

Real Estate Tycoon

Flexible City Sim Game

Requires **Unity 2020.3.30+**

Supports Android, iOS, WebGL, Windows and Mac

Dear Customer,

Thank you so much for purchasing this game kit. Here you can find the most important information on how to use this kit with maximum proficiency. All script and code assets are fully commented, but if you ever needed a hand on a block of code or anything else, feel free to contact us at <http://www.finalbossgame.com>. We'll try our best to support you with your questions as soon as possible and till you feel satisfied.

Overview

Real Estate Tycoon game kit is a flexible time-management game that engages the player with its simple and addictive game play. This kit provides 3 modes to play including Time-Trial, Endless and Career, each bringing different missions and game play to your users.

The kit uses a very flexible design and you can use it to build a 2D or 3D game at no time. You can also use the codes and classes to learn how to build a similar time-management tycoon game from scratch.

This is a full template, meaning that you have access to a complete game flow with menu, level selection, pause, game over and main game scenes.



This project accepts both touch and mouse inputs, and thus, can be tested on **Android, iOS, WebPlayer** and **Stand-Alone** platforms simultaneously. This kit works flawlessly on both Unity3d Free and Pro.

!!Important:

- When you load the kit for the first time, always add all scenes to the “Scenes in build” list via **file->BuildSettings**, to make sure you experience a smooth transition and game flow.
- The game requires these **Tags** to work:
 - buyableButton
 - globalGameController
 - cityCar
 - levelSelectionPin

Make sure to maintain them in your personalized game or replace them with proper equivalent.

- This kit uses a flat 3d design. All the objects and GUI elements you see in the kit are 3D cubes with their Y scale set to 0.001 to simulate the 2D feel of the game. We used this design structure, because it gives you full authority over the content in your own game. You can easily replace the cubes with your real 3D models, to make a 3D game, or you can replace the cubes with 2D sprites to transform the project to full 2D. There is absolutely no limitation of any type, and it's up to you to choose between 2D or 3D design.
- All game objects must start with a Y position value of 0.5. They have controllers that fix their Y positions according to their Z (depth). But always set all game items Y position to 0.5;
- If you want to create a new level in the career mode, simply clone of the available level-items and then modify its text label and missions via *CareerLevelSetup* component.
- If you want to use another map as the game background or you just want to define new positions for the building in the game, simply modify transform component of *_dummySpawnPositions* items in hierarchy. They are the dummy position helpers that *globalGameManager* use to place building.
- To ease the precise placement of buildings in the scene, we have placed 12 *inactive HouseHolder* in the scene. You can activate each one of them and move them aside to find a suitable clone position and then feed their transform values to the dummyPositionHelpers . Please take care to apply the positions to the correct tier dummies.

Game Play

In Real Estate Tycoon, you take the role of a real estate agent who has a certain amount of cash, and has to buy and sell buildings to make a profit. Your ultimate goal is to make as much money as possible in the shortest time frame.

Basically you have to be clever and buy the buildings when they are cheap, and sell them when they reach their price peaks. You also have access to multi tier buildings with customizable attributes like price range, length of availability in the market, number of price change cycles (price going up and down) , and you can also select different textures for each tier to spice up your game.



When a building appears for the first time, it has a price tag that shows the building's current price. If you have enough money and can afford its price, you can buy it. Otherwise you can let it be, because it might get cheaper in the next market wave.

When you purchase a building, you can sell it in a short amount of time, to make a profit, or you can wait for the building to reach to its peak price. That's when you make the most profit.

If you sell a building cheaper than the purchased price, you will lose money.

The current kit features 3 play modes:

Endless: You can play as long as you like. You can beat the optional goal and the game saves your highest score (Money) for you.

Time-Trial: You have a limited time to make a fortune and reach to the indicated goal. You have to think fast, purchase several buildings at once and have strategy to beat this mode. It also saves your best high score (Time) for you.

Career: In career mode, you have access to several levels that ask for different objectives. You can customize the tier availability for each level, starting balance, target time & balance settings and more...

Introduction to Scripts, Classes and GameObjects

All scripts in this kit are fully commented and have short description about what they do. But here we take a look at some classes used in the kit.

globalGameController

This is the most important class of the kit. GlobalGameManager is responsible for all events inside the main game loop, like controlling the time, balance, mission, cloning new buildings, monitoring status of the game and many more. This class also initializes the game scene based on user selection in menu & career scenes. It changes the game mode and set the appropriate variables to meet the requirements.

GlobalGameController reads most of the settings from playerPrefs. Later you will see how we use **CareerMapManager** to save the user settings into the playerPrefs.

This class also clones new buildings based on the conditions given to it, and then initialize new buildings with different random variables for a smooth and stochastic game play experience.

HouseController

This is the main controller for all the buildings in the game. It is responsible for the enter/exit animation, Price going up and down, saving the buy/sell states and also available loop cycles for the buildings price.

globalTouchManager

This class handles all user interactions (Touch/tap/click) on the buildings in the game. It is also responsible for setting the correct flags when user buys or sells a building.

FixPositionOnAllResolutions

Attach this class to any 2D or 3D object and it will always position itself on a safe spot and never leaves the view.

CareerLevelSetup

This is a simple controller which indicates the mission items for each level. You can simple set the values of the variables within the inspector of each level-button in hierarchy and they will be processed automatically by the **CareerMapManager** class.

CareerMapManager

This class saves the user selected settings into playerprefs to make them available for the **globalGameController** to interpret.