



Automachef Mod Creation Guide

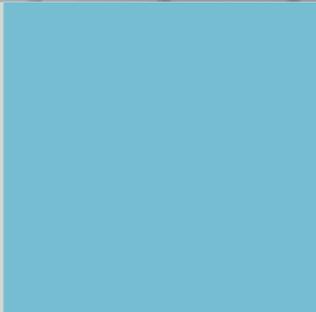
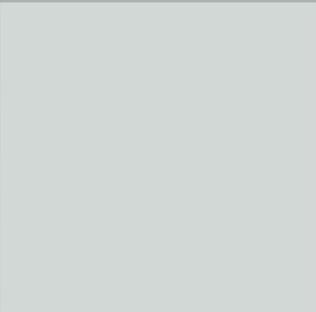
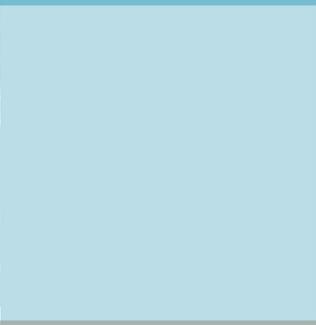


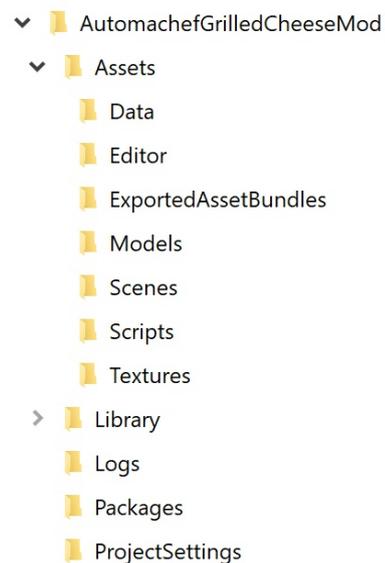
TABLE OF CONTENTS

Initial Setup.....i	Assembling the Mod.....viii
Mod Structure.....ii	Adding the Mod to the Game.....x
Creating Content.....ii	Limits.....x
- Creating a Level.....ii	Appendix I: List of Internal Names of Machines.....xi
- Creating Special Events.....iv	Appendix II: List of Internal Names of Ingredients.....xiii
- Creating Ingredients....v	Appendix III: List of Internal Names of Liquid Ingredients.....xv
- Creating liquid Ingredients.....vii	Appendix IV: List of Internal Names of Dishes.....xvi
- Creating Dishes.....vii	

INITIAL SETUP

- Download the AutomachefSDK.
- Create a new Unity Project using Unity version 2018.3.7f1. Other Unity versions might work, but that is not guaranteed.
- Copy and paste the Assets folder from the AutomachefSDK into your new project. Make sure you do not end up with a folder called Assets inside the Assets folder of your new Unity project. You should end up with YourProject/Assets/Data, YourProject/Assets/Models, etc.
- The AutomachefSDK project includes several assets used as an example on how to set up things properly, including a Dish, several Ingredients, a Liquid Ingredient and a few Levels. You should delete them if you don't plan to use them as part of your mod. This includes everything in the Assets/Data, Assets/Models and Assets/Textures folders. Maybe you could keep the model with the empty plate (Assets/Models/Plate.fbx) as a starting point for your own dishes so they look the same as those included in the base game.
- Do not delete the Data or ExportedAssetBundles folders inside the Assets folder. You can delete their contents but keep the folders themselves.

- This is how the folder structure should look like after merging the contents of the SDK into your new Unity project. Assuming your project is called AutomachefGrilledCheeseMod, of course. You can create other things besides Grilled Cheese mods, I'm sure!



MOD STRUCTURE

Automachef mods can contain the following assets:

- **Levels** (all mods should contain at least one level in order to be playable). These are Scriptable Objects from the UGCLevel class.
- **Special Events**, such as rush hours, food critic visits and dish crazes. These are Scriptable Objects from the UGCSpecialEvent class. They are referenced in Levels.
- **Ingredients, Dishes and Liquid Ingredients**. All of these are MonoBehaviours that should be added to your own custom GameObjects.
- Every ingredient and every dish requires a unique "internal name". This is not the name that is displayed to players, it is an internal string that identifies each one of them. It also should be different from any internal names for ingredients or dishes that are already included in the base game.

CREATING CONTENT

CREATING A LEVEL

The first thing you should do is creating a Level so your mod can be played. In order to do this you should go to the Automachef SDK menu and select "Create New Level". This will instance a new Level Scriptable Object in your Assets/Data folder. You should then open it and use the Inspector to set the following attributes:

- **Friendly Name**: The title that will be displayed to players when selecting or playing the level.

