



Object Oriented Software Development

Week 5



Bagel

Java library developed by a UniMelb student for creating basic games

May differ from other libraries – careful when researching

You MUST use Bagel



The screenshot shows the Bagel Java library documentation website. The browser address bar displays the URL: `people.eng.unimelb.edu.au/mcmurtrye/bagel-doc/index.html?bagel/AbstractGame.html`. The page layout includes a left sidebar with navigation links for 'All Classes', 'Packages' (listing `bagel`, `bagel.map`, and `bagel.util`), and 'Classes' (listing `AbstractGame`, `Drawing`, `DrawOptions`, `Font`, `Image`, `Input`, `Shader`, and `Window`). Below these are 'Enums' (`Keys`, `MouseButtons`) and 'Exceptions' (`BagelError`). The main content area has a top navigation bar with tabs: 'OVERVIEW', 'PACKAGE', 'CLASS' (selected), 'TREE', 'DEPRECATED', 'INDEX', and 'HELP'. Below this are links for 'PREV CLASS', 'NEXT CLASS', 'FRAMES', and 'NO FRAMES', followed by a summary line: 'SUMMARY: NESTED | FIELD | CONSTR | METHOD' and 'DETAIL: FIELD | CONSTR | METHOD'. The main content displays the 'bagel' package and the 'Class AbstractGame', which extends `java.lang.Object` and `bagel.AbstractGame`. It includes the declaration: `public abstract class AbstractGame extends java.lang.Object` and a description: 'The base class for all Bagel games.' A 'Constructor Summary' section is also visible, with a 'Constructors' tab selected. It lists two constructors: `AbstractGame()` with the description 'Create the game with a default window size (1024x768) and title ("Game").' and `AbstractGame(int width, int height)` with the description 'Create the game with a default title ("Game").'

The window



Divided into pixels; by default, the window is 1024 pixels wide and 768 pixels high

The origin is **top-left**

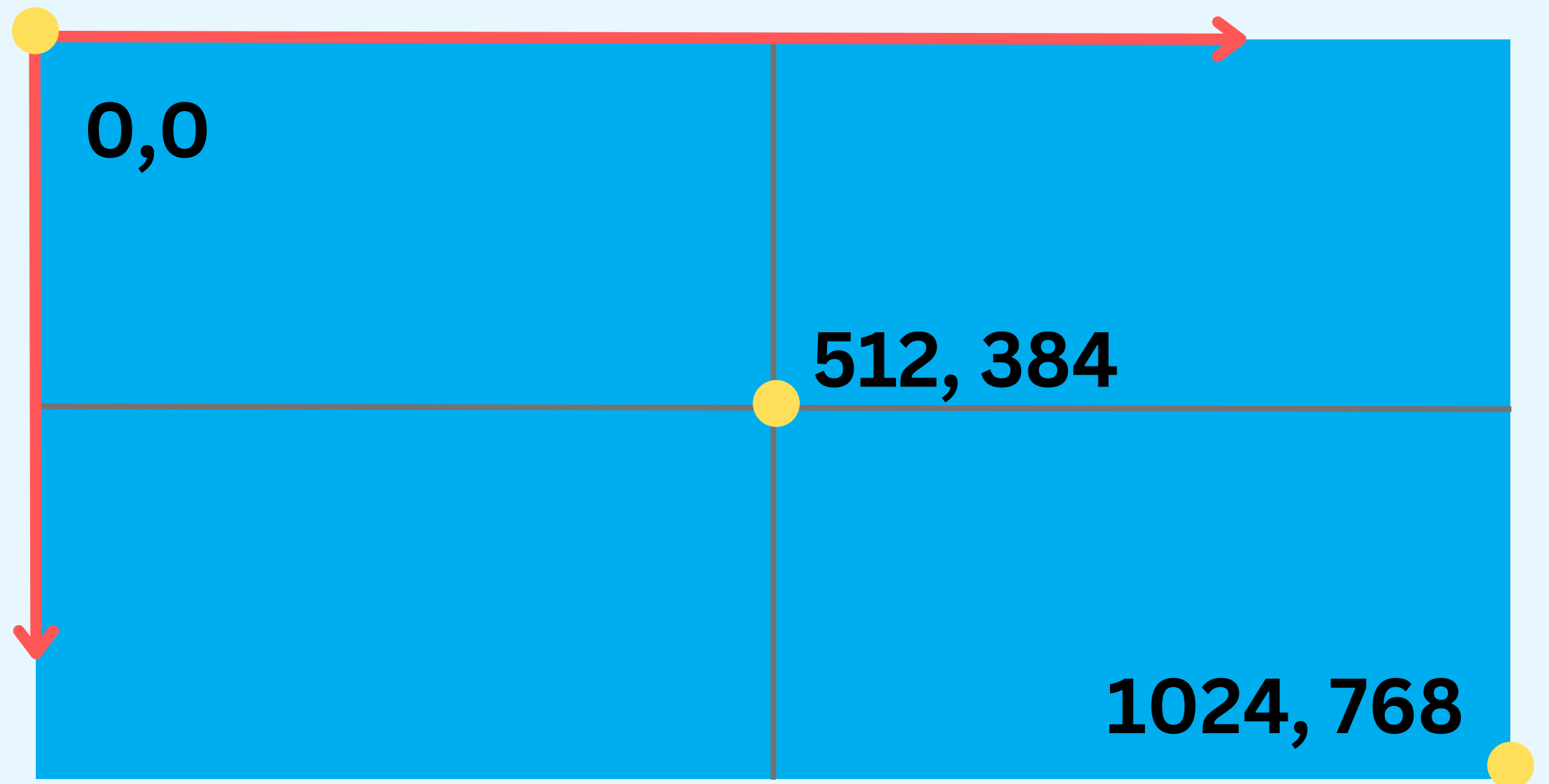




Image rendering

In Bagel, the screen is cleared and re-rendered for every frame

This occurs in the **update()** method - it is called for every frame

We will be working with 60fps - so **update()** is called 60 times a second

If your monitor has a higher refresh rate, will look different

Kahoot!



Pairs or individually





Share a single device



2. Now, you are ready to commit and push the `bagel-starter-pack` - refer to Week 3 Lectures or Workshop 3 for GitLab commands.

- (a) Create a new folder in the cloned repository, call it "Week 5" or "Workshop 4".
- (b) Copy all the contents of `bagel-starter-pack` into the new folder you just created.
- (c) Run `git add` to add the files, and `git commit -m "copied template"` to make your first commit.
- (d) Then, run `git push` to push to the remote repository; you should commit and push at the end of each question below.

This PC > Documents > git > cmuir-workshops >	
Name	Status
Workshop 3	
Workshop 4	

This PC > Documents > git > cmuir-workshops > Workshop 4 >		
Name	Status	Date modified
res		21/08/2022 5:30 PM
src		21/08/2022 5:30 PM
.gitignore		21/08/2022 5:30 PM
pom		21/08/2022 5:30 PM

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\cmuir>cd C:\Users\cmuir\OneDrive\Documents\git\cmuir-workshops

C:\Users\cmuir\OneDrive\Documents\git\cmuir-workshops>git add .
```