# Hypercasual Mobile Games

A bunch of “hypercasual” mobile game ideas. To perhaps give Yishay an incentive to start programming, otherwise to generate some more ad revenue (and pad my portfolio).

The prerequisites:

* One button/One control/One mechanic => preferably not even text or tutorial needed to explain
* If possible, minimalist art style and stuff.
* If possible, don’t even add a menu or secondary type of screen
* As many opportunities to show ads and keep people playing as possible.
* Make the *idea* so solid that it doesn’t *need* any work/images/etcetera.

**IDEA (cookie cutter):** you are constantly presented specific cookies + a recipe/request => how many parts and what they want. You must slice across the screen to cut it precisely the way people want.

* Use my slicer code for that. But *really clean it up*.
* How to keep textures constant? **Simply keep the whole texture! But *mask/remove* whatever isn’t inside that part anymore.**
* (And the bodies can be resliced anyway.)
* How to generate random orders? We let the *computer* pre-slice it, then calculate the result of that

**IDEA (conveyor pusher):** stuff goes by on a conveyor belt (automatically, of course). There are several “pusher machines”: tapping them just extends their arms and will *push* anything they hit off the conveyor belt. Use that to deliver the items to their destinations.

**IDEA (double duty):** you do two runs:

* One where you are just playing some character and try to get as far as possible.
* Then, it *replays what you just did*, and it’s your job to stop/kill your old self as soon as possible.

There must be an incentive to do the first run as good as possible. Perhaps, you *only* get good traps/actions in the second run, if you grab them during the first? (And the score is higher.)

In the second run, you need incentive to really kill yourself soon. (The later you are, the less your score is multiplied? Or your score counts *down*?)

The challenge is of course: remembering how you moved in certain sections. And using whatever you have, in whatever order it’s given, to stop that.

**IDEA (stacking):** something with stacking. There is one (or several) piles on screen. Tap at the top to add something to a pile. Tap the pile itself to remove the bottom item.

* The item could always be the same. Or there could be a “current item”, like with tetris.
* Removing something might also have specific consequences. (That becomes the next item. Or special cubes have some “on\_destroy” action?)

**IDEA (the cleaners)**: the field starts with lots of dirty spots. You can do something to clean them (swipe, draw a line, tap to active some machine, slide a broom across). Objective? Clean the whole field.

* Yes, you can tap a certain *row* or *column*. A broom will slide across it and clean everything.
* Of course, there are different *brooms*, and there are things in the field that *stop them*. (Actual solid walls, special teleporters or deflectors, etc.)

**IDEA (reverse pong):** the paddles move on their own. You control the ball. Keep it in play as long as possible.

**IDEA ():** you’re trying to get the apple to fall on Newton’s head. You have a tree full of apples. Tap one to make it come loose. Some are worth more points, or do different things, or whatever.

* Obviously, Newton moves around.
* And there are other people who do *other* things when an apple hits them. (Which you might want, or *not* want.)
* To keep going longer, you’ll have to *regrow* apples as the game goes on and time ticks by. (Apples that haven’t been used for a while will *rot*, or *fall down by themselves*.)
* Ad opportunities: extra life to continue, different apple types to unlock, different arenas/people to unlock

**IDEA ():** you’re a hacker trying to crack codes. At the bottom of your screen is a *lock* or *password bar*. Click on a cell to change it to something else. Find the correct code within X seconds.

* The correct code is obviously randomly generated and follows some *pattern/ruleset* you learn over time.
* Maybe it’s a network of things. Out of those *cells*, lines appear, which do stuff across the map. The password is the thing that provides the correct *input* to make the network “unlock”.
* This is relatively easy to generate. Provide an endpoint, then work *backwards*, adding more modules as you go. Wherever you end, that’s the password.

**IDEA (reverse soccer):** People are standing around the field. Randomly they’ll extend their legs, or move, or whatever. You move the *ball*! Move it so that the right team scores.

**IDEA (medium):** You’re a frog sitting on lilypads, on a river, constantly moving. You just need to stay alive and not fall in the water. Tap some place to jump there.