



5 Minutes with the MacOS / iOS Zone Allocator

WhiskeyCon Singapore, March, 2017



Who am I?

- Stefan Esser
- from Germany
- in Information Security since 1998
- SektionEins GmbH from (2007 - 2016)
- AntidOte UG (2013 - now)



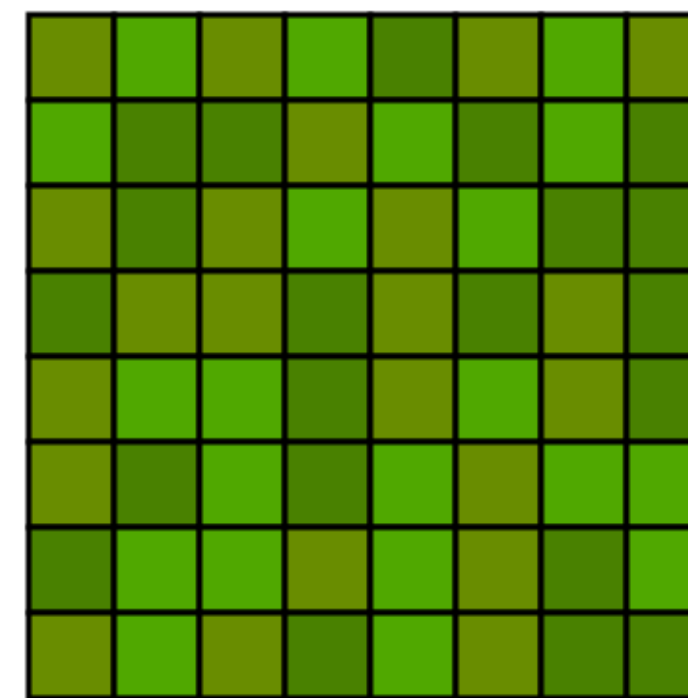
What is this talk about?

- between iOS 9.0 and iOS 9.1 the Zone Allocator had a bug
- it was fixed later due to refactoring of `zcram()`
- bug is not a security bug but influences heap layout
- might cause trouble for previously working heap-feng-shui code

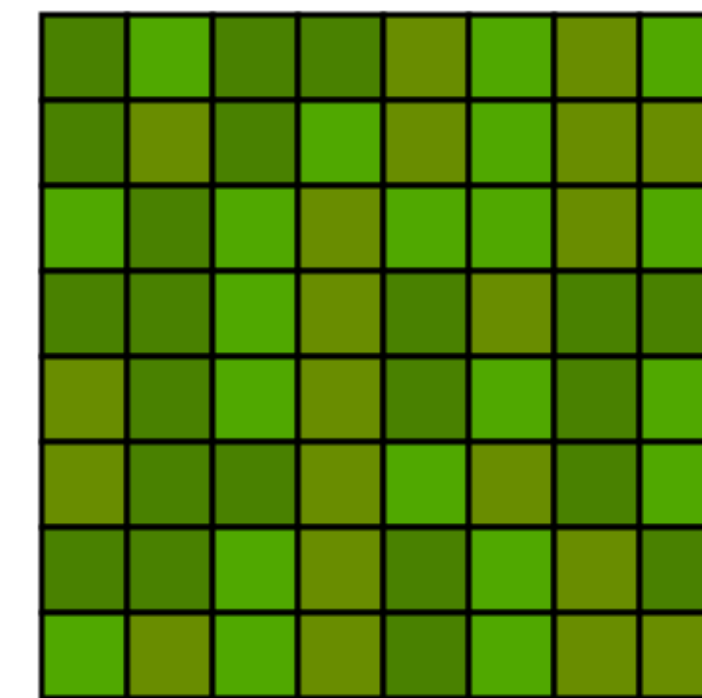


iOS Zone Allocator Allocations in iOS <= 6

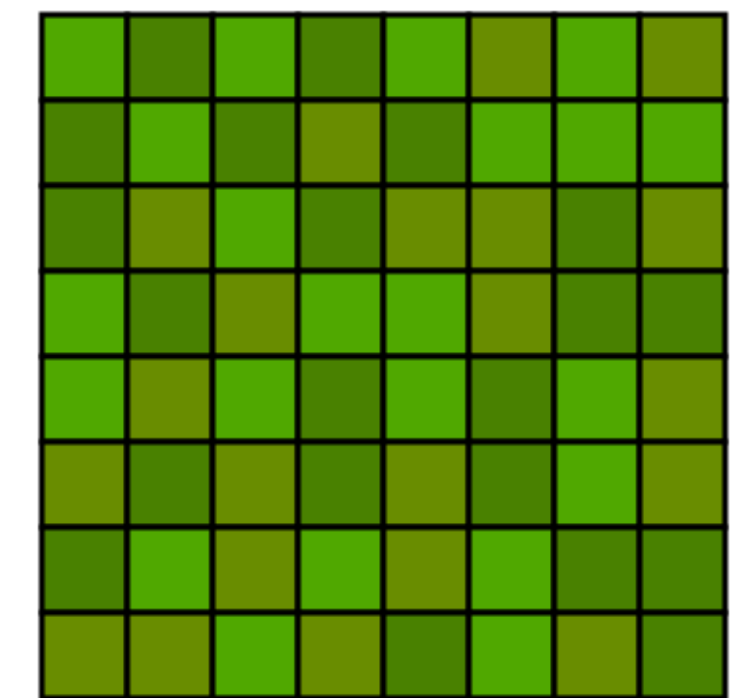
- memory page is split into elements
- in this example **allocation size 64**
- every single element is used
- **64 elements** per page



