# Connect-4 Game Design Doc

By Wenjin Ma

1. Introduction

The purpose of this software is to implement a cross-platform classic game – Connect 4 based on libgdx.

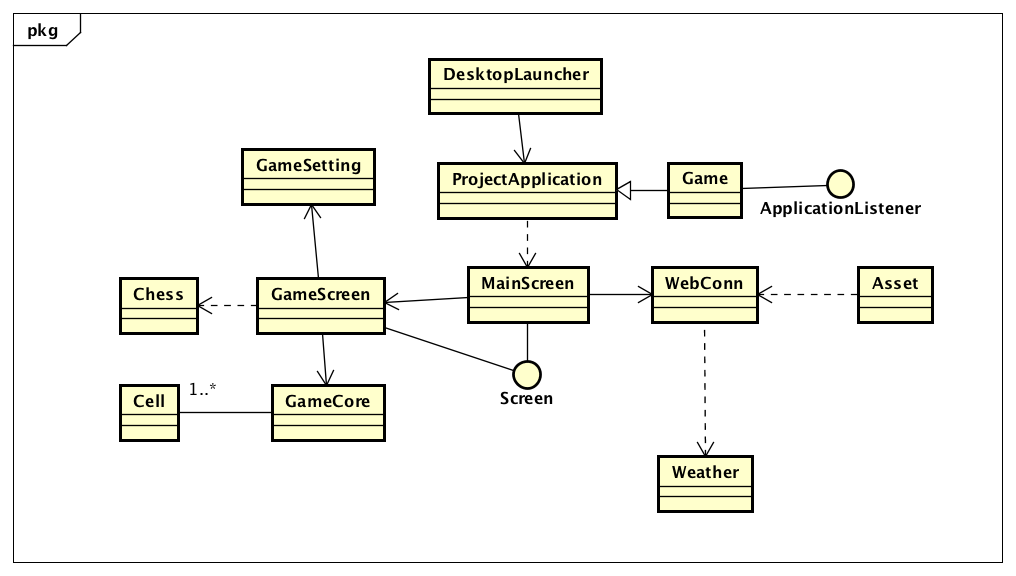
1. Features

* Show the current weather in San Francisco
* Play the Connect-4 Game

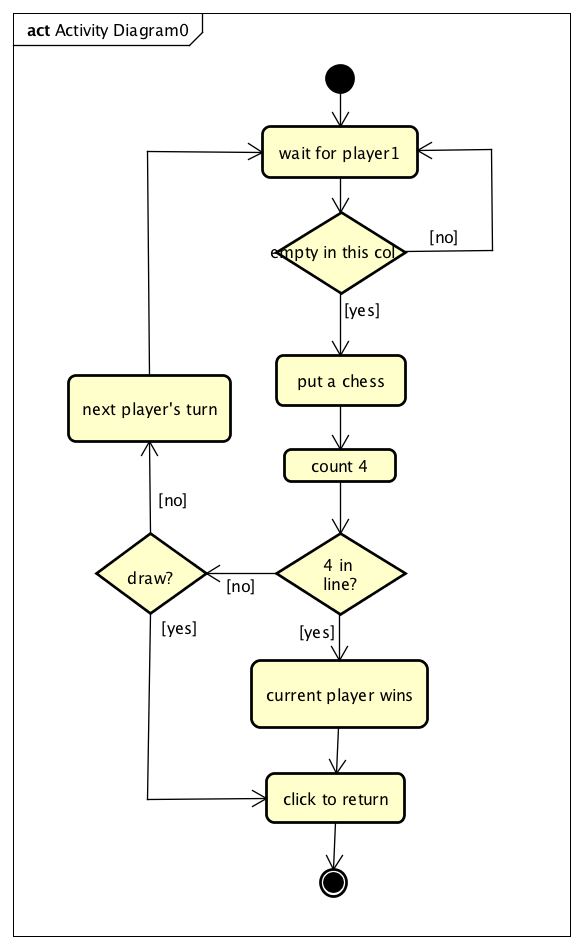
1. Architecture

This is a client-side android application based on libgdx library.

* 1. class diagram



* 1. activity diagram



* 1. decomposition description

Describe the core functions of each class.

* DesktopLauncher

Launch projectApplication class.

* ProjectApplication

Load the MainScreen, load all assets to assetmanager.

* WebConn

Web connection to the weather api, and fetch data.

* Assets

Create assetmanager, load assets from mainscreen and gamescreen.

* Weather

Save weather data given by webconn.

* MainScreen

Set up all mainscreen UIs, add all listeners.

* GameScreen

Set up game board, add all listeners, refresh board and screen based on GameCore.

* GameCore

Core function class when playing the game. Check playing turns, check playing state, check winning conditions.

* GameSetting

Initiate game board size, win conditions and all other game settings.

* Chess

The class to put on board when players click.

* Cell

The basic class to initiate the game board.

1. Component Design

This part, we take a closer look at what each component does in a functional way.

