Fruit Juice Maker

Unity 3d Complete Game Kit / Ready to Release

Requires **Unity V5.3.4+**
Supports Android, iOS, WebPlayer, 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 project efficiently. 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 www.finalbossgame.com . We'll try our best to support you with your questions as soon as possible.

Overview

Did you ever dream of owning and managing a small fruit juice shop? Do you like to serve your customers with delicious juices and smoothies? Do you want to make a new game of your own which you can add new customers, add new juices each with their own recipes and creating new levels each with different missions and properties? Well, Fruit Juice Maker game kit is an easy to use and learn project that can greatly speed up your prototypes of the same type, and can be used as a strong base for further development and experiments. It has all nuts and bolts to build a fully functional game. It came with:

- **8** premade customers (with different textures and patience level)
- **12** predefined ingredients
- **14** predefined products (based on the available ingredients)
- Unlimited number of Blender machines
- Candies to relax customers and refill their patience levels.
- Full game flow with Menu, Shop, IAB, Rate-Us, Pause, Level Selection and Game scenes.

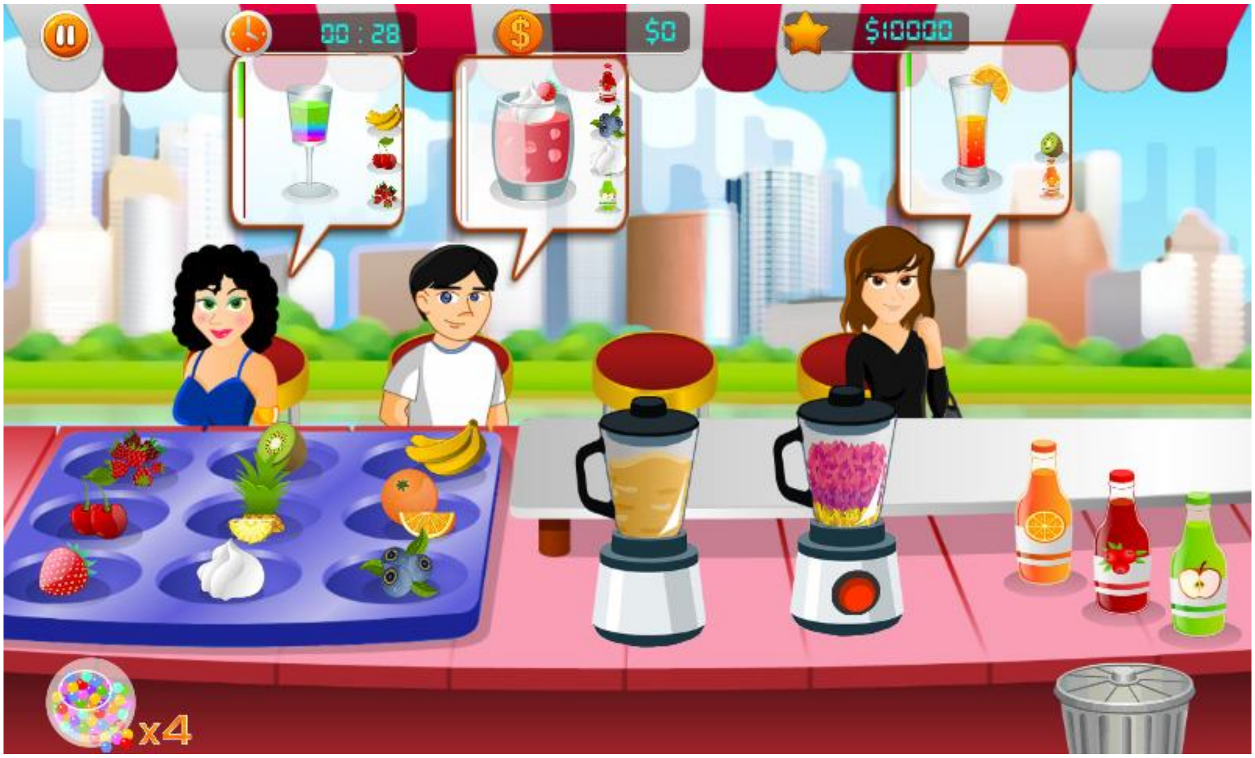
However, you are not limited by these numbers and can easily define and add more items to the game to deliver a rich game play experience to your players.