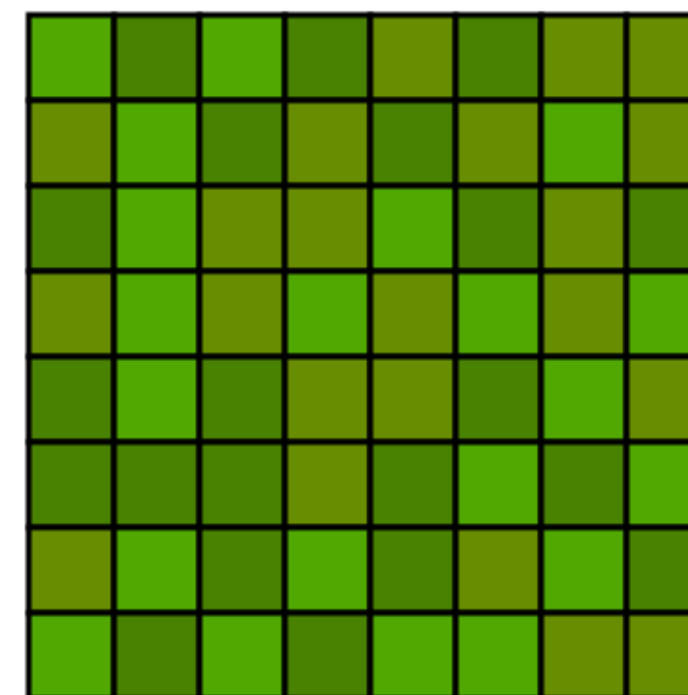
0x300c7



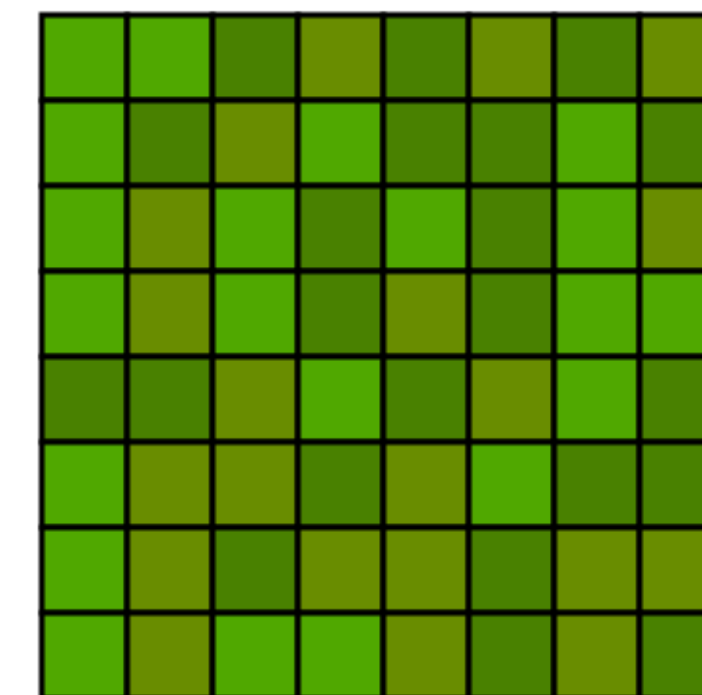
0x3013b



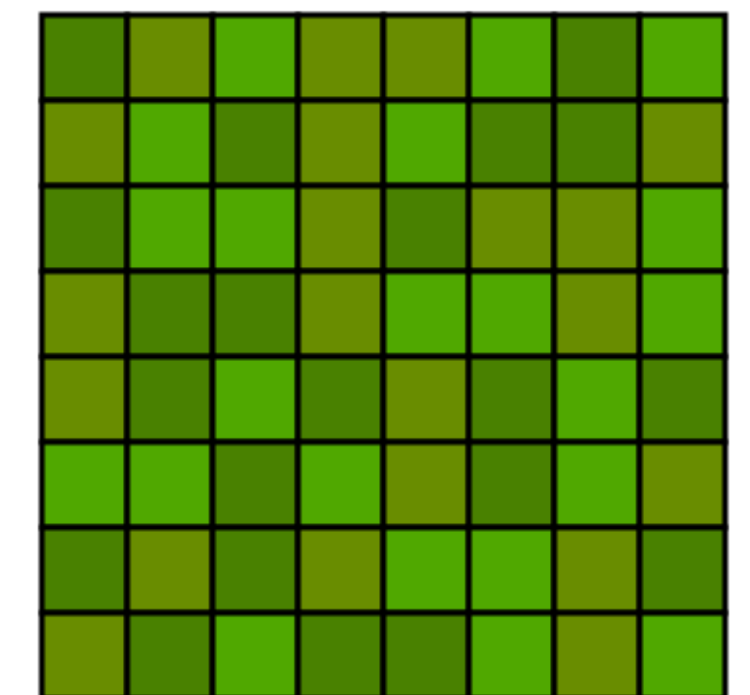
0x30290



0x30dad



0x30dbe

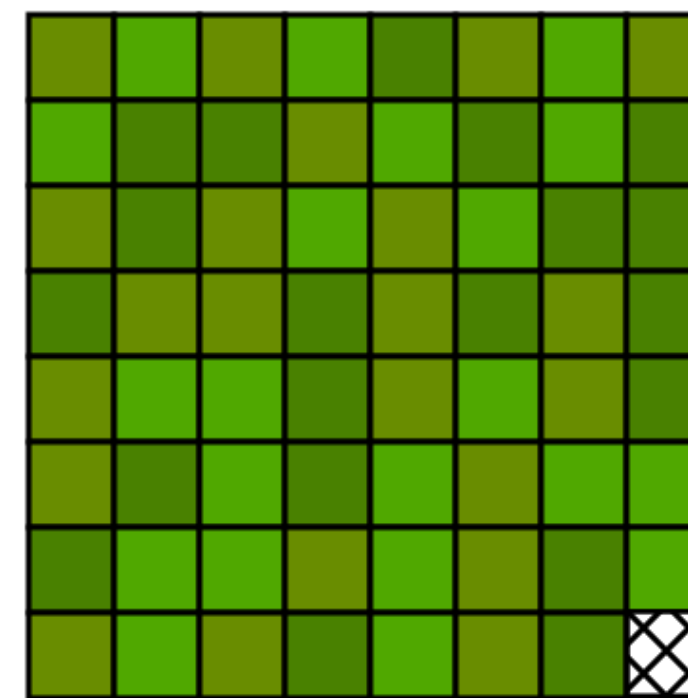


0x30e29

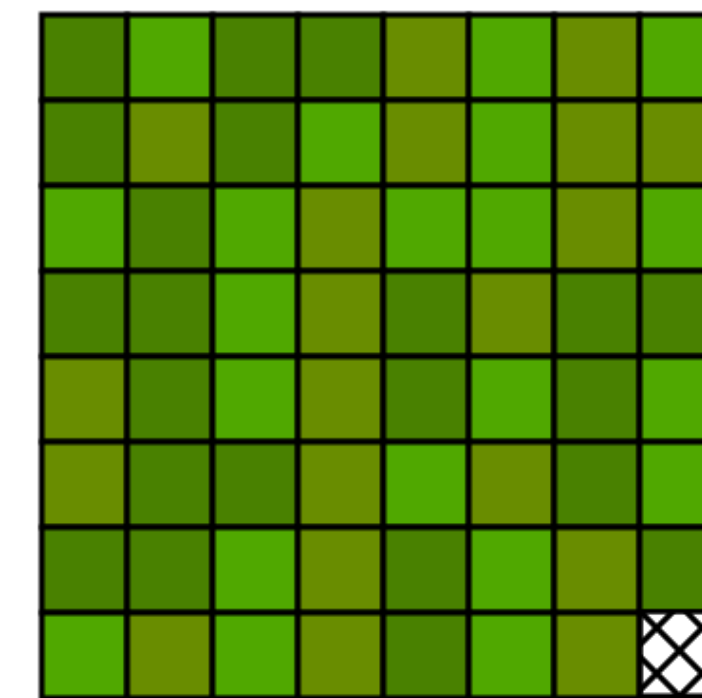


iOS Zone Allocator Allocations in iOS 7 & 8

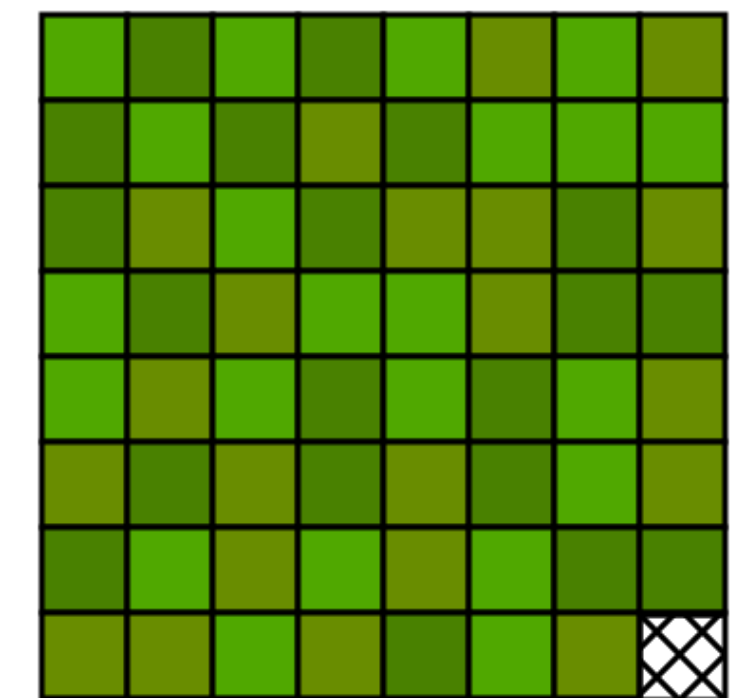
- Apple added **meta data to end of page**
- **one less element (63)** fits into a page
- exactly what we expect due to the meta data at end



