presents

QA and Automation

Make a Better Game and Save Time + Money Doing It

What will I talk about?

1. How is QA done?
2. How to write tests?
3. How to automate?

Who am I?

- Nepo (they/them)
- QA (8y), freelance (2y)
- Consulting, work-for-hire, mentoring
- Barcelona Game Creators meetup

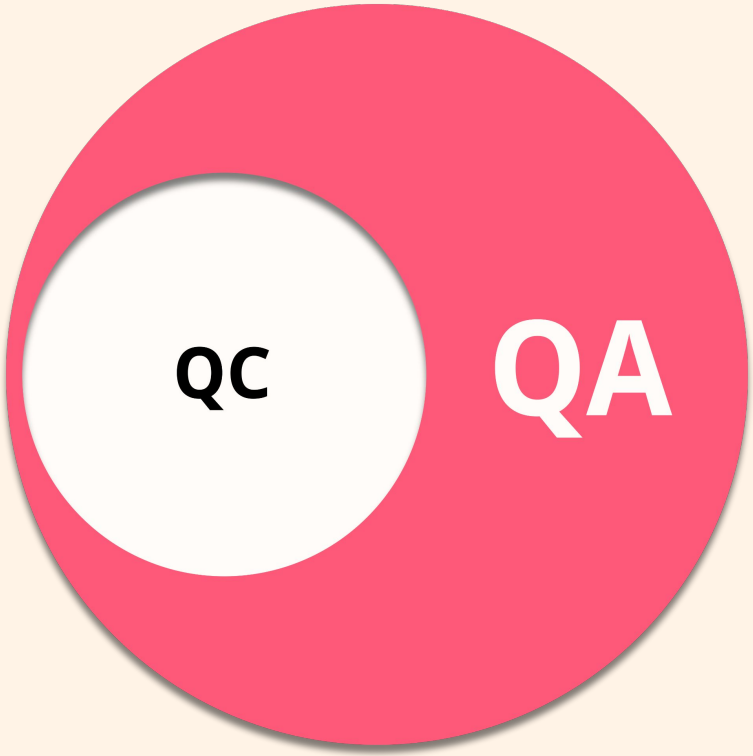
NOKORPO

Part 1

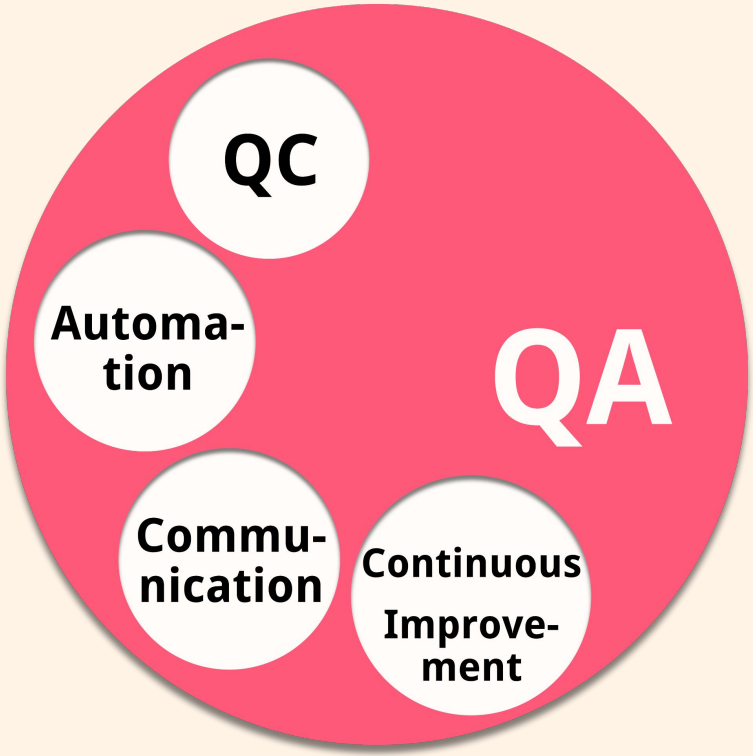
How is QA done?

What does QA *actually* mean?

QA ≠ QC



QA ≠ QC



What is QA?

Ideation	Preproduction	Production	Postproduction	Release	Live Players
- Brainstorm - Investigate - Prototype	- GDD - Vertical slice - Development plan	- Implement - Iterate - Iterate - Iterate - …	- Balancing - Bug fixing - Polishing	- Build - Testing - Marketing	

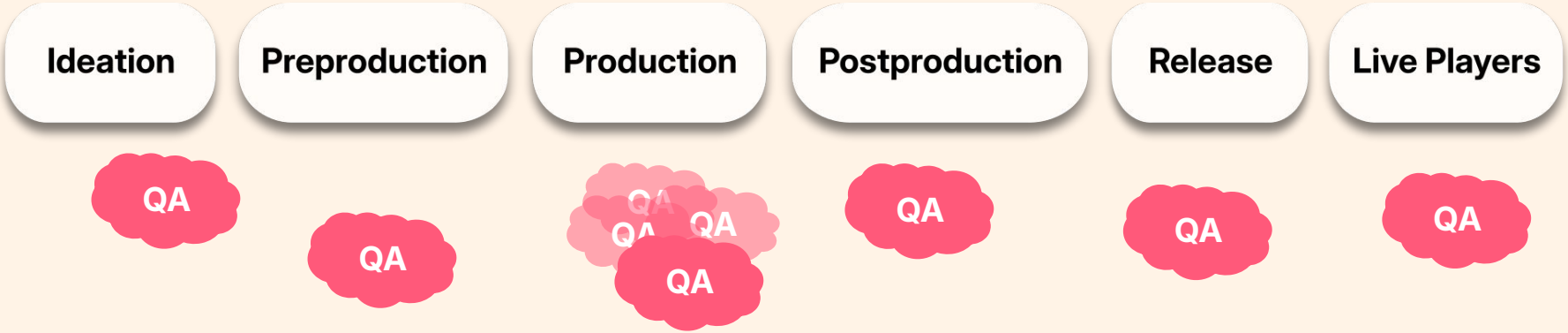
What is QA?

Ideation | Preproduction | Production | Postproduction | Release | Live Players

QA

What is QA?

What is QA?



QA works on **processes** throughout the entire development process

QA strategy: the objective

Objective: Find bugs?

Yes, but...