The game accepts both touch and mouse inputs, and thus, can be tested on **Android**, **iOS**, **WebPlayer** and **Stand-Alone** platforms.

This game kit needs no 3rd party plug-ins to works. It runs and builds out of the box. All you need to do is to load the kit inside Unity, set the project on the desired platform and hit "Build" to receive your game in no time!

Game Play

In this game, you are the manager of a small fruit juice shop. There are customers which show up in your store and ask for a certain juice. The ingredient of that specific juice will be shown in small icons beside the main request and you have to drag those ingredients from inventory to a blender machine and hit the blend button. Blender will start to blend the ingredients and delivers the final product within a few seconds. If the recipe is right, blender will transform into a glass of selected juice, and if the recipe is wrong, the end product will be a crap juice which should be discarded into the recycle bin.



In each mission, you have to serve as many customers as possible to collect enough money to reach the target. You will unlock the next level upon completion of the previous ones.

Your customers can get board if they wait too much for their orders. You can use candies to reset their patience level to the default settings. You can buy additional candies inside the in-game shop.

Classes

This Starter kit uses a few separate classes to control the game's main routine. All these classes are fully commented and you can easily guess the dataflow. But I try to introduce them here also:

- **MainGameController:** This is the main controller of the game. It holds all static variables, manages game time settings, money, score, and also is in charge of random customer creation. This class will also manage game's start/finish/completed/failed states.

- **CustomerController:** This is the main customer controller. It controls customer events, animations and is responsible for time (patience) management of each customer. It also determines each customer's wish and its ingredients and then shows them in a bubble over customer's head. It can also receive delivery from other classes and check them to see if they match the given order or not and then respond accordingly.

- **ProductManager:** This simple class is the heart of the game. Here you can define and edit products and set their special attributes. Each product has Price and Ingredients and in this class, you can easily define these values. We show you how to add new products, later on.

- **IngredientsController:** This class controls the behavior of available ingredients on screen, if they can be picked and dragged into blender machines, and also manages their movements.

- **BlenderController:** This class is responsible for all things related to blending/mixing and delivery Queues. When a customer asks for a juice, you need to drag and drop the required ingredients into a blender and mix them in order to make the requested juice. The blender will transforms into a new glass of juice when the blending is over. You need to deliver the juice to the correct customer to receive your money. We have 2 blender machines inside the game kit, but there is no limitation on how many blenders can be inside a level.

- **ProductMover:** This class will manages products movements (from ingredients panel to blender machines) and let player drag and move them with mouse or touch. It also checks if the products are dragged over blender machines and if so, add their ID to that blender delivery queue. If player pick a product and release it without landing it over something interactive, this class will automatically destroy that instantiated game object.

- **TrashBinController:** This class is responsible for receiving and destroying wrong deliveries. Player can move their undesired and wrongly made products to this Object to discard them.

- **CandyController:** Candies are used to entertain bored customers and refill their patience bars. This class manages the drag, drop and message handling of this routine.

How to add new Products, Ingredients, Customers & Levels?

To add a new customer to the game:

1. Pick a customer prefab from folder "Prefabs/Customers" and duplicate it with Ctrl+D;
2. Rename the new customer prefab to something appropriate.
3. In the inspector, set the desired **customerPatience** (in seconds) and their specific needs via **customerNeeds** (ID : int). Eventually you can leave the customerNeeds at 0 to let the controller pick a random product from products Array everytime.
4. You can set to show/hide customers wish list ingredients by setting/unsetting its value from inspector.
5. Set the available products for this customer by carefully choosing products within **AvailableProducts** array.
6. Set the available ingredients for the selected products (in step 5) via **AvailableIngredients** array. Please note that if you add a product to **AvailableProducts** array, and forget to add its required ingredients to **AvailableIngredients**, it leads to an error.
7. Define new material and texture for this new customer and set them via prefab's inspector.
8. Go to GameController object in the scene and update its **customers** Array with your newly created customer prefabs.
9. Repeat these steps to add as many customers as you like.
10. You are done!