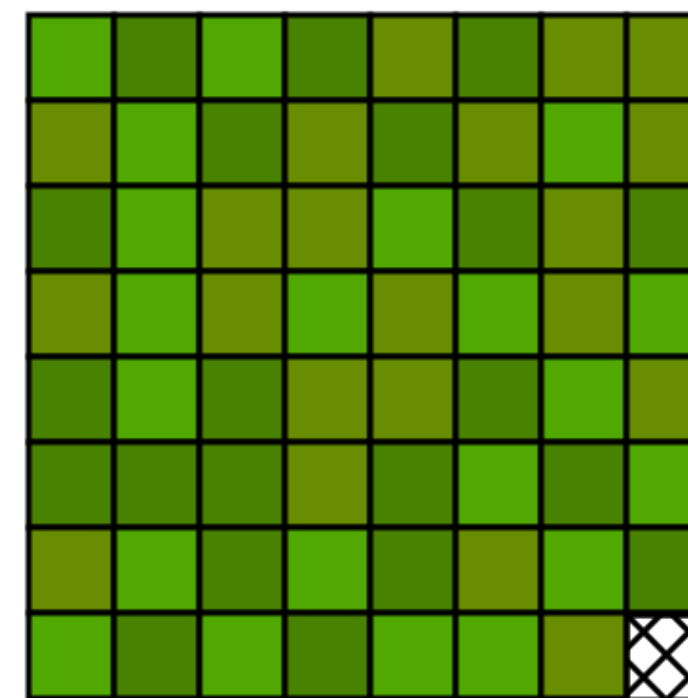
0x300c7



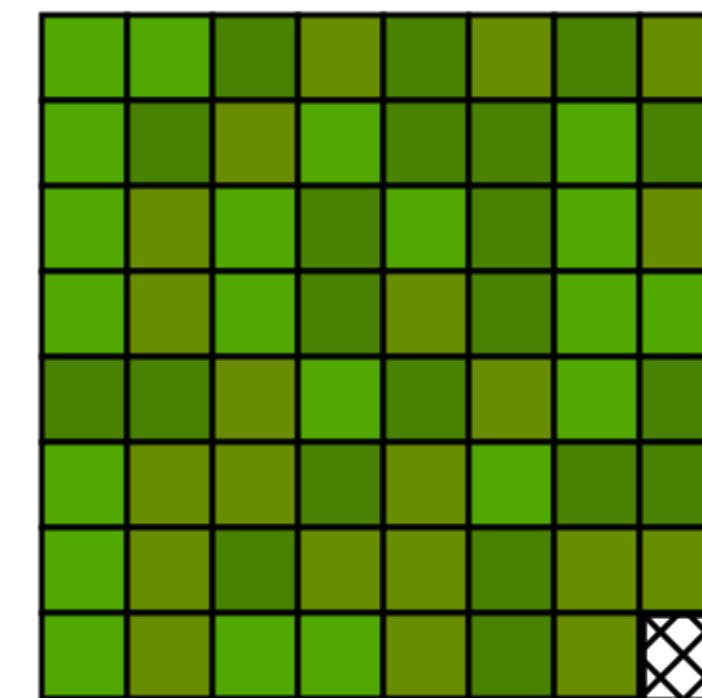
0x3013b



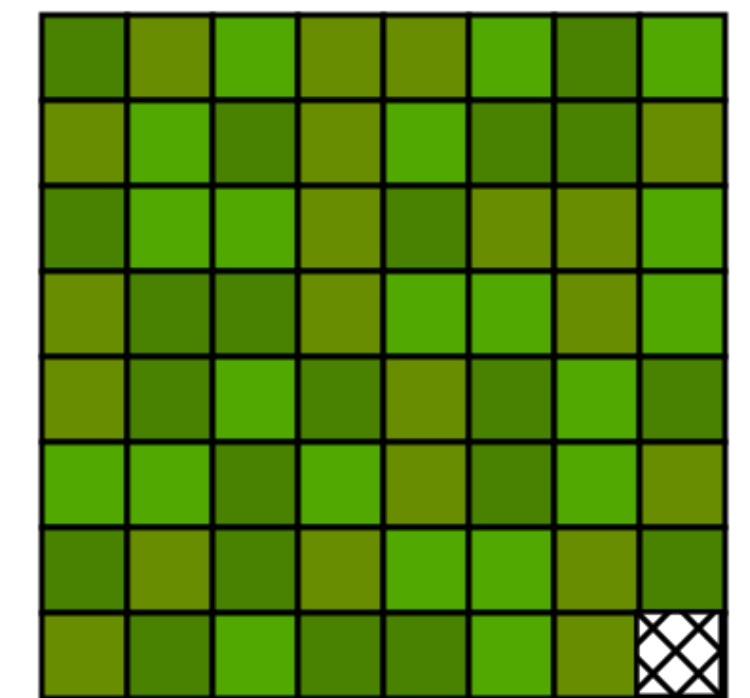
0x30290



0x30dad



0x30dbe

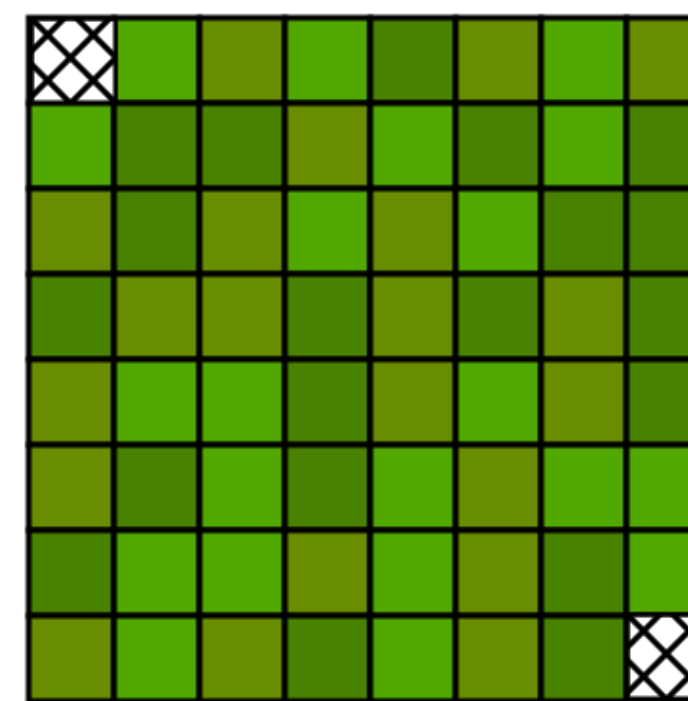


0x30e29

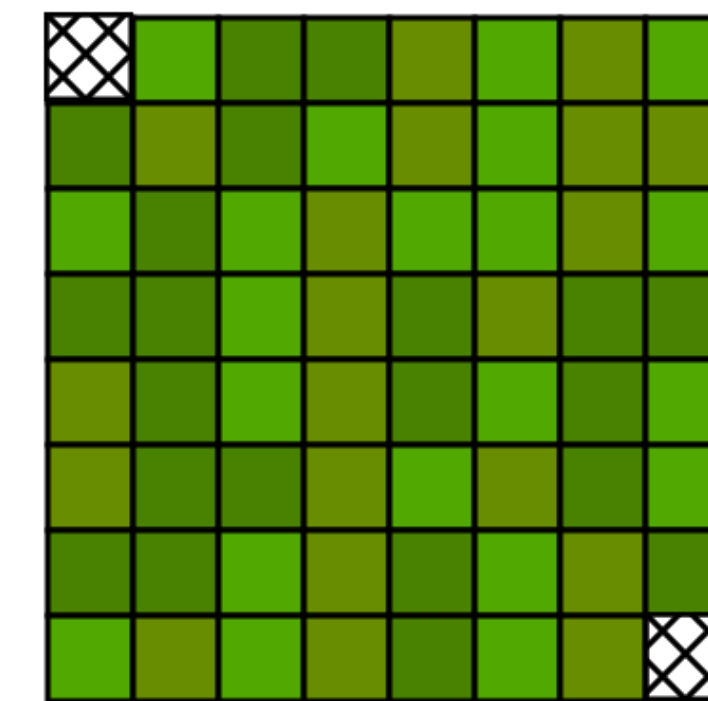


