CS102	**Fall 2021/2022**	Project Group	H
Instructor:	**Uğur Güdükbay**		
Assistant:	**Serkan Demirci**		

Criteria	TA/Grader	Instructor
Presentation		
Overall		

~ EscApe ~

Invicta

Deniz Tuna Onguner, Hasan Ege Tunç, Ali Kaan Şahin, Kerem Tekik, Tarık Berkan Bilge

Detailed Design Report

Draft Version

10 December 2021

Introduction

EscApe is a 2D desktop game that involves two game modes: single player and multiplayer game modes. In both of these modes, players aim to escape from a furious ape throughout the parkour that consists of grounds separated by spaces, and there are various obstacles on grounds (i.e. short walls/columns, low ceilings) that player should not hit to continue the game and obtain much higher scores.

Details

2.1 System Overview

Escape will be a desktop game. As a library choice, we will be using libGDX [1] and as a database we consider to use SQL and SQLite [2].

2.2 UML Diagram

Figure-1:

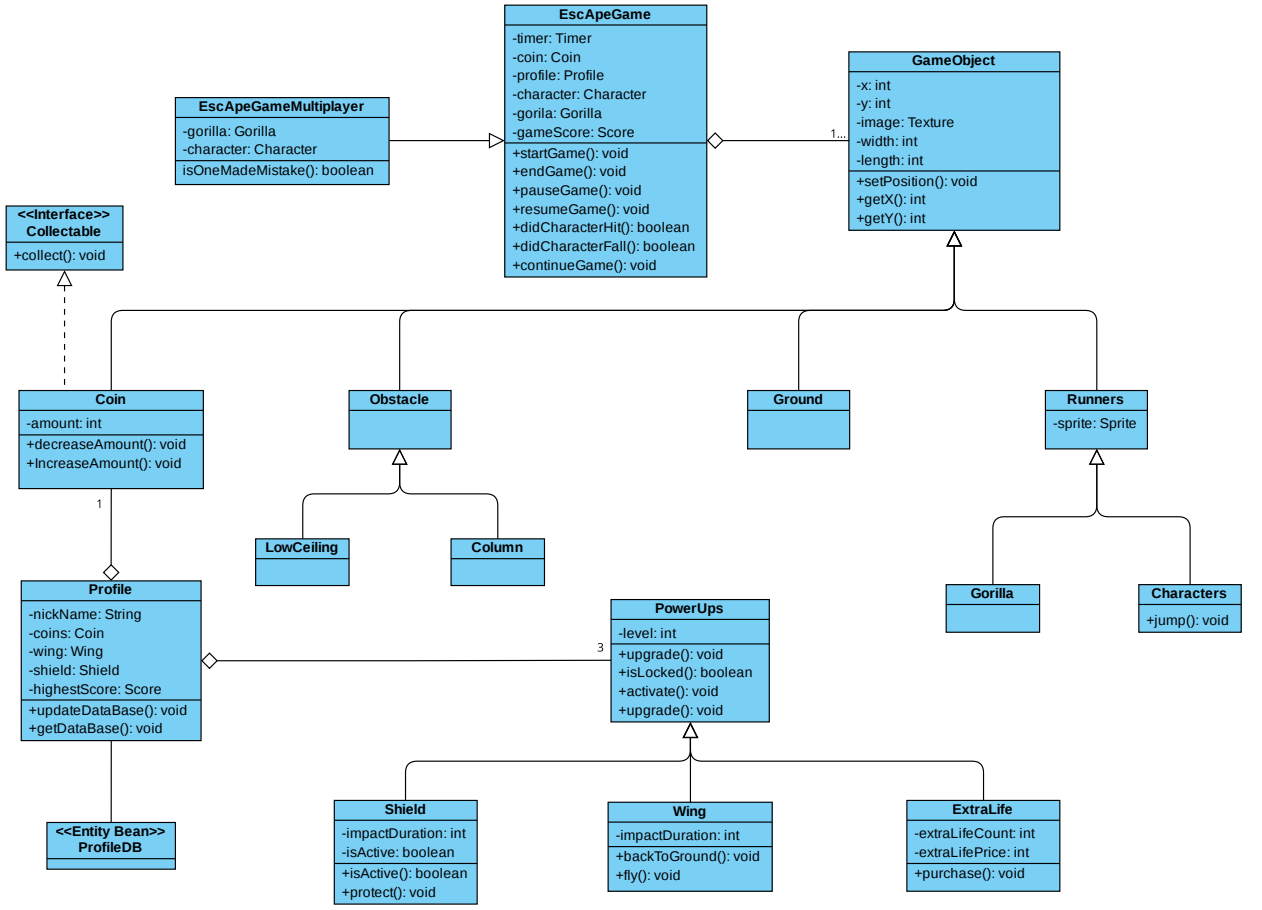
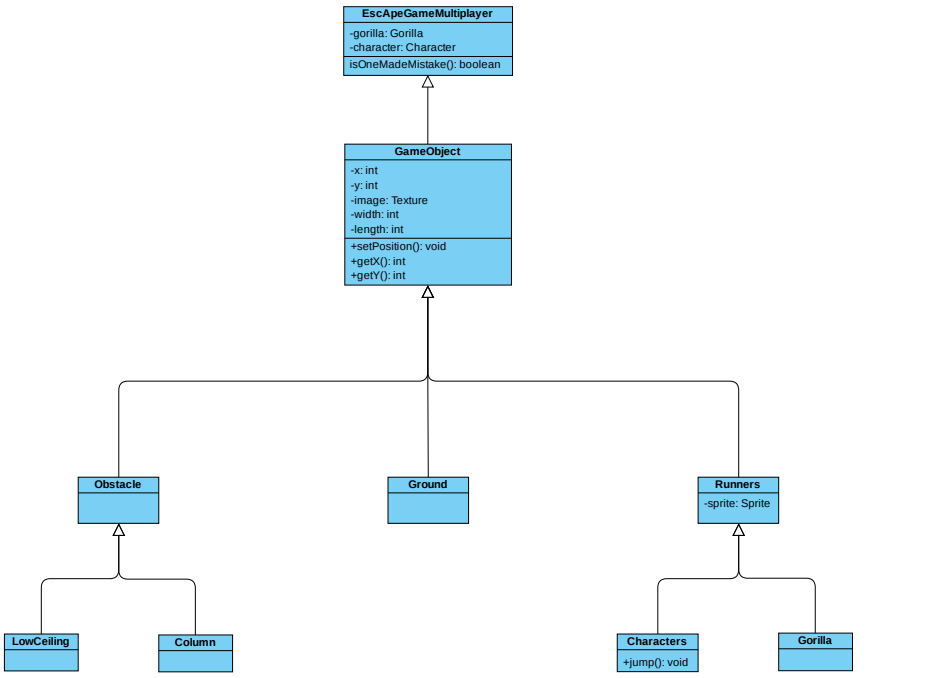


Figure-2:



2.3 Explanation of UML Diagram

class EscapeGameSinglePlayer

Timer timer: corresponds to the timer object to measure the time.
Coin coin: corresponds to coins collected in a single game.
Profile profile: corresponds to the profile plays the game.
Character character: corresponds to the character that runs the parkour.
Gorilla gorilla: corresponds to the gorilla that appears at the beginning and end of the game.
Score gameScore: corresponds to the score that is obtained during that game.
void startGame( ): starts the game.
void endGame( ): ends the game, utilizes methods to update coin and high-score information of active profile.
void pauseGame( ): pauses the game.
void resumeGame( ): resumes the game.
boolean didCharacterHit (Character ch): checks whether character hit any obstacle.-
boolean didCharacterFall (Character ch): checks whether character fell.
void continueGame (ExtraLife extra): continues the game depending on the user request, and extra's count. If extra > 0, the game continue; otherwise, game ends.

