uf_bird

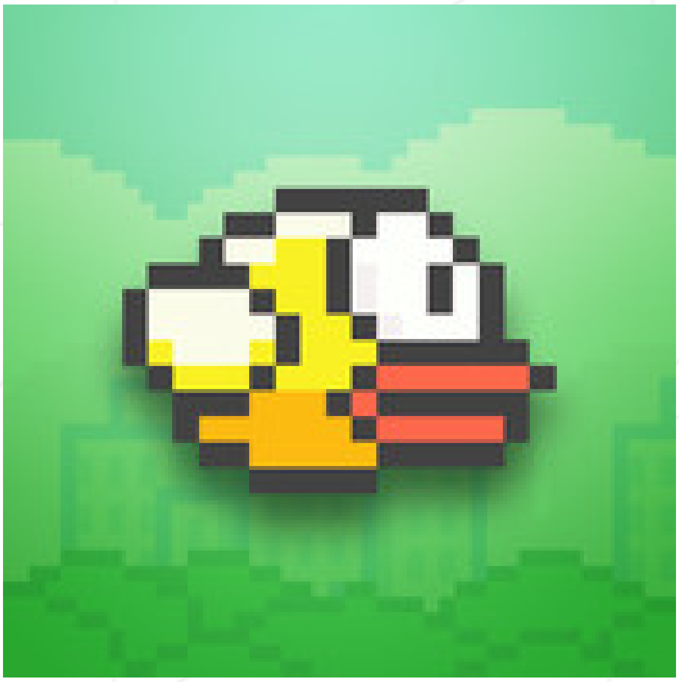
Game development

Mykyta Tupikov nikitatupikov22@gmail.com
UNIT Team born2code@unit.ua

Summary: This project is the introduction to the libGDX library and mobile game development for mobile devices.

Chapter I

Introduction



This is the first project in a series of projects with a focus on Java and libGDX framework.

The game is a side-scroller where the player controls a bird, attempting to fly between columns of green pipes without hitting them.

- You will use libGDX - a cross platform game development framework.
- libGDX allows us to simultaneously develop for iPhone, iPad, Android, Mac and PC, along with HTML, but for now our goal is to develop only for mobile platforms.
- You will use Java. New to Java? Learn it.

Chapter II

Goals

This project aims to make you familiar with:

- Game development

- Model-View-Controller pattern

- Graphics, textures, atlases, etc.

Chapter III

General instructions

- This project will be evaluated only by humans.
- This project must respect the Model-View-Controller pattern. This means project code cannot be placed in only one package, you must logically sort your classes to appropriate packages: model, view, controller.
- Using OOP as much as you can, so you, or someone else can understand and add something new to your project.
- You are allowed to use language features up to Java 8 included.
- You are are allowed to use any external libraries.
- You app must be named 'uf_bird'
- App must have unique icon.
- During your evaluation you can show your game on your Android/iOS device, or on emulator.

`Builder, Factory, Singleton patterns`

Chapter IV

The project

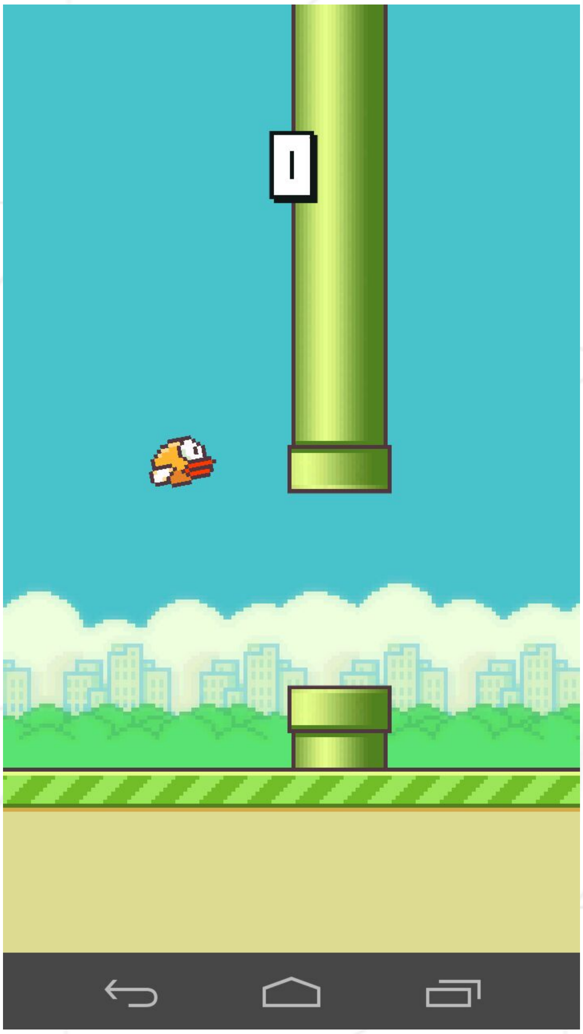
IV.1 Mandatory part

You need to implement a clone of famous Flappy Bird game based on the gameplay and conditions described below. The program needs to follow the Model-View-Controller architecture.

IV.2 Gameplay

- The aim of uf_bird is to navigate the bird through the pipes. The player scores one point for each set of pipes successfully passed. Player's score must be shown anywhere on the screen.

- Bird moves continuously to the right, between sets of pipes. If player touches the pipes, ground or screen border, he loses. Bird flaps upward each time player taps the screen; if the screen is not tapped, it falls because of gravity.



Each pair of pipes that bird navigates between earns the player a single point.

- There is no variation or evolution in gameplay throughout the game, as the pipes always have the same gap between them and there is no end to the running track.

- You must implement 'start' and 'game over' screens. On 'start' screen you can only tap to start the game. On 'game over' screen you must show how many points player earned through game and add button which leads to 'start' screen.

IV.3 Technical side

- For all game objects you **must** implement appropriate classes with the above behavior. So you can't put code of bird, background and pipes in one place. **Think!**

- Pipes must move from right side of screen to left. Each pipe from it's pair must be on the same x coordinate.

- There must be an animation for bird (flapping wings), and for ground (it must go from left to right, like pipes). For animations you **must** use appropriate animation class and texture atlases.

IV.4 Bonus

As long as the mandatory part rules and the general instructions are respected, you can add such bonuses:

- Add music and sounds to game.
- Menu where you can choose skins for your bird.
- Store you scores in DB and create a ’high score’ menu to show them. Also on ’game over’ screen show the maximum score earned.

Chapter V

Turn-in and peer-evaluation

Turn-in and peer-evaluation. Turn your work and author file in using your GiT repository, as usual. Only work present on your repository will be graded in defense.