iOS Zone Allocator Allocations in iOS 9

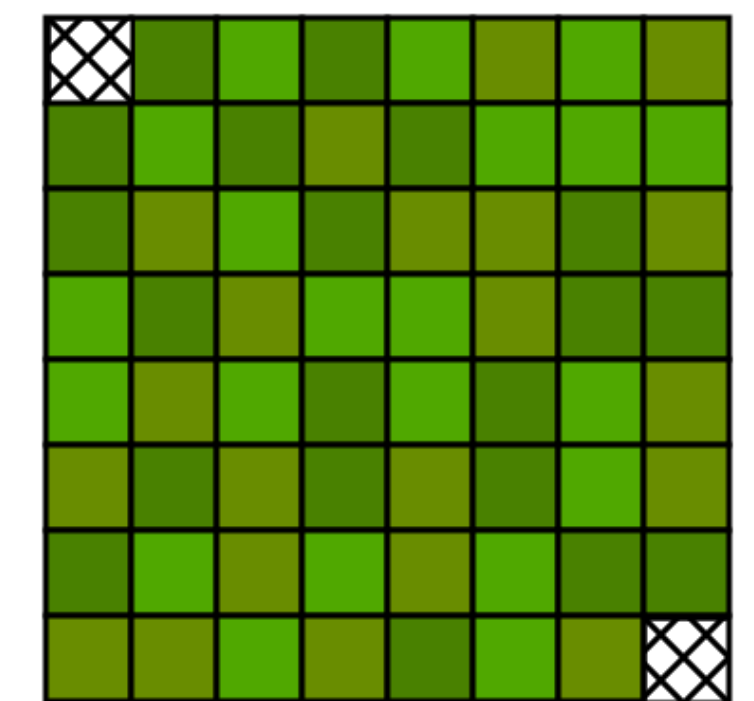
- Apple moved **meta data to beginning of page**
- block in beginning cannot be used
- but **why only 62 elements?**
- **why** is the **last block** still **unused**?
- there must be something wrong