* ProjectApplication

1. Load the main screen.

screen = new MainScreen(game);

setScreen(screen);

1. Load assets

Assets.get().loadMainScreen();

Assets.get().loadGameScreen();

* MainScreen

1. Load the main screen.

manager = Assets.get().getManager();

1. Add all actors to main screen.

stage.addActor(sfxBtn);

stage.addActor(apiBtn);

stage.addActor(gameBtn);

stage.addActor(pennyLabel);

1. Fetch weather data from web api.

fetchWeather();

1. Jump to game screen if clicked the game button.

Screen gameScreen = new GameScreen(game, currentScreen);

currentScreen.hide();

game.setScreen(gameScreen);

* GameScreen

1. Load the game board.

setupBoard();

1. Add all actors.

stage.addActor(cell);

1. Add chess to cell slot.

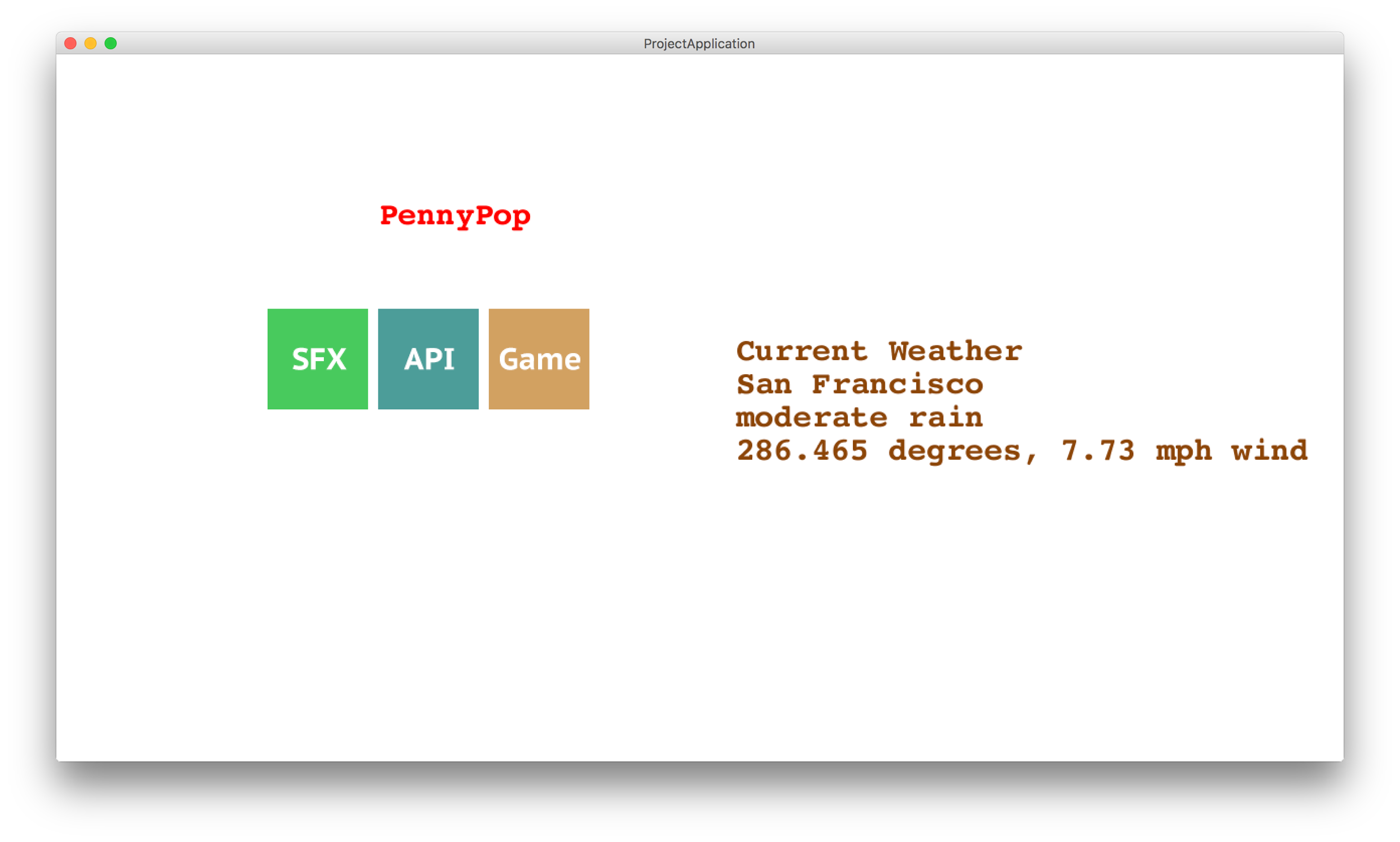
gamecore.addToSlot(gamecore.checkTurn(), chess.getRow(), chess.getCol());

1. Check winning state.

gamecore.getState() == GameCore.WIN

1. Interface

* Main screen



* Game screen