QA strategy: the objective

Ideation | Preproduction | Production | Postproduction | Release | Live Players

QA strategy: the objective

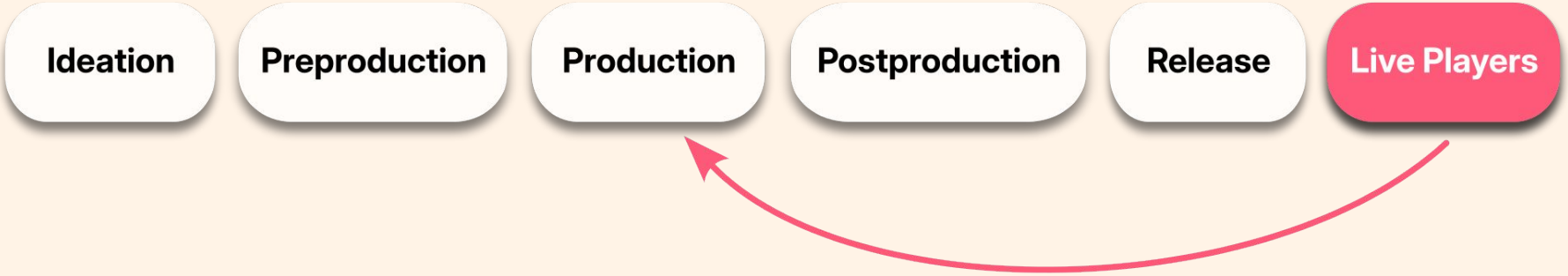
QA strategy: the objective

- Objective: find bugs **before the release**

QA strategy: the objective

Ideation | Preproduction | Production | Postproduction | Release | **Live Players**