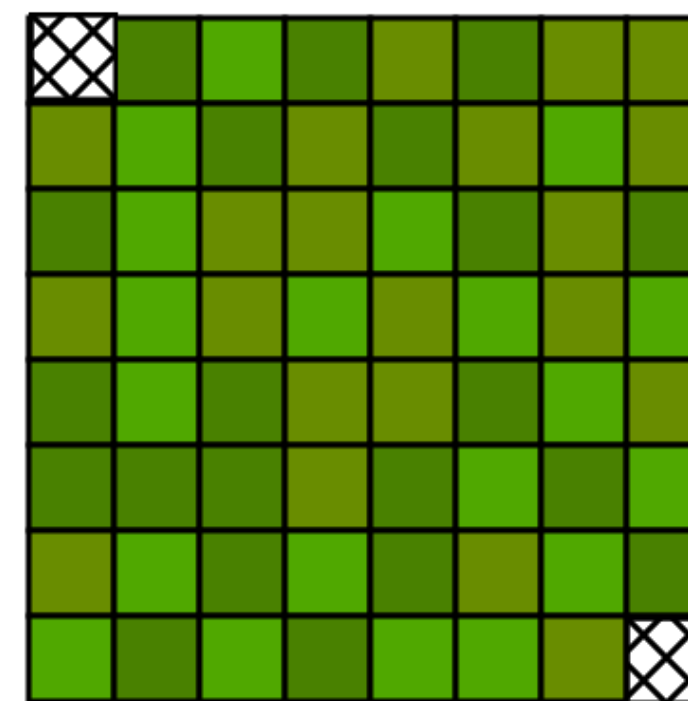
0x300c7



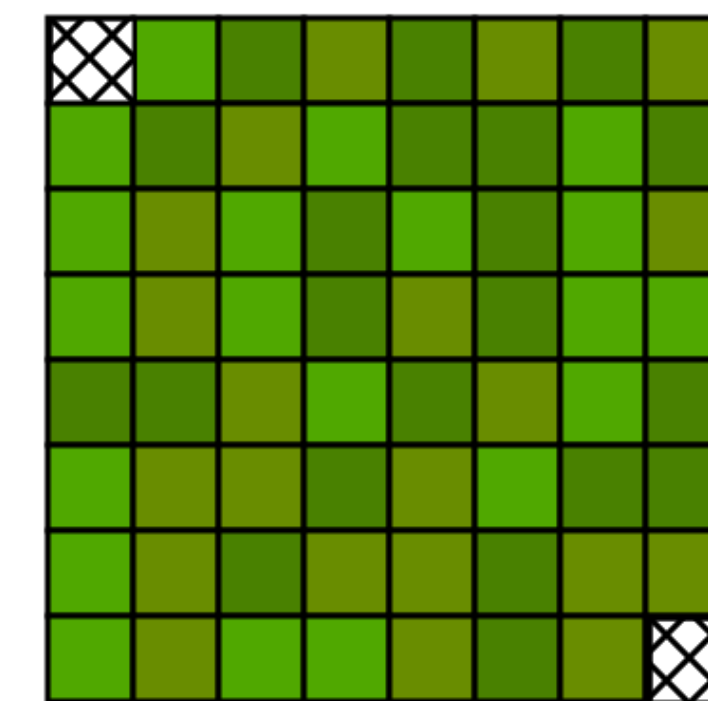
0x3013b



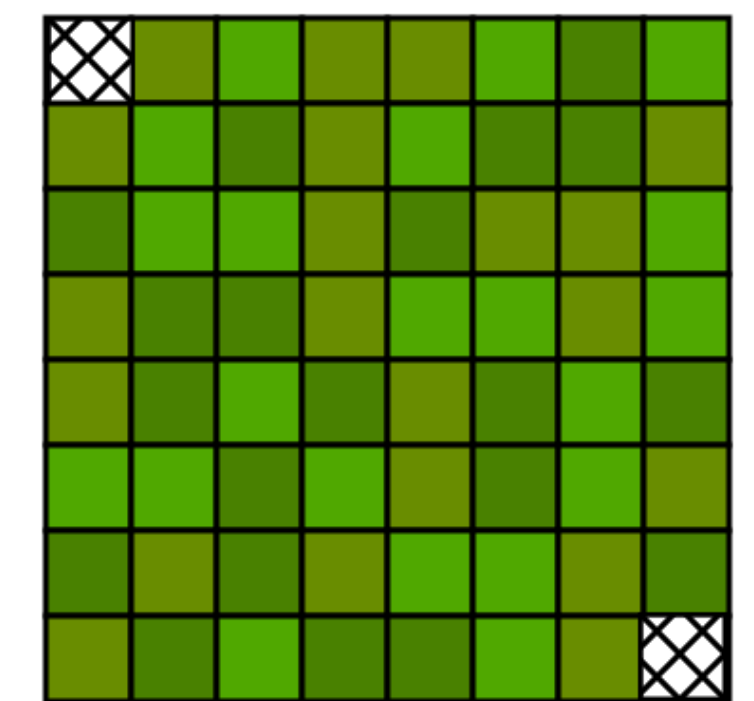
0x30290



0x30dad



0x30dbe



0x30e29



zcram() (I)

- first element positioned aligned after meta data

```
vm_offset_t first_element_offset;
if (zone_page_metadata_size % ZONE_ELEMENT_ALIGNMENT == 0){
    first_element_offset = zone_page_metadata_size;
} else {
    first_element_offset = zone_page_metadata_size +
        (ZONE_ELEMENT_ALIGNMENT - (zone_page_metadata_size % ZONE_ELEMENT_ALIGNMENT));
}
```

- following elements are added one by one

```
for (pos_in_page = first_element_offset;
    (newmem + pos_in_page + elem_size) < (vm_offset_t)(newmem + PAGE_SIZE);
    pos_in_page += elem_size) {
    page_metadata->alloc_count++;
    zone->count++; /* compensate for free_to_zone */
    free_to_zone(zone, newmem + pos_in_page, FALSE);
    zone->cur_size += elem_size;
}
```



zcram() (II)

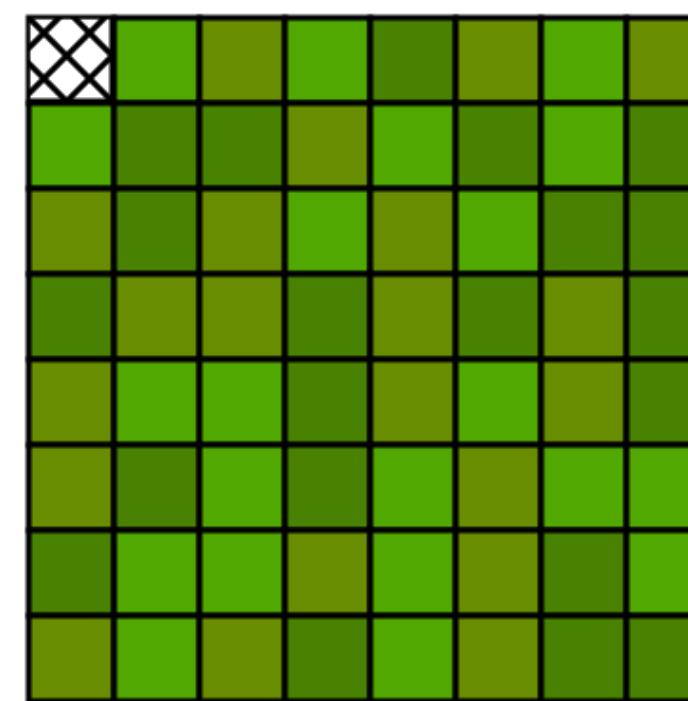
- checks in each iteration if end of element is still within page
- check is broken it uses **< (newmem + PAGE_SIZE)**
- if element ends exactly on page boundary it is considered out of bound
- must be **<= (newmem + PAGE_SIZE)** otherwise always loses last element

```
for (pos_in_page = first_element_offset;  
    (newmem + pos_in_page + elem_size) < (vm_offset_t)(newmem + PAGE_SIZE);  
    pos_in_page += elem_size) {  
    page_metadata->alloc_count++;  
    zone->count++; /* compensate for free_to_zone */  
    free_to_zone(zone, newmem + pos_in_page, FALSE);  
    zone->cur_size += elem_size;  
}
```

