

Quality Assurance, Sprint 2

Emulator Device Specification:

Resolution: 1280x800

CPU: Intel Atom (x86_64)

RAM: 1536MB

Our user uses a Samsung Tab 4 – so the specs are as close as the actual device. As for this sprint, the app only has to run on this device with no significant defects.

Defects in Garden App

Garden App is the front end of the app, so most of the tests are done manually in Android emulator.

The toast popups are not very responsive

Status: Closed

Reproducibility: Always

Priority: Low

Description: Found by tapping rapidly on different plant buttons in View Species List. The popups are not very responsive as user clicks the buttons. This is an issue with Android's Toast.makeText. No fix needed – these toasts are just placeholders for features that will be implemented later.

Species list view's OnClickListener not responding to taps correctly

Status: Fixed

Reproducibility: Always

Priority: High

Description: Found by clicking on the species buttons on Species List View. The buttons do not register taps. Fixed by refactoring OnClickListener and adding a statement that was missing.

Buttons on home screen not showing in correct positions

Status: Fixed

Reproducibility: Always

Priority: Low

Description: Found by running the app on emulator. The buttons are misplaced. Fixed by modifying main activity's xml with better positioning params.

Minimizing the app in View Garden causes the app to crash

Status: Open

Reproducibility: Always

Priority: Medium

Description: Found by minimizing the app when the app is in View Garden. Trying to bring the app back up causes the app to stop. Not fixed yet – cause of crash is unknown, may need some time to look into it.

The tap/drag handling code favors drag too much on a physical device

Status: Open

Reproducibility: Always

Priority: High

Description: Found by dragging on the garden view on a physical android device. On the emulator, it appears to work fine, but on a physical device, it is hard to place a circle because the tolerance for the beginning and ending coordinate is so tight. Not fixed yet – this will take some time to fix.

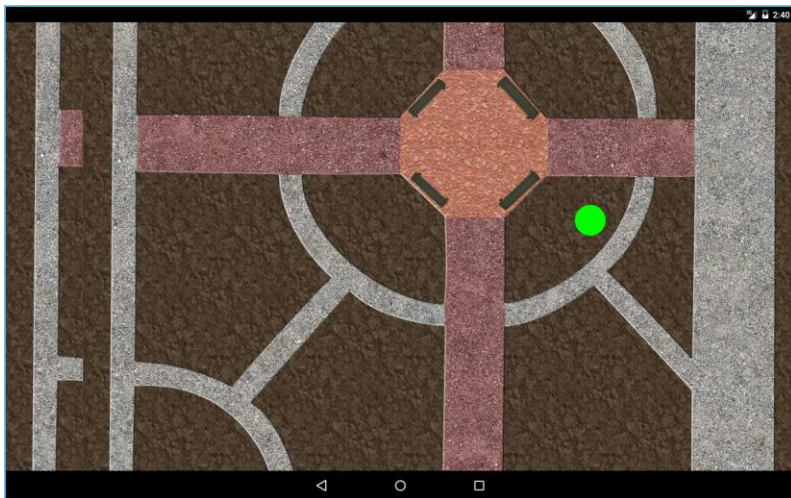
Garden rendering issue on 1280x800 resolution

Status: Open

Reproducibility: Always

Priority: High

Description: Found by dragging on the garden view on emulator. As shown in the screenshot below, the left edge of the screen is rendered incorrectly. This is caused by garden background texture being smaller than the screen resolution. Not fixed yet – while this affects user experience, the app still functions. This needs to be fixed in the future.



Circles appears to be in different sizes on different screen resolutions

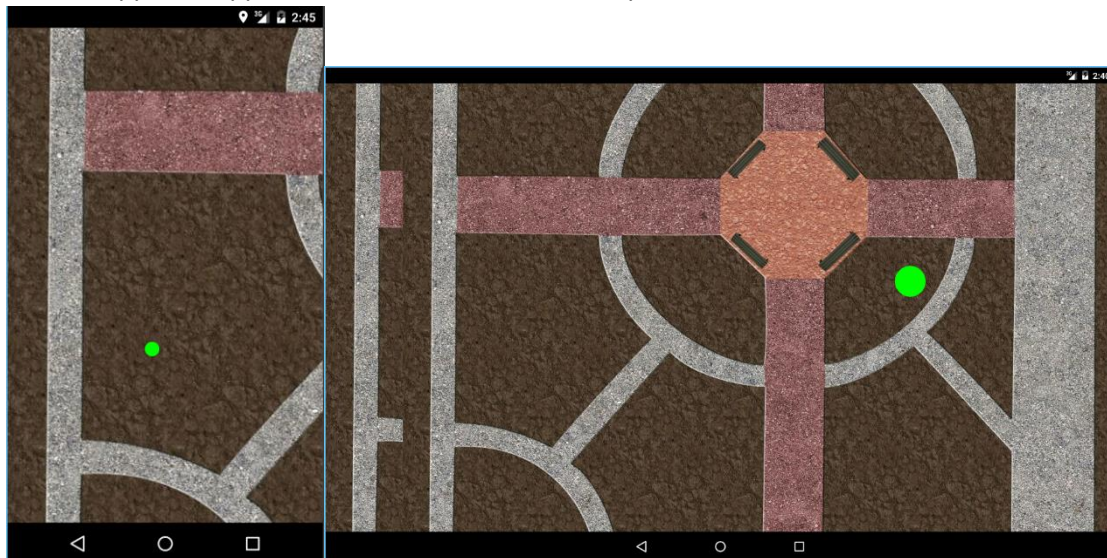
Status: Open

Reproducibility: Always

Priority: Low

Description: Found by tapping the screen on the garden view on emulator in different resolutions. As shown in the two screenshots below, the green circle is smaller on 1080x1920 than on 1280x800. Not

fixed yet – the goal is to get the app running on 1280x800, the resolution of the device our user plans to run this app on. Support for other resolutions is not prioritized.



Defects in Garden Engine

Garden engine is tested with Unit Tests and code inspections.

Garden serialize and deserialize use different formats

Status: Fixed

Reproducibility: Always

Priority: High

Description: Found by running garden unit tests. The `GardenToString()` produces a different text format than the one `StringToGarden()` uses to deserialize. Fixed by modifying the `GardenToString` and `StringToGarden` to use the same text format.

Garden engine lacks documentation

Status: Open

Reproducibility: Always

Priority: High

Description: Found by code inspection. There is no JavaDoc documentation for any of the classes in garden engine. This should be prioritized and fixed as soon as possible in order to maintain the code base.

Garden engine needs refactoring

Status: Open

Reproducibility: Always

Priority: High

Description: Found by code inspection. There are some design issues in Garden.java that need to be fixed.

- `boolean StringToGarden(String garden)` should be static `Garden StringToGarden(String garden)`. This method should not change the state of an already created garden like it is currently doing, but create a new garden.
So one may construct a garden with
`Garden myGarden = Garden.StringToGarden(gardenString);`
For consistency it would also be a good idea to make `GardenToString` a static method, i.e. `static String GardenToString(Garden garden)`
- Add public getters and setters for the fields in `Species` and `Plant`. Also add JUnit tests for those.
- `String[] ListSpecies()` should be changed to `String[] GetSpeciesNames()` to avoid confusion. Do not enforce garden string formatting outside of this class. `String GetSpeciesInfo(String species)` is not very usable. I think a good implementation would be `Species GetSpecies(String speciesName)`.
So one may get the species info with
`String sun = myGarden.GetSpecies(myGarden.GetSpeciesNames()[0]).GetSun();`
`int water = myGarden.GetSpecies(myGarden.GetSpeciesNames()[0]).GetWater();`