- **Number of Dishes**: The total number of dishes players have to deliver to customers in order to win.
- **Required Dishes**: A list of the internal names of the dishes that customers might order.
- **Side Dishes**: A list of the internal names of the dishes that customers might order as side dishes.
- **Funds Limitation**: The maximum budget available to players to purchase the machines to complete the level.
- **Power Limitation**: The maximum energy in watt-hours that players can use to complete the level.
- **Power Usage Limitation**: The maximum power in watts that the kitchen machines can draw before a blackout occurs. Set to 0 if you do not want a limit.
- **Ingredients Count Limitation**: The maximum number of ingredients that players can use to complete the level.
- **Min Time Between Orders**: The minimum number of seconds that will pass between customers.
- **Max Time Between Orders**: The maximum number of seconds that will pass between customers.
- **Side Dish Probability**: The probability that a customer will order a side dish with the main course. Between 0 (0%) and 1 (100%). Setting it to 0.5, for example, means 50% and so on.
- **Environment**: The environment in which the level is set. Possible values are:
 - City Small
 - City Medium
 - City Large
 - City Large With Drive Thru
 - Beach Small
 - Beach Large
 - Airstrip Small
 - Airstrip Large

- Airport
- Airport With Takeout
- Skyscraper
- Skyscraper With Takeout
- Concert
- Park

- **Special Events**: A list of UGCSpecialEvent Scriptable Objects that will occur during the execution of the level. Do not include more than one Special Event of each type.

- **Starting Time**: The time in the clock when the level starts running. It's important to set this properly for Special Events to work. This number should be in 24 hours format and with minutes expressed using decimals. So for example 11:15 a.m. would be 11.25 and 3:30 p.m. would be 15.5.

- **Available Machines**: A list of internal names of machines that will be available to players when playing the level. It defaults to all machines so you can then delete those that you don't want.

- **Has Fire Disaster**: Can fires start in the kitchen?

- **Has Insects Disaster**: Can insect infestations start in the kitchen?

- **Has Breakdowns**: Can machines break down?

CREATING SPECIAL EVENTS

You can add special events to your levels by going to the Automachef SDK menu and selecting the "Create Special Event" option. Once the Scriptable Object has been created you need to configure it using the Inspector. They contain the following properties:

- **Type**: What type of special event are you setting up? Possible values are:
 - Rush Hour
 - Food Critic

- Craze

- **Start Time**: At what time the event starts. Remember to set the Starting Time property in the level properly!
- **End Time**: At what time the event ends. This only affects Rush Hour and Craze events, as the Food Critic is a one time only kind of event.
- **Orders Multiplier**: The factor of multiplication that will be used for the Rush Hour event. For example, using 2 as the value will make orders come at twice the normal pace, using 3 will make them come three times as frequently and so on. This does not affect the Craze and Food Critic event types.
- **Order To Spawn**: The internal name of the Dish that will be ordered by the Food Critic or that will be ordered by all customers during a Craze. This does not affect the Rush Hour event type.

After configuring the Special Event you need to add it to any levels you want by opening them in the Inspector again and using drag-and-drop to add the Special Event to the Special Events List in the Level Scriptable Object.

CREATING INGREDIENTS

The process for creating new ingredients and dishes is slightly different from creating levels, as they are not Scriptable Objects but MonoBehaviours that need to be attached to a GameObject. First of all you should create the 3D model for the ingredient and import it into Unity. Afterwards you can drag the 3D model into the scene, select it, click on Add Component on the Inspector and add the UGCIngredient script to it. After doing this you will need to configure the following fields:

- **Internal Name**: This should be a unique string that identifies this ingredient. It cannot be the same as any other internal name used in the mod or in the base game. Please check the Appendix to get a list of all names that are already in use in the base game.
- **Friendly Name**: The name that will be displayed to players in the game.