QA strategy: the objective



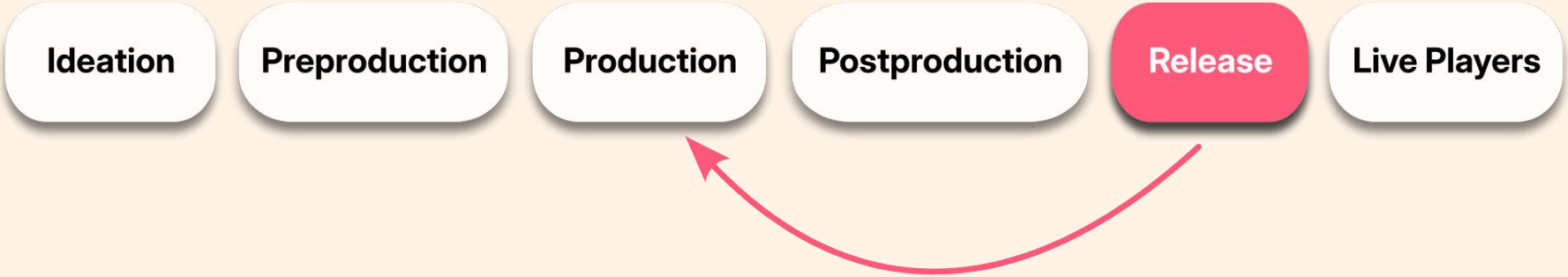
- Investigate
- Work on fix

- Verify it works

- Build
- Upload new version

- Download
- Play
- Report error

QA strategy: the objective

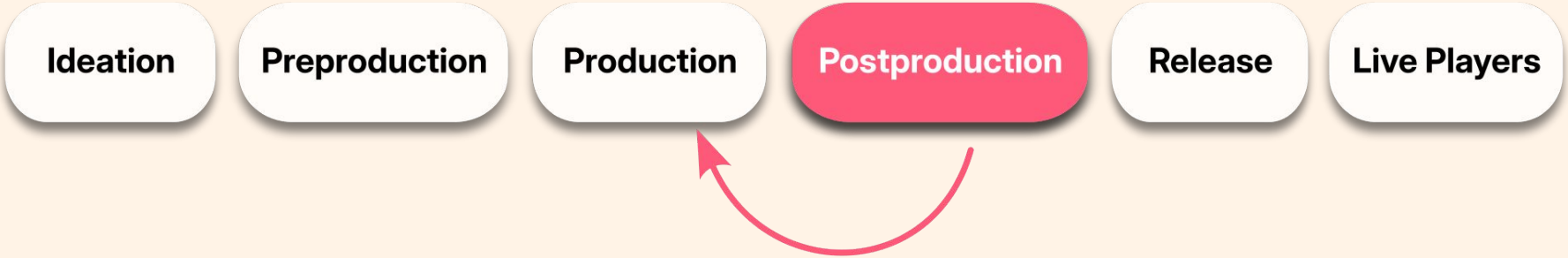


- Investigate
- Work on fix

- Verify it works

- Build
- Upload new version

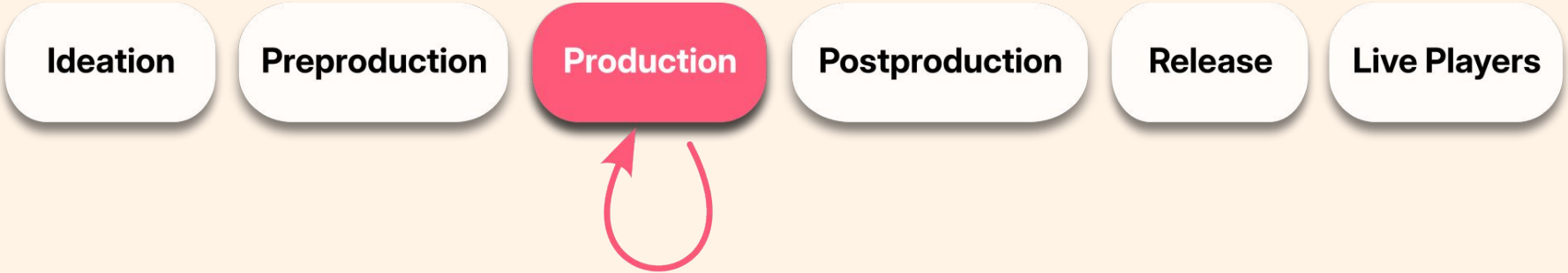
QA strategy: the objective



- Investigate
- Work on fix

- Verify it works

QA strategy: the objective

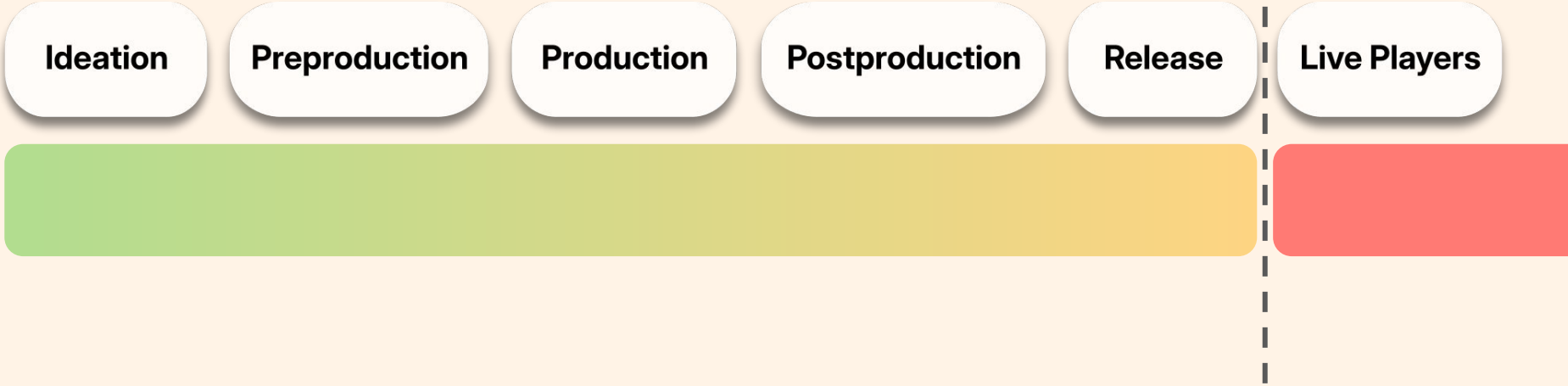


- Investigate
- Work on fix

QA strategy: the objective

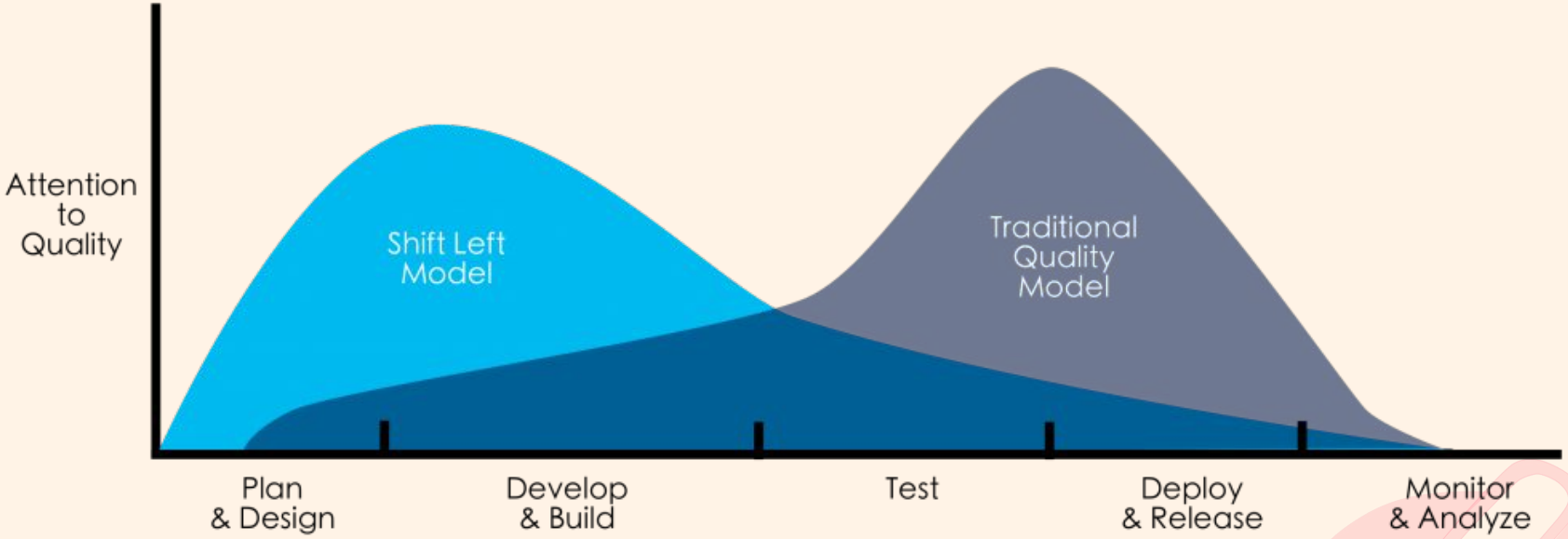
Ideation | Preproduction | Production | Postproduction | Release | Live Players

QA strategy: the objective



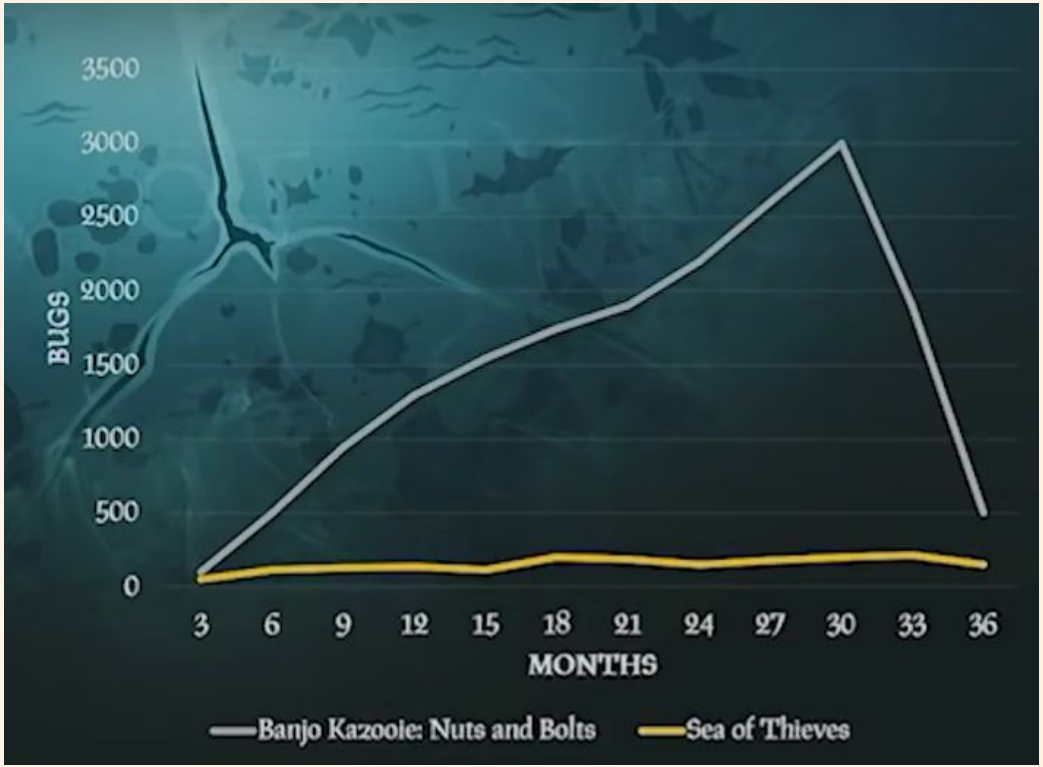
- Objetive: find bugs **as soon as possible**

Shift left testing



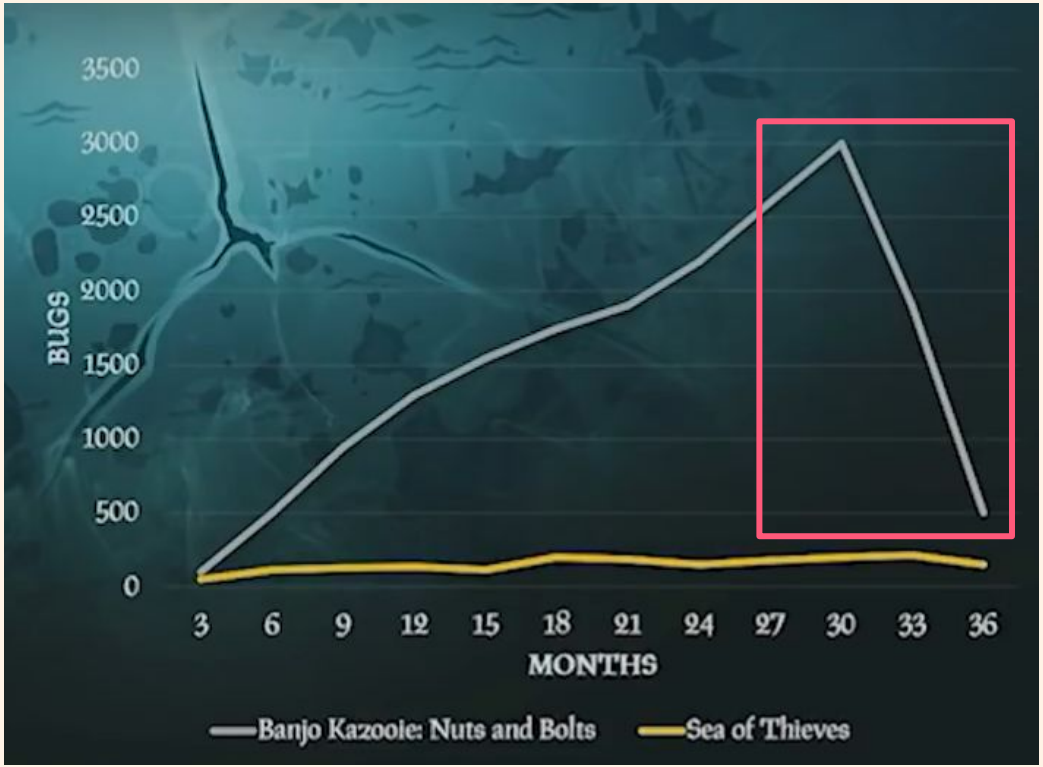
"It's 2017: Test automation is not optional when building mobile apps!" — Geert van der Cruijsen