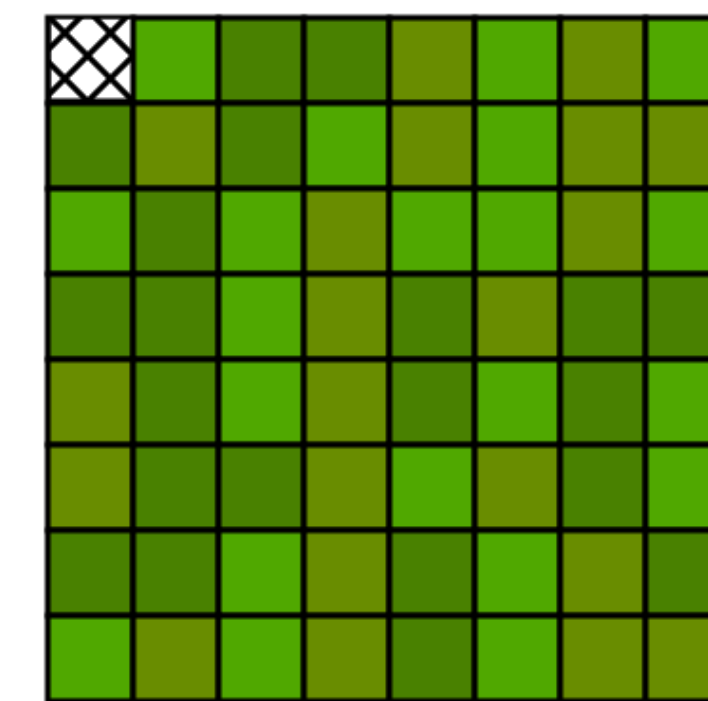



iOS Zone Allocator Allocations in iOS 9 >= 9.2

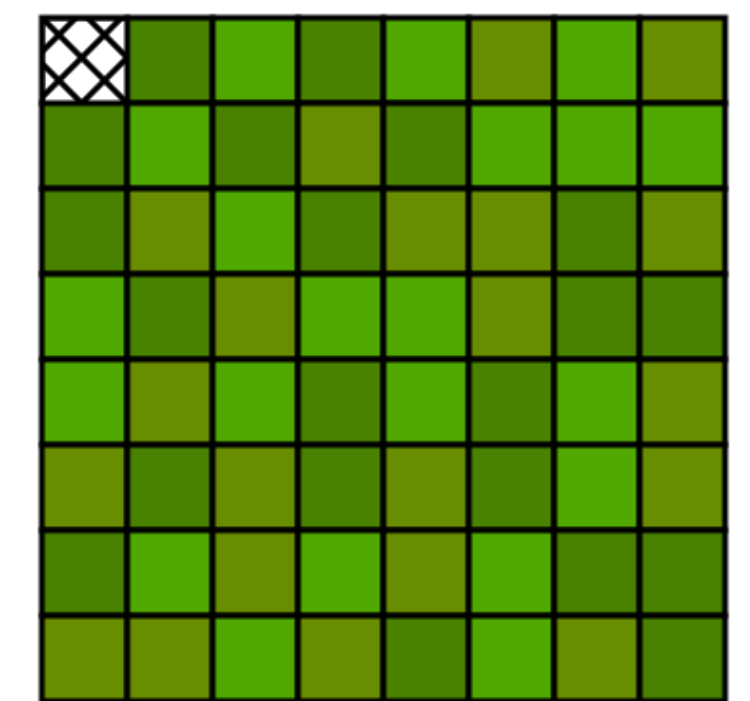
- Apple **refactored code** - freelist order now randomized
- refactoring fixed bug
- only meta data at beginning not used
- **63 elements** fit per page