- **Icon**: A square texture with the icon that will be used for this ingredient in the game UI.
- **Technique**: How this ingredient comes into existence. Possible values are:
 - **Assembling**: This ingredient is created in an Assembler, like for example the Breaded Chicken Pieces in the base game.
 - **Packaging**: This ingredient is created in a Packaging Machine, like for example the Fries in the base game.
 - **Dispensing**: This is an ingredient that can be obtained from Dispensers.
 - **Mixing**: Only for liquid ingredients!
 - **Other**: This ingredient comes from being fried, grilled, processed or is otherwise derived from another ingredient.
 - **Storage Tank**: Only for liquid ingredients!
- **Result Grill**: The internal name of what players get when grilling this ingredient.
- **Result Process**: The internal name of what players get when putting this ingredient into a Food Processor.
- **Result Fry**: The internal name of what players get when frying this ingredient.
- **Result Wet**: The internal name of what players get when this ingredient is wet by a Sprinkler.
- **Result Bake**: The internal name of what players get when this ingredient is put into an Oven and baked.
- **Result Steam**: The internal name of what players get when this ingredient is put into a Food Steamer and steamed.
- **Recipe**: A list of internal names of ingredients used to manufacture this ingredient if the Assembling or Packaging techniques are selected. In order to require more than one ingredient simply write

it more than once on the list. For example, the list for a Double Bacon Burger in the base game would be: BurgerBun, Patty, Patty, CheeseSlice, CheeseSlice, CrispyBaconStrips.

- **Recipe Liquid Ingredients**: A list of internal names of all liquid ingredients used to manufacture this ingredient if the Assembling technique is selected. Also used when defining liquid ingredients that come from Mixing.

- **Liquid Ingredients Quantities**: This list should have exactly the same length as the Recipe Liquid Ingredients list. The first item of this list corresponds to the amount of milliliters of the first item of the other list used to make this ingredient, the second one to the second one and so on.

- **Time to be Assembled**: The number of seconds that a regular Assembler in normal operation mode would take to assemble this ingredient if Technique was set to Assembling. Ignored otherwise.

CREATING LIQUID INGREDIENTS

The process for Liquid Ingredients is exactly the same as for regular Ingredients, although these do not require a 3D model. You should add the UGCLiquidIngredient component to an empty GameObject instead. Their Technique field should always be set to either Storage Tank or Mixing. They also require an additional field called "Color" that should be configured to the color this ingredient will display when in Storage Tanks and Liquid Mixers.

CREATING DISHES

The process for Dishes is exactly the same as for regular Ingredients but using the UGCdish component instead of the UGCIngredient one. They have three additional fields that need to be configured:

- **Time to be Spoiled**: How many seconds before this dish goes bad after being produced.

- **Expected Ingredients**: The total number of ingredients that is required to produce one unit of this dish. This is used in kitchen efficiency calculations.

- **Expected Power**: The total amount of energy in watt-hours that is expected to be used to produce one unit of this dish. This is used in kitchen efficiency calculations. For reference and as a starting point here are some values used in the base game:

- Plain Burger: 29
- Cheesy Dog: 36
- Fried Chicken: 48
- Double Bacon Burger: 57
- Fancy Salad: 68
- Extreme Burger: 91

ASSEMBLING THE MOD

After creating all of the necessary GameObjects for Ingredients, Dishes and Liquid Ingredients remember to drag them into the Data folder so they can be converted into prefabs. The next step is to go through all of your assets (scriptable objects, game objects, dishes, ingredients, 3D models, textures, materials) and ensure that they are configured to use the asset bundle called "mod" by selecting it with the dropdown in the bottom right corner of the screen. Please refer to the following screenshot:



Once you are satisfied with what you've created make a final inspection check to:

- Check that all of your materials, models, textures, levels, special events, ingredients, liquid ingredients and dishes are set to use the Asset Bundle called "mod".
- Double check the Internal Names of all ingredients and dishes to ensure there are no duplicates and that none of them are in use by the base game.
- Make sure that all ingredients and dishes have a Friendly Name and an icon set. Also verify that the results of different processes such as frying and grilling point to existing Internal Names, either within your mod or in the base game.
- Remember that all mods should contain at least one level in order to be playable.
- Verify that all references to dishes in levels and to ingredients in recipes point to Internal Names that actually exist, either within your mod or in the base game.
- Make sure that all dishes have their manufacturing technique set to Assembling or Packaging and that they have a recipe with at least one ingredient. Also verify that your recipes do not contain any cycles (like a dish that requires itself as part of the recipe) or trees that are too deep, as that might cause the recipes to not be displayed properly in game.
- Take a look at the limits explained in the "Limits" section of this document so you are sure you are not exceeding any of them.

The final step is to go to the Automachef SDK menu and selecting the "Build Mod" option, which will export a single file with .automachefmod extension to wherever you decide to save it on your filesystem.

ADDING THE MOD TO THE GAME

Adding the mod so it can be played in the game is very simple. You simply need to copy your file with the .automachefmod extension into the folder that contains your saved games. This will be in %appdata%\..\LocalLow\HermesInteractive\Automachef for Windows systems. In there you should open the Saves folder, then go into the folder that is called after your Steam ID (you will probably find only that one unless you are sharing your Windows user account with people who use other Steam accounts), then create a new folder there called "mods" if there is none. You can then paste your .automachefmod file into the "mods" folder. Now you are ready to launch the game and give your creation a go!