Shift left testing



"Automated Testing of Gameplay Features in 'Sea of Thieves'", Robert Masella, GDC

Shift left testing



Crunch!

"Automated Testing of Gameplay Features in 'Sea of Thieves'", Robert Masella, GDC

What is a process?

Ideation	Preproduction	Production	Postproduction	Release	Live Players
- Brainstorm - Investigate - Prototype	- GDD - Vertical slice - Development plan	- Implement - Iterate - Iterate - Iterate - …	- Balancing - Bug fixing - Polishing	- Build - Testing - Marketing	

What is a process?

“A series of **tasks** performed in order to achieve **a goal**“

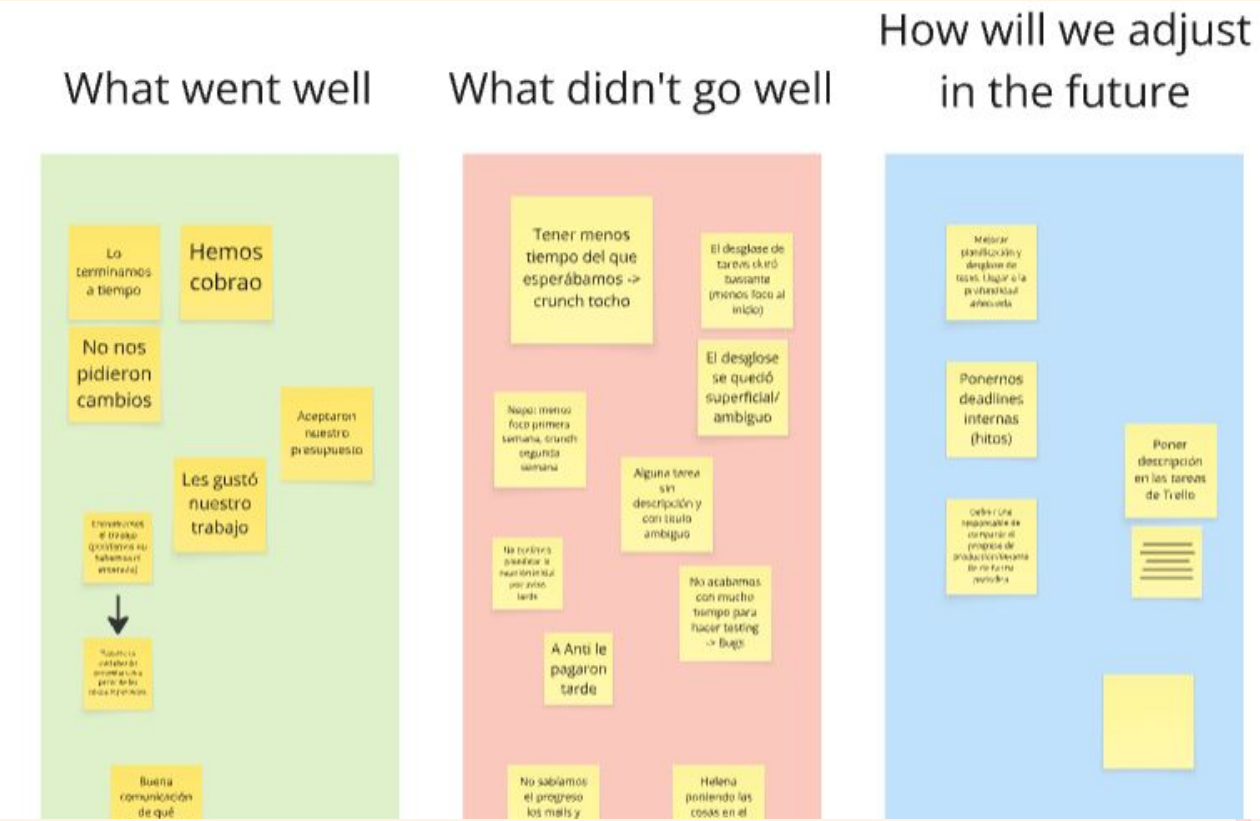
- Make code/art and **integrating it** in the game
- Upload a build to Steam and **releasing it**
- Have a meeting to **make decisions** about the project/team
- A brainstorming to **find new ideas**
- ... and many more examples

Testing processes

That’s a beautiful idea! But...

- How is a process tested?
- Can a meeting be tested?

Post-mortem


What went well
What didn't go well
How will we adjust in the future
Lo terminamos a tiempo
Hemos cobrao
No nos pidieron cambios
Aceptaron nuestro presupuesto
Les gustó nuestro trabajo
Buena comunicación de qué
Tener menos tiempo del que esperábamos -> crunch tocho
El desglose de tareas chirló bastante (menos foco al inicio)
El desglose se quedó superficial/ ambiguo
Alguna tarea sin descripción y con título ambiguo
No acabamos con mucho tiempo para hacer testing -> Bugs
A Anti le pagaron tarde
No sabíamos el progreso los mails y
Helena poniendo las cosas en el
Ponemos deadlines internas (hitos)
Poner descripción en las tareas de Trello

Retrospective

Retro (30')

Desde 27 Mar

Cada columna representa una categoría. Tenemos que pensar en nuestra semana bajo esas categorías y compartir lo que nos ha afectado. Tendremos 5-10 minutos para escribir lo que se nos ocurra, y el resto del tiempo para discutir cómo evitar las cosas malas y fomentar las buenas.

Good

Bad

Agreements

Documentar las ideas que se nos vayan ocurriendo en el Drive - Nepo creará el doc

Experimento: entrar en reunión cuando estemos currando (aunque sea muteado)

Actions

ROTI

Espacio reservado para dejar dudas, consejos, peticiones o amenazas de muerte.

Retrospective

Retro (30')

Desde 27 Mar

Cada columna representa una categoría. Tenemos que pensar en nuestra semana bajo esas categorías y compartir lo que nos ha afectado. Tendremos 5-10 minutos para escribir lo que se nos ocurra, y el resto del tiempo para discutir cómo evitar las cosas malas y fomentar las buenas.