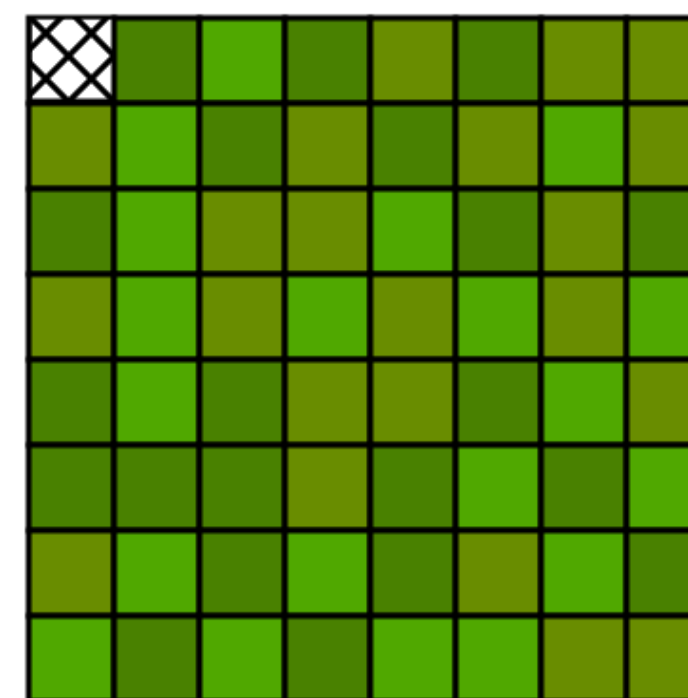
0x300c7



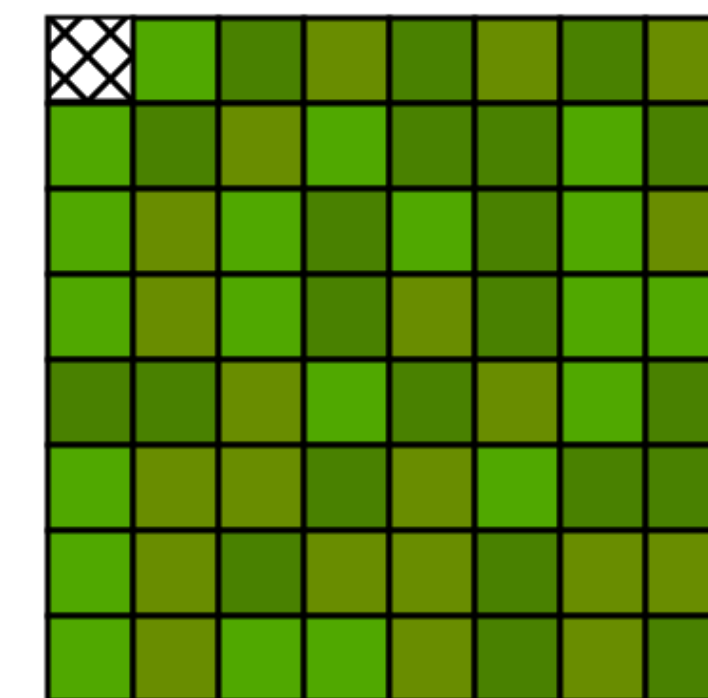
0x3013b



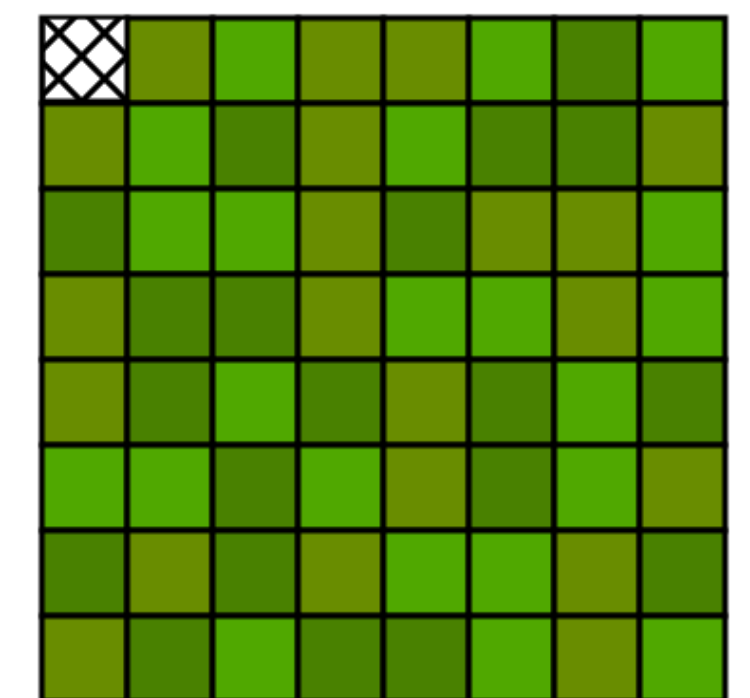
0x30290



0x30dad



0x30dbe

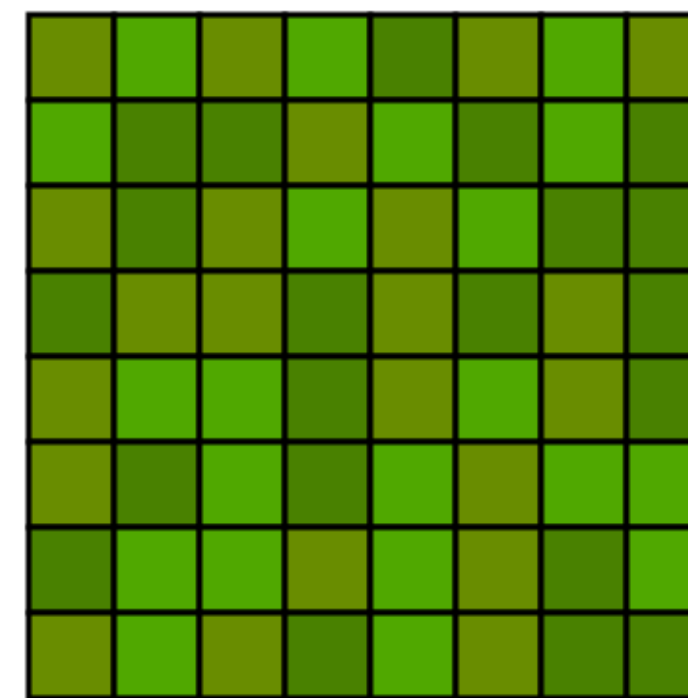


0x30e29

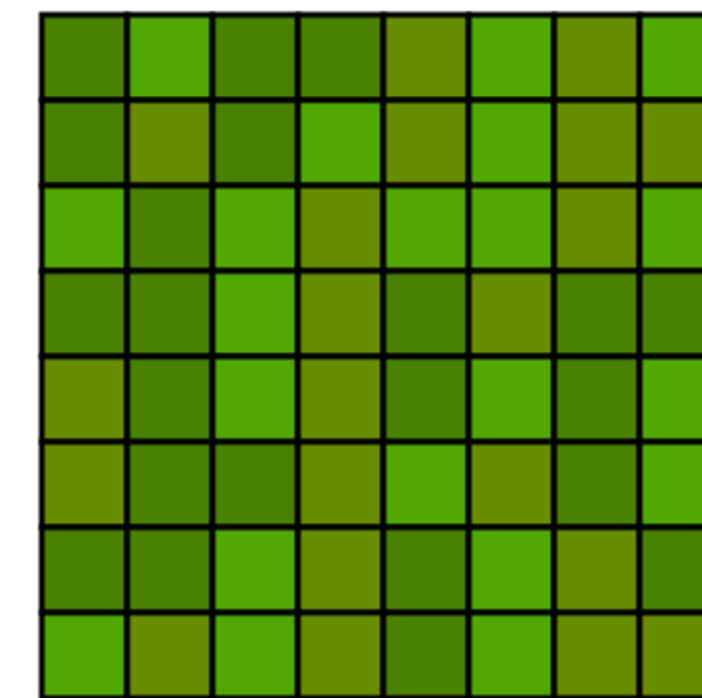


iOS Zone Allocator Allocations in iOS 10

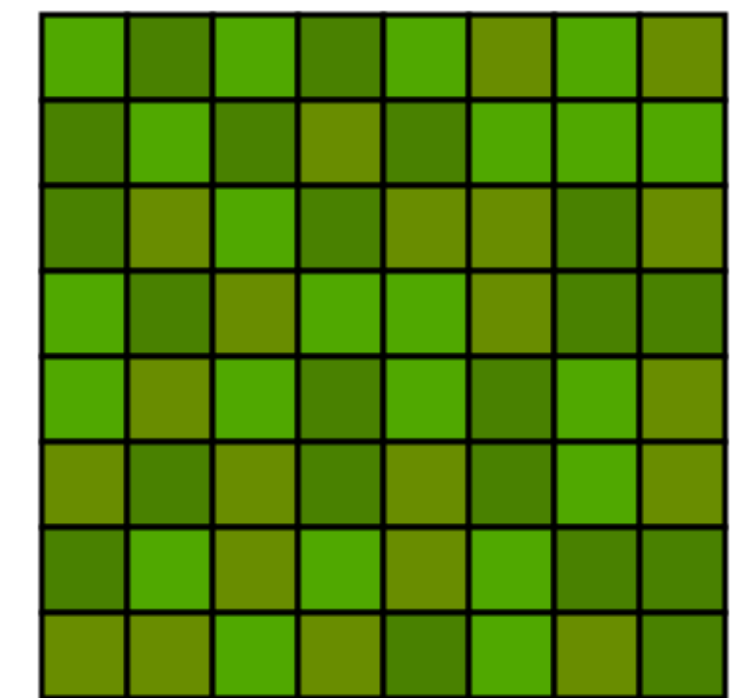
- Apple moved meta data out of page