To add a new product:

1. Pick a product prefab from "Prefabs/Products" and duplicate it with Ctrl+D;
2. Rename the new product prefab to something appropriate.
3. From the inspector, set the **Price** (Int), **TotalIngredients** (int, range from 1 to 6) and **ingredientsIDs** (Int, with index from **1** to **TotalNumberOfAvailableIngredients [currently 12]**)
4. For example, the definition for a new product is like this:
   a. Price = 150
   b. TotalIngredients = 4
   c. IngredientsIDs: [1, 3, 8, 10]
5. Define new material and texture for this new product and set them via prefab's inspector.
6. Select all customer prefabs in "Prefabs/Customers" and update their AvailableProducts array with the newly created product. This will enabled them to be able to select this new product as their wish list.
7. Repeat above steps to add as many products as you like.
8. You are done!

To add a new normal ingredient:

1. Pick a normal ingredient prefab from “Prefabs/Ingredients” and duplicate it with Ctrl+D;
2. Rename the new ingredient prefab to something appropriate.
3. Define new material and texture for this new ingredient and set them via prefab’s inspector.
4. Do not edit prefab’s FactoryID. (Must be left at 0).
5. Do not touch other public variables in inspector. Leave them at their default state.
6. We are done with Prefab.
7. Go to Hierarchy and select **IngredientsHolder** game object. It contains 12 ingredients already.
8. Duplicate one of the available **Ingredient-xx** child objects and position it properly in the editor.
9. Set the newly created **ingredient-xx** child object’s Factory ID to an incremented size (13, 14, 15 , … and so on).
10. Select all children **ingredient-xx** objects (from hierarchy) and update their Ingredients array with the newly created ingredient prefab from step 1.
11. Select all customer prefabs in “prefabs/Customers” and update their **availableIngredients** array with the newly created prefab from step 1.
12. You are done!

To add a new Level to Level-Selection scene (extending the career mode):

1. Select and clone one “Level-Item” button from the available buttons on “Level-Selection” scene.
2. Rename this cloned button to something appropriate like “Level-9”…
3. Select the button and configure it’s class properties (Career-Level-Setup) via inspector:
    a. **LevelID**: index of the level we want to load. Must be unique.
    b. **AvailableBlender**: How many blender machine we need for the level.
    c. **LevelPrize**: Money that will be given to player after beating the level.
    d. **Goal Balance**: mission of the level to beat.
    e. **AvailableTime**: time given to player to beat the level mission money. (in seconds)
    f. **CanUseCandy**: set if player is allowed to use candy to refill customer’s patience bars.
    g. **AvailableProducts**: array containing available indexes for products that are available in this level. Customers will always check their wishes with this array and choose from available products you carefully select for each level.

***Note**:

You can select a few simpler products for the starting levels and then increase the difficulty of your game by choosing complex products (consisting of many ingredients) in larger quantities. Look at the default configuration for levels 1 to 8 to get familiar with the setup.

How to setup Admob IDs and settings:

1. Open “Intro” scene.
2. Select “AdManager” game object in the hierarchy and set your own Admob IDs into the respected fields.
3. You are all set. Check the game on your Android device to see if everything is working properly.

Final Word

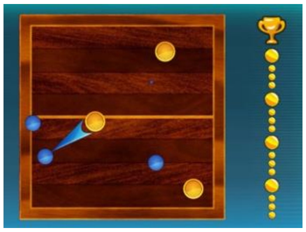
If you have any questions, feel free to ask us at http://www.finalbossgame.com and we will get back to you as soon as possible.

Our Other Cool Game Kits

Frenzy Farming Game Kit
Unity Assetstore

SKY BURGER
Unique & Highly addictive game-play
Compatible with Android, iOS and Standalone
Unlimited number of levels & Missions
AdMob banner, Interstitial & Video Ads
Quick Rate + Share widgets included!
Mobile Friendly
Very easy to Re-Skin
Runs out of the box!

Sky Burger Game Kit
Unity Assetstore



Turn Based Ball Fight Game Kit
Unity Assetstore

Swing Game template
Unity Assetstore

Stealth Action Game Kit
Unity Assetstore

Restaurant & Cooking Starter Kit
Unity Assetstore

Endless Space Pilot Game Kit
Unity Assetstore

Snakes & Ladders Framework
Unity Assetstore

Finger Soccer Game Kit
Unity Assetstore

Monster Blaster! Game Kit
Unity Assetstore

Real Estate Tycoon Game Kit
Unity Assetstore

Head Soccer game kit
Unity Assetstore