Good

Bad

Agreements

- Documentar las ideas que se nos vayan ocurriendo en el Drive - Nepo creará el doc
- Experimento: entrar en reunión cuando estemos currando (aunque sea muteado)

Actions

ROTI

Asking feedback

Space at the end of a meeting to leave feedback

Espacio reservado para dejar dudas, consejos, peticiones o amenazas de muerte.

La persona que esté hablando de un post-it puede decir "vale, siguiente cosa"

Duró demasiado

Iterating on processes

Can a meeting/process be tested?

- **Look back**
- Think about what could be **changed** to **improve it**

Iteration unlocks **continuous improvement**

Does this work?

Project A brainstorming:

- No guide, no structure
- 4h meeting, no choice made. 2h more next week to choose an idea

Project B brainstorming:

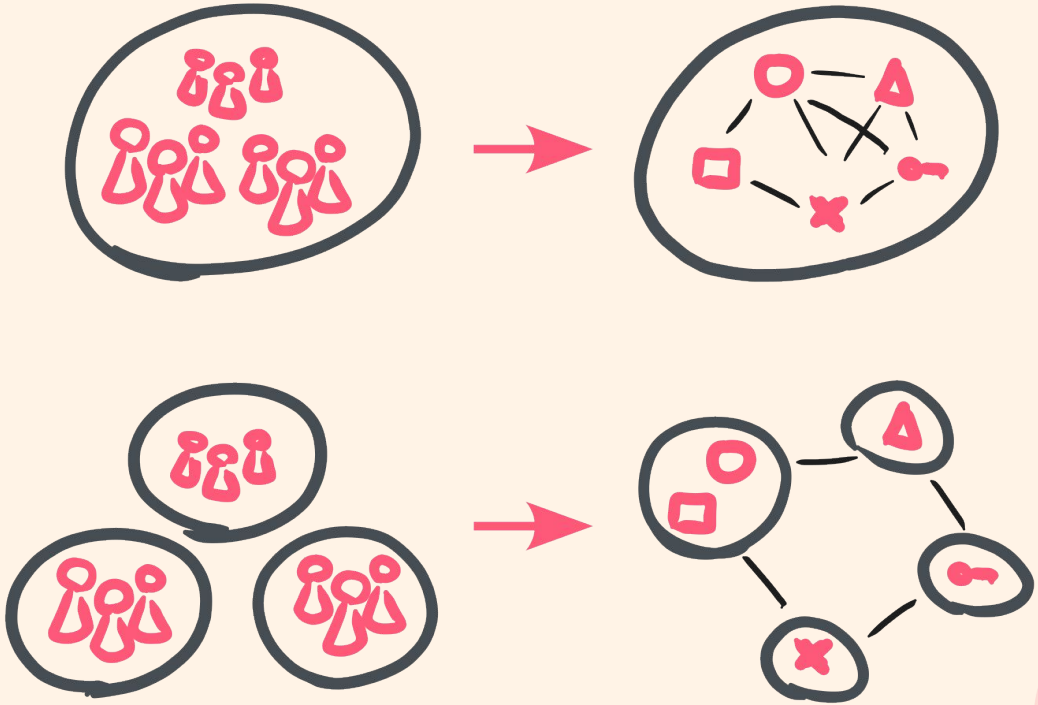
- 1 session, split in parts:
    1. Come up with ideas based on categories
    2. High level game proposal
- Choice made in 2h

Conway’s Law

“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

— Melvin Conway

Conway's Law



Summary

- QA can be applied throughout **all the development process**
- Objective: find+prevent defects **as soon as possible**
- Reduce **feedback loops**
- Look back (**good, bad**) and think about **what to change**

Part 2

How to write tests?

Test implementation

Pyramid testing model, Mike Cohn & Lisa Crispin

Test implementation

Pyramid testing model, Mike Cohn & Lisa Crispin

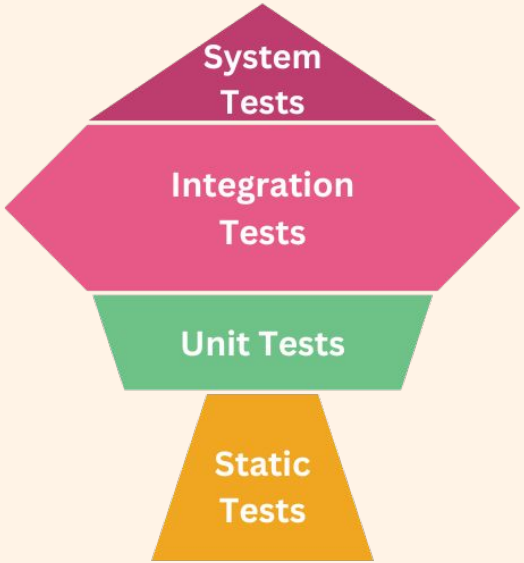
Test implementation

Pyramid testing model, Mike Cohn & Lisa Crispin

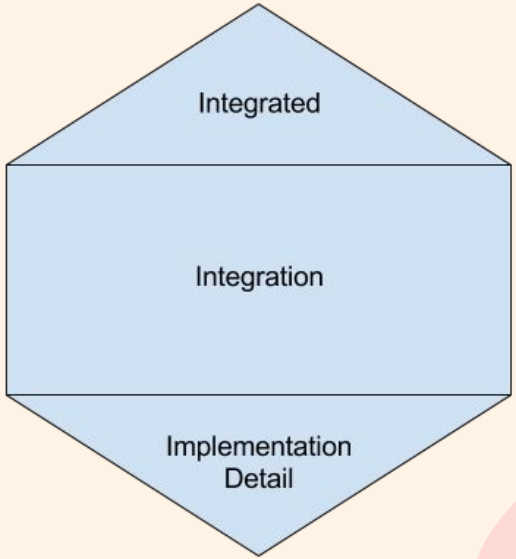
- Old
- Smells like waterfall
- Why do you even need so many unit test?!

