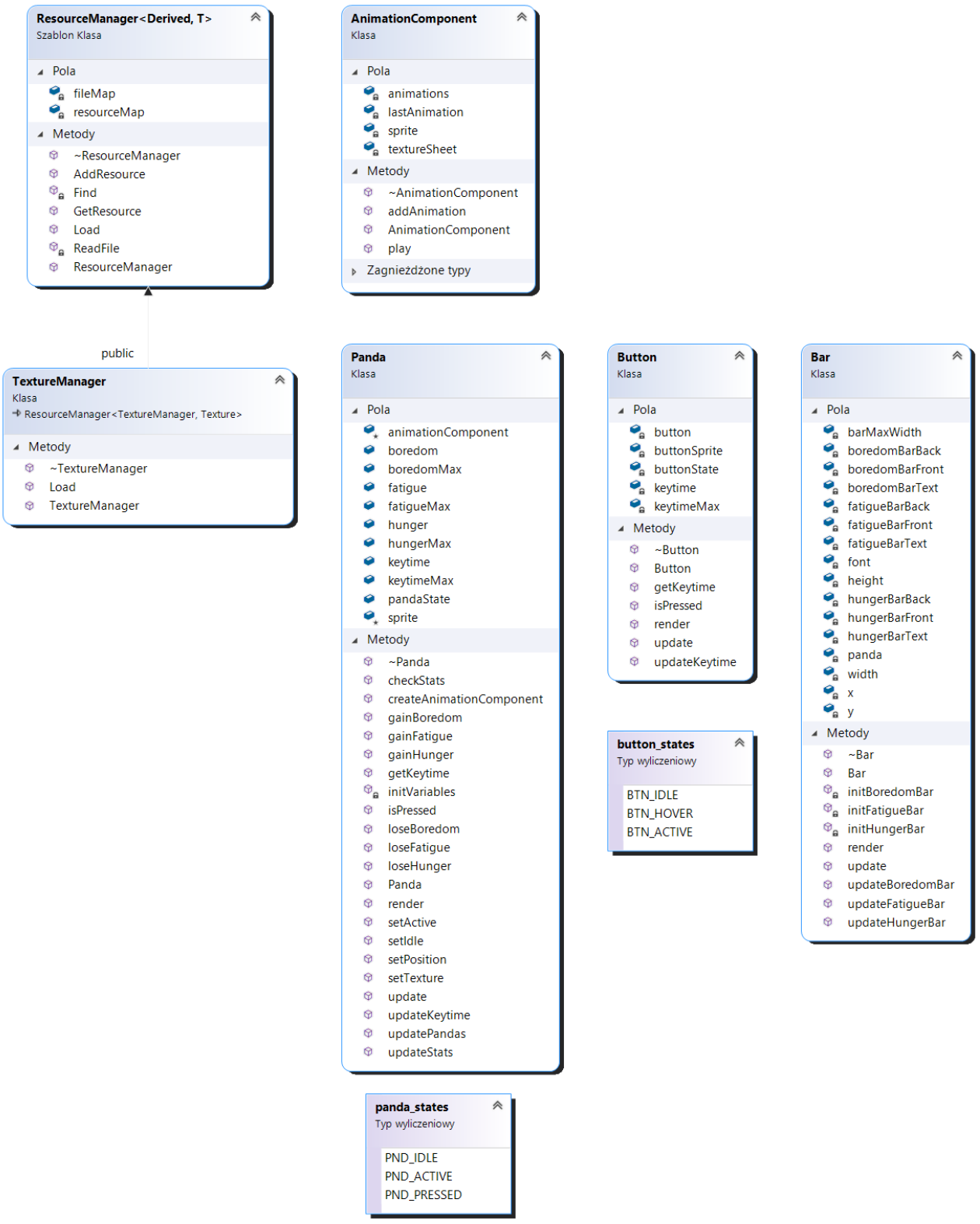
Next, the player needs to exit that screen by pressing escape. After that if the player wants to see his score against the other one's score, the button Leaderboard finds usage. Selecting it gives access to the leaderboard. It contains all results of previous games sorted from the best to the worst score.



Picture 8. Leaderboard

2.3. Class diagram




ResourceManager<Derived, T>
Szablon Klasa
Pola
fileMap
resourceMap
Metody
~ResourceManager
AddResource
Find
GetResource
Load
ReadFile
ResourceManager
AnimationComponent
Klasa
Pola
animations
lastAnimation
sprite
textureSheet
Metody
~AnimationComponent
addAnimation
AnimationComponent
play
Zagnieżdżone typy
public
TextureManager
Klasa
ResourceManager<TextureManager, Texture>
Metody
~TextureManager
Load
TextureManager
Panda
Klasa
Pola
animationComponent
boredom
boredomMax
fatigue
fatigueMax
hunger
hungerMax
keytime
keytimeMax
pandaState
sprite
Metody
~Panda
checkStats
createAnimationComponent
gainBoredom
gainFatigue
gainHunger
getKeytime
initVariables
isPressed
loseBoredom
loseFatigue
loseHunger
Panda
render
setActive
setIdle
setPosition
setTexture
update
updateKeytime
updatePandas
updateStats
panda_states
Typ wyliczeniowy
PND_IDLE
PND_ACTIVE
PND_PRESSED
Button
Klasa
Pola
button
buttonSprite
buttonState
keytime
keytimeMax
Metody
~Button
Button
getKeytime
isPressed
render
update
updateKeytime
button_states
Typ wyliczeniowy
BTN_IDLE
BTN_HOVER
BTN_ACTIVE
Bar
Klasa
Pola
barMaxWidth
boredomBarBack
boredomBarFront
boredomBarText
fatigueBarBack
fatigueBarFront
fatigueBarText
font
height
hungerBarBack
hungerBarFront
hungerBarText
panda
width
x
y
Metody
~Bar
Bar
initBoredomBar
initFatigueBar
initHungerBar
render
update
updateBoredomBar
updateFatigueBar
updateHungerBar

3. Additional informations

The program uses a text file to store texture names with additional paths to them.

If the file or the texture from it is not loaded correctly the exception with a message will appear and the program will not work. The same way the absence of any of the given texture will be noted.

The program uses a text file to store scores with players' names.

If the file is not loaded correctly the exception with a message will appear and the program will not work. The same thing will happen if the file is empty.

4. Conclusions

It is the first time I programmed a whole game with the usage of SFML. It was quite an experience.
I learned a lot about Graphic modules, such as usage of sprites, textures, text, fonts, how to set them, set their position and operate on them. Additionally to that I also learned how to make a sheet for animation. All represented above graphics I made and prepared on my own. Also one of the interesting and quite easy aspects was handling the time in SFML.
For me, the most difficult part was programming the Window module. Opening and managing SFML Window itself was fine, but managing the window events was quite difficult. Making it work and resolving the problems was hard, especially one, with a very slow display of inputted letters. Fortunately, the instructor helped me with this matter.
I learned a lot about different types of containers, how to manage objects, access them and so on. All in all programming this game was pretty challenging, for most of the time interesting and quite fun.