- once again full page used for allocations
- **exactly 64 elements** fit into page



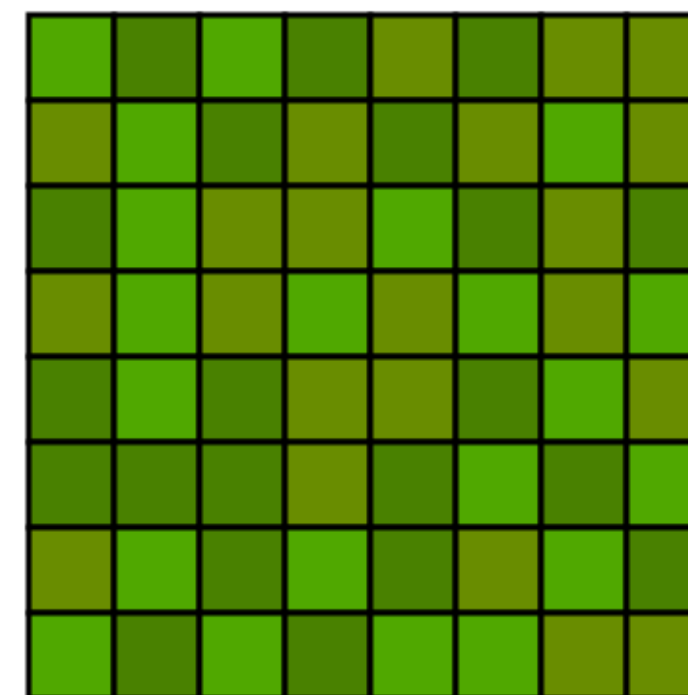
0x300c7



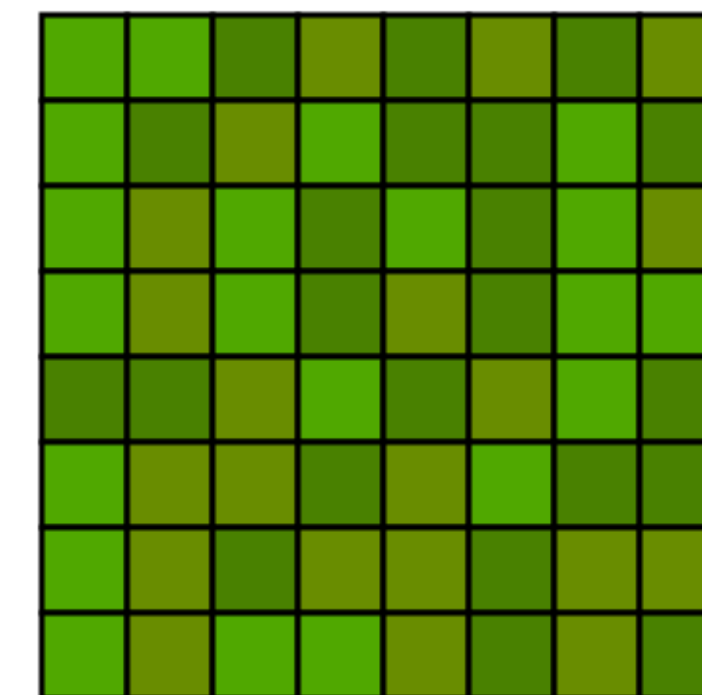
0x3013b



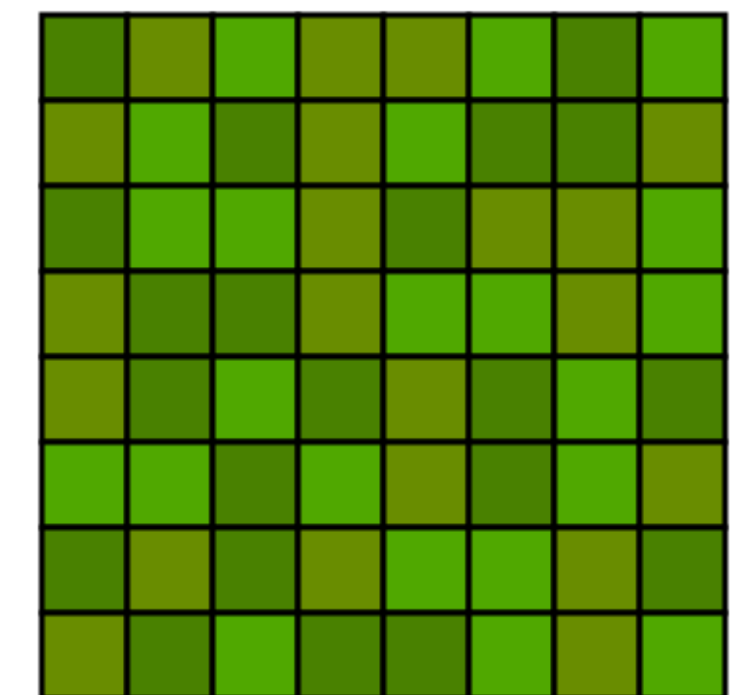
0x30290



0x30dad



0x30dbe



0x30e29

Questions?

www.antid0te.com
stefan@antid0te.com

© 2017 by [ANTIDOTE](#). All rights reserved