Test implementation

“Write tests. Not too many. Mostly integration” — Guillermo Rauch



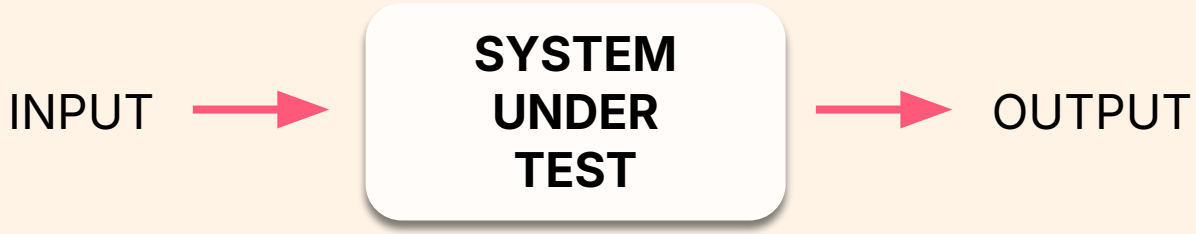
Trophy testing model, Kent C. Dodds



Microservices honeycomb testing model, Spotify

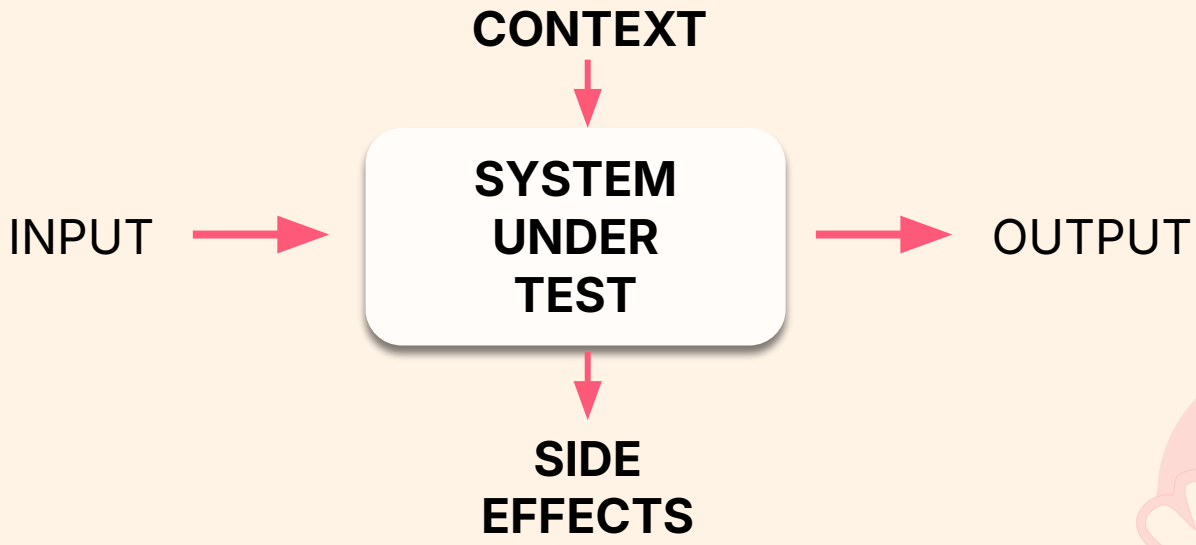
Why integration?

- When was the last time you wrote **a single unit** of code?
- What about a **pure function**?



Why integration?

- The code we write is **integrated** with an **engine, libraries...**
- The **state** of these systems **influences results** and **context**



Why integration?

- If we don’t test **the whole thing**, we cannot ensure it **works correctly**
- We *have to* write **integration tests**

How do I write integration tests?

GIVEN

One or more **units**

WHEN

Act on a unit

THEN

Check that system changed as expected

How do I write integration tests?

GIVEN

Prepare the **scene**

WHEN

Call a **method**

THEN

Assertions on the Nodes

How do I write integration tests?

```gdscript
var sut: PackedScene = load("res://modules/characters/gato/gato.tscn")

func test_state_machine_exists():
    #GIVEN
    var gato: Node = add_child_autofree(sut.instantiate())

    #WHEN
    var state_machine: Node = gato.find_child("StateMachine")

    #THEN
    assert_not_null(state_machine, "StateMachine does not exists")
```

How do I write integration tests?

```gdscript
const ACTION_NAME := "act"
const TEMP_FILE := "user://temp.txt"
const FILE_CONTENT := '{"control_schemes":[{"input_actions":[],"name":"one","toggle_joystick":f
func test_reset_changes() -> void:
    # GIVEN
    InputMap.add_action(ACTION_NAME)

    var file := FileAccess.open(TEMP_FILE, FileAccess.WRITE)
    file.store_string(FILE_CONTENT)
    file.close()

    var storage_stub = load("res://addons/input_remapper/service/storage_service.gd").new()
    storage_stub.settings_file = TEMP_FILE
    var input_remapper = sut.new(storage_stub)

    # simulate user changes
    var input_action = load("res://addons/input_remapper/model/input_action_button.gd")\
        .new(ACTION_NAME, Helper.create_input_event(KEY_0))
    input_remapper.get_current_scheme().input_actions.append(input_action)

    # WHEN
    input_remapper.reset_changes()

    # THEN
    assert_eq(input_remapper.get_current_scheme().name, "one")
    assert_eq(input_remapper.get_current_scheme().input_actions.size(), 0)
    assert_true(InputMap.has_action(ACTION_NAME))
    var events := InputMap.action_get_events(ACTION_NAME)
    assert_eq(events.size(), 0, "The action has no input events")
    input_remapper.free()

    InputMap.erase_action(ACTION_NAME)
    DirAccess.remove_absolute(TEMP_FILE)
```

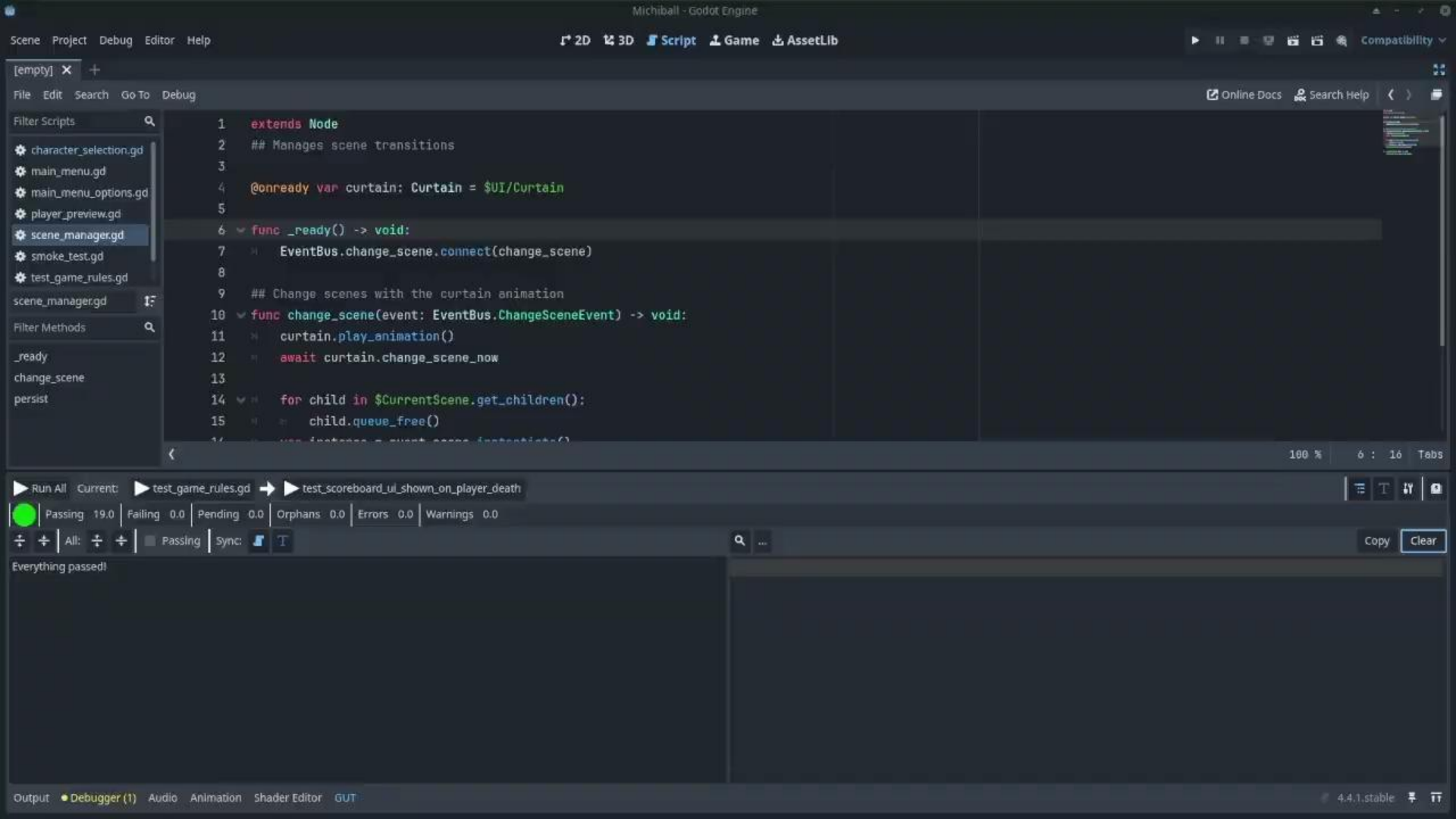
How do I write integration tests?