class EscapeGameMultiplayer extends EscapeGameSinglePlayer:

Character character2: corresponds to the second character that runs the parkour.
Gorilla gorilla2: corresponds to the second gorilla.
isOneMadeMistake( ): checks whether any character made mistake to terminate the game.

abstract class GameObject

int x: corresponds to x-coordinate of the bottom-left corner of the object.
int y: corresponds to y-coordinate of the bottom-left corner of the object.
int width: corresponds to the width of game object.
int length: corresponds to the length of game object
Texture image: corresponds to image that represents the object.
void setPosition (int x, int y): sets the position of the object, depending on the passed x and y values.
int getX(): returns the x-coordinate of the bottom-left corner of the object.
int getY(): returns the y-coordinate of the bottom-left corner of the object.

class Coin

int amount: corresponds to the quantity of coins, which are collected during a single game , or, which any specific profile has.
void decreaseAmount (int amount): subtracts the amount passed as parameter from the coin's amount.
void increaseAmount (int amount): adds the amount from the coin's amount.

class Runners:

Sprite runner: represents the movable version of texture

class Characters extends Runners:

void jump ( ): makes character jump

abstract class PowerUps

int level: represents the level of the power-up.
boolean isLocked: shows whether the power-up is locked or not.
abstract void activate( ): activates power ups during a game.
abstract void upgrade ( ): upgrades the power-ups.

class Wing extends PowerUps

int impactDuration: corresponds to seconds during which power-up remains activated.
void fly( ): provides a departure and flight during impact duration.
void backToGround( ): after impact duration ends, lands character to the game parkour.

class Shield extends PowerUps

int impactDuration: corresponds to seconds during which power-up remains activated.-
boolean isActive: corresponds the status of shield.
void protect( ): if shield is active, protects user from hit source ape catches.

class ExtraLife

int lifeCount: corresponds to the number of purchased extra lives.
final int PRICE: corresponds to price of extra life.
purchase(Coin coin): purchase one extra life, if the coin's amount is greater than the PRICE, else does nothing.

class Profile

Coin coins: corresponds to the coins information of profile.
Wing wing: corresponds to the wing information of profile.
Shield shield: corresponds to the shield information of profile.
String nickName: corresponds to the name of profile.
Score highestScore: corresponds to the highest score of profile.
void updateDataBase( ): updates the data of profile.
void getDataBase( ): access the data of profile.

2.4 Task Distribution

Hasan Ege Tunç:
Deniz Tuna Onguner:
Kerem Tekik:
Tarık Berkan Bilge:
Ali Kaan Şahin:

Summary & Conclusions

To conclude, EscApe is a 2D desktop game with both single player and multiplayer game modes. The player's aim is to escape from the ape as much as possible by avoiding hitting obstacles, such as low ceilings ,short walls and columns or falling into spaces. Scores are calculated depending on the amount of time the player survives. LibGDX will be used as the game library to generate sprites and textures and SQL and SQLite will be used as databases to store profiles' stats and possession.EscapeGameSinglePlayer is the class in which a single player game is created with all game objects that must be a piece of single player game. EscapeGameMultiPlayer class extends the EscapeGameSinglePlayer class to create multiplayer game with the additional character, gorilla and isOneMadeMistake( ). GameObject is the ab-

stract class from which every object(obstacle, coin, runners and ground) that are locatable on the game screen extends. PowerUps is the abstract super class of Wing, Shield and ExtraLife. Profile class stores one's nickname, number of coins and power-up information, and updates the number of coins and power-up information by using updateDataBase() and getDataBase().

References

[1] LibGDX. libGDX. (2021, September 1). Retrieved December 10, 2021, from https://libgdx.-com/.

[2] Sqlite Home Page. SQLite Home Page. (n.d). Retrieved December 10, 2021, from https://www.sqlite.org/.