LIMITS

The following limits are in place in order to prevent mods from causing UI or performance issues:

- Friendly Names in Ingredients, Dishes and Levels are limited to 30 characters.
- The number of Levels in a mod is limited to 10.
- The number of Ingredients and Dishes in a mod is limited to 50.
- The number of Ingredients in a recipe is limited to 10.
- Dishes should only have their Technique set to "Assembling" or "Packaging" and they should always have at least one ingredient in their recipe.

APPENDIX I

LIST OF INTERNAL NAMES OF MACHINES

- AdvancedAssembler
- AdvancedComputer
- AdvancedOrderReader
- Assembler
- Belt
- BeltBridge
- Computer
- ConvectionFryer
- ConveyorGrill
- Counter
- Dispenser
- DumbRobotArm
- ExtinguisherArm
- Fryer
- Gate
- Grill
- HighSpeedBelt
- HighSpeedDispenser
- LargeStorageTank
- LargeStorageUnit
- LiquidMixer
- LongRobotArm
- MechanicMechanic
- OrderReader
- Oven
- PackagingMachine
- Processor
- Pump
- Repeater
- RobotArm

- SmartSplitter
- Splitter
- Sprinkler
- StackingRobotArm
- Steamer
- StorageTank
- StorageUnit
- Teleporter
- XTerminator

APPENDIX II

LIST OF INTERNAL NAMES OF INGREDIENTS

- AshPile
- BaconStrips
- Breadcrumbs
- BreadedChickenPieces
- BreadSlice
- BurgerBun
- Carrot
- CarrotSlices
- Cheese
- CheeseSlice
- ChoppedMushroom
- CNBellPepper
- CNChoppedBellPepper
- CNCookedPork
- CNDriedChili
- CNDumplingFilling
- CNDumplings
- CNDumplingsUncooked
- CNDumplingWrapper
- CNGinger
- CNGratedGinger
- CNGroundPork
- CNPork
- CNRice
- CNSteamedRice
- CookedHotDog
- CookedMincedMeat
- CrispyBaconStrips
- Croutons
- Egg
- EggContents
- FriedBreadedChickenPieces
- FriedEgg
- FriedPotatoWedges
- GratedCheese
- GrilledChicken

- GrilledChickenPieces
- GrilledMushroom
- HotDogBun
- Insects1
- Insects2
- Insects3
- Lettuce
- MincedMeat
- Mushroom
- Patty
- Potato
- PotatoPuree
- PotatoWedges
- RawChicken
- RawChickenPieces
- RawHotDog
- RawPatty
- ShreddedCarrot
- ShreddedLettuce
- TGButter
- TGChoppedOnion
- TGCondensedMilk
- TGCornbreadUncooked
- TGCornmeal
- TGFreshHerbs
- TGO onion
- TGPieCrust
- TGPumpkinPieUncooked
- TGPumpkinPuree
- TGRawTurkey
- TGRoastedPumpkin
- TGRoastedPumpkinPuree
- TGRoastedSweetPotatoes
- TGRoastTurkeyUncooked
- TGSweetPotato
- TGSweetPotatoPuree
- TGSweetPotatoWedges
- TGSweetPumpkin
- TGTurkeyStuffing
- ToastedBreadSlice
- Tomato
- TomatoPuree
- TomatoSlice
- WetGoop

APPENDIX III

LIST OF INTERNAL NAMES OF LIQUID INGREDIENTS

- BBQSauce
- CNChickenStock
- CNSoySauce
- CNWater
- Honey
- HoneyMustardDressing
- HotSauce
- Ketchup
- Mayo
- Mustard
- SaladDressing
- SpicyBBQSauce
- SpicyMayo
- TGMilk

APPENDIX IV

LIST OF INTERNAL NAMES OF DISHES

- BaconFries
- BeachBurger
- BigSalad
- BLT
- CheeseBurger
- CheesyDog
- ChickenMelt
- ChickenSalad
- CNDumplingsServing
- CNKungPaoChicken
- CNSteamedRiceServing
- DoubleBaconBurger
- ExtremeBurger
- FancySalad
- FriedChicken
- Fries
- HotWings
- LoadedFries
- PlainBurger
- PlainHotDog
- SideSalad
- SpicyBLT
- TexasBurger
- TGCornbread
- TGPumpkinPie
- TGRoastTurkey
- TGSweetPotatoesSide
- UltimateDog