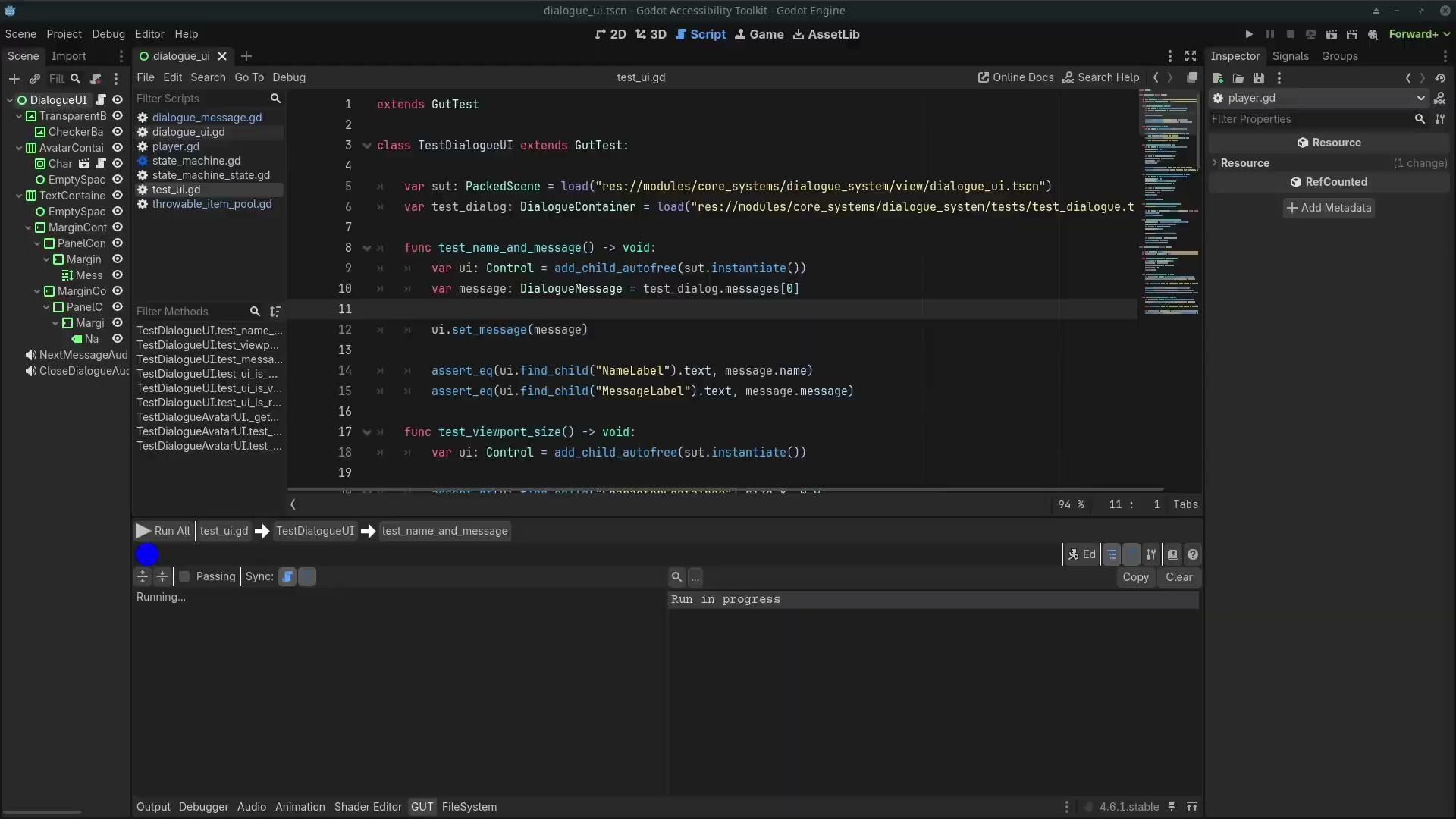
GIVEN

WHEN

THEN

```gdscript
const ACTION_NAME := "act"
const TEMP_FILE := "user://temp.txt"
const FILE_CONTENT := '{"control_schemes":[{"input_actions":[],"name":"one","toggle_joystick":f
func test_reset_changes() -> void:
    # GIVEN
    InputMap.add_action(ACTION_NAME)

    var file := FileAccess.open(TEMP_FILE, FileAccess.WRITE)
    file.store_string(FILE_CONTENT)
    file.close()

    var storage_stub = load("res://addons/input_remapper/service/storage_service.gd").new()
    storage_stub.settings_file = TEMP_FILE
    var input_remapper = sut.new(storage_stub)

    # simulate user changes
    var input_action = load("res://addons/input_remapper/model/input_action_button.gd")\
        .new(ACTION_NAME, Helper.create_input_event(KEY_0))
    input_remapper.get_current_scheme().input_actions.append(input_action)

    # WHEN
    input_remapper.reset_changes()

    # THEN
    assert_eq(input_remapper.get_current_scheme().name, "one")
    assert_eq(input_remapper.get_current_scheme().input_actions.size(), 0)
    assert_true(InputMap.has_action(ACTION_NAME))
    var events := InputMap.action_get_events(ACTION_NAME)
    assert_eq(events.size(), 0, "The action has no input events")
    input_remapper.free()

    InputMap.erase_action(ACTION_NAME)
    DirAccess.remove_absolute(TEMP_FILE)
```


Michiball - Godot Engine
Scene Project Debug Editor Help
2D 3D Script Game AssetLib
Compatibility
[empty]
File Edit Search Go To Debug
Online Docs Search Help
Filter Scripts
character_selection.gd
main_menu.gd
main_menu_options.gd
player_preview.gd
scene_manager.gd
smoke_test.gd
test_game_rules.gd
scene_manager.gd
Filter Methods
_ready
change_scene
persist
1 extends Node
2 ## Manages scene transitions
3
4 @onready var curtain: Curtain = $UI/Curtain
5
6 func _ready() -> void:
7 EventBus.change_scene.connect(change_scene)
8
9 ## Change scenes with the curtain animation
10 func change_scene(event: EventBus.ChangeSceneEvent) -> void:
11 curtain.play_animation()
12 await curtain.change_scene_now
13
14 for child in $CurrentScene.get_children():
15 child.queue_free()
100 %
6 : 16
Tabs
Run All
Current:
test_game_rules.gd
test_scoreboard_ui_shown_on_player_death
Passing 19.0
Failing 0.0
Pending 0.0
Orphans 0.0
Errors 0.0
Warnings 0.0
All:
Passing
Sync:
Copy
Clear
Everything passed!
Output
Debugger (1)
Audio
Animation
Shader Editor
GUT
4.4.1.stable


dialogue_ui.tscn - Godot Accessibility Toolkit - Godot Engine
Scene Project Debug Editor Help
2D 3D Script Game AssetLib
Forward+
Scene Import
dialogue_ui
File Edit Search Go To Debug
test_ui.gd
Online Docs Search Help
Inspector Signals Groups
player.gd
Filter Properties
Resource
Resource (1 change)
RefCounted
Add Metadata
Filter Scripts
dialogue_message.gd
dialogue_ui.gd
player.gd
state_machine.gd
state_machine_state.gd
test_ui.gd
throwable_item_pool.gd
Filter Methods
TestDialogueUI.test_name_...
TestDialogueUI.test_viewp...
TestDialogueUI.test_messa...
TestDialogueUI.test_ui_is_...
TestDialogueUI.test_ui_is_v...
TestDialogueUI.test_ui_is_r...
TestDialogueAvatarUI._get...
TestDialogueAvatarUI.test_...
TestDialogueAvatarUI.test_...
1 extends GutTest
2
3 class TestDialogueUI extends GutTest:
4
5 var sut: PackedScene = load("res://modules/core_systems/dialogue_system/view/dialogue_ui.tscn")
6 var test_dialog: DialogueContainer = load("res://modules/core_systems/dialogue_system/tests/test_dialogue.t
7
8 func test_name_and_message() -> void:
9 var ui: Control = add_child_autofree(sut.instantiate())
10 var message: DialogueMessage = test_dialog.messages[0]
11
12 ui.set_message(message)
13
14 assert_eq(ui.find_child("NameLabel").text, message.name)
15 assert_eq(ui.find_child("MessageLabel").text, message.message)
16
17 func test_viewport_size() -> void:
18 var ui: Control = add_child_autofree(sut.instantiate())
19
94 % 11 : 1 Tabs
Run All test_ui.gd TestDialogueUI test_name_and_message
Passing Sync:
Running...
Copy Clear
Run in progress
Output Debugger Audio Animation Shader Editor GUT FileSystem
4.6.1.stable

Test runners

GUT by Butch Wesley (link)

gdUnit4 by Mike Schulze (link)

GoDotTest by Chickensoft Games (link)

... or write your own!

More test types!

- Tests for **specific issues**
- Find an issue and cover it with a test
- Goal: prevent something bad from happening **without your knowledge**
- 100% coverage isn’t helpful

Screenshot test

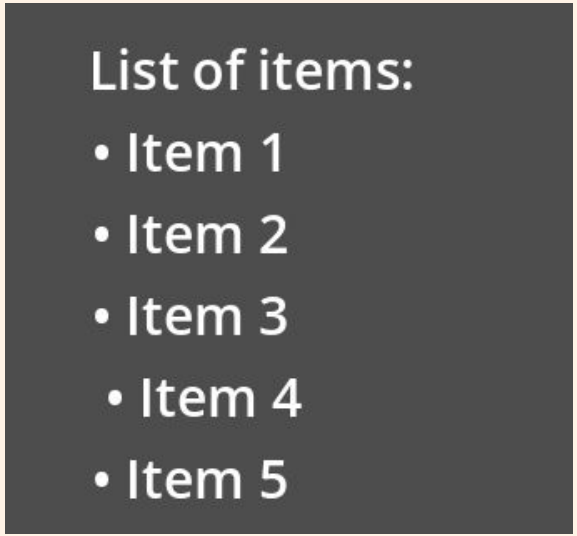
Compare screenshots to find differences

ORIGINAL SCREENSHOT

Screenshot test

What is the difference between these 2 images?

ORIGINAL SCREENSHOT



TEST SCREENSHOT

Screenshot test

What is the difference between these 2 images?

ORIGINAL SCREENSHOT

DIFFERENCE

■ Missing

■ New

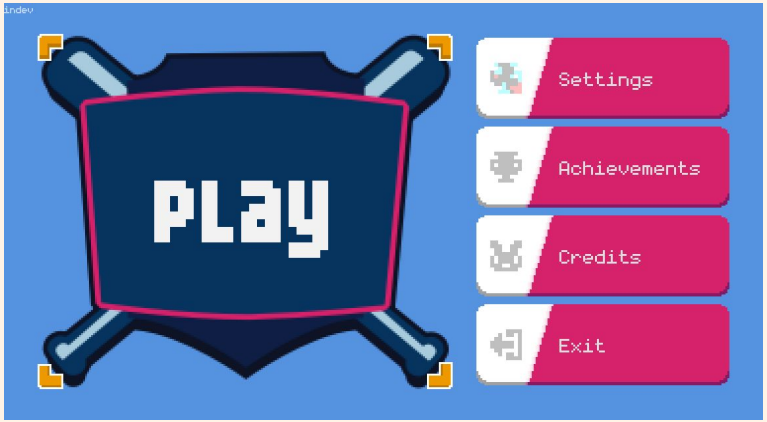
Screenshot test

What for?

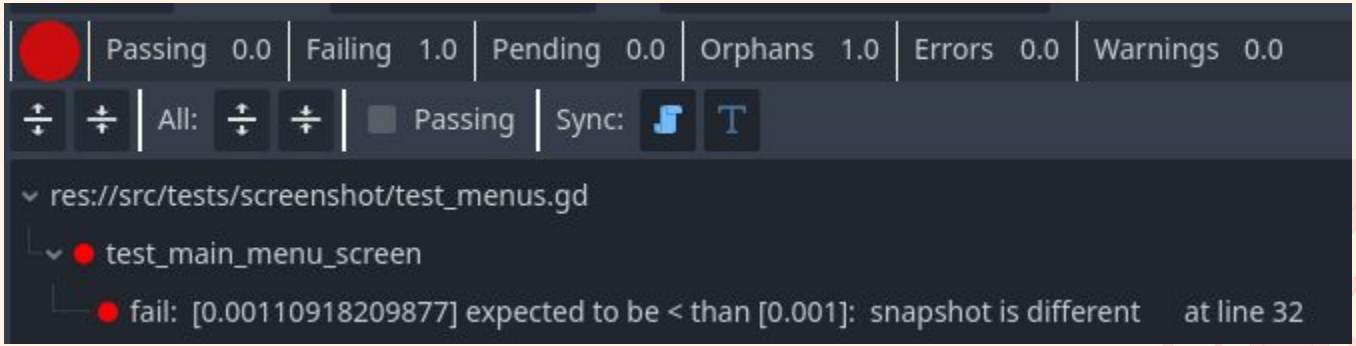
- UI testing
- Texture checks
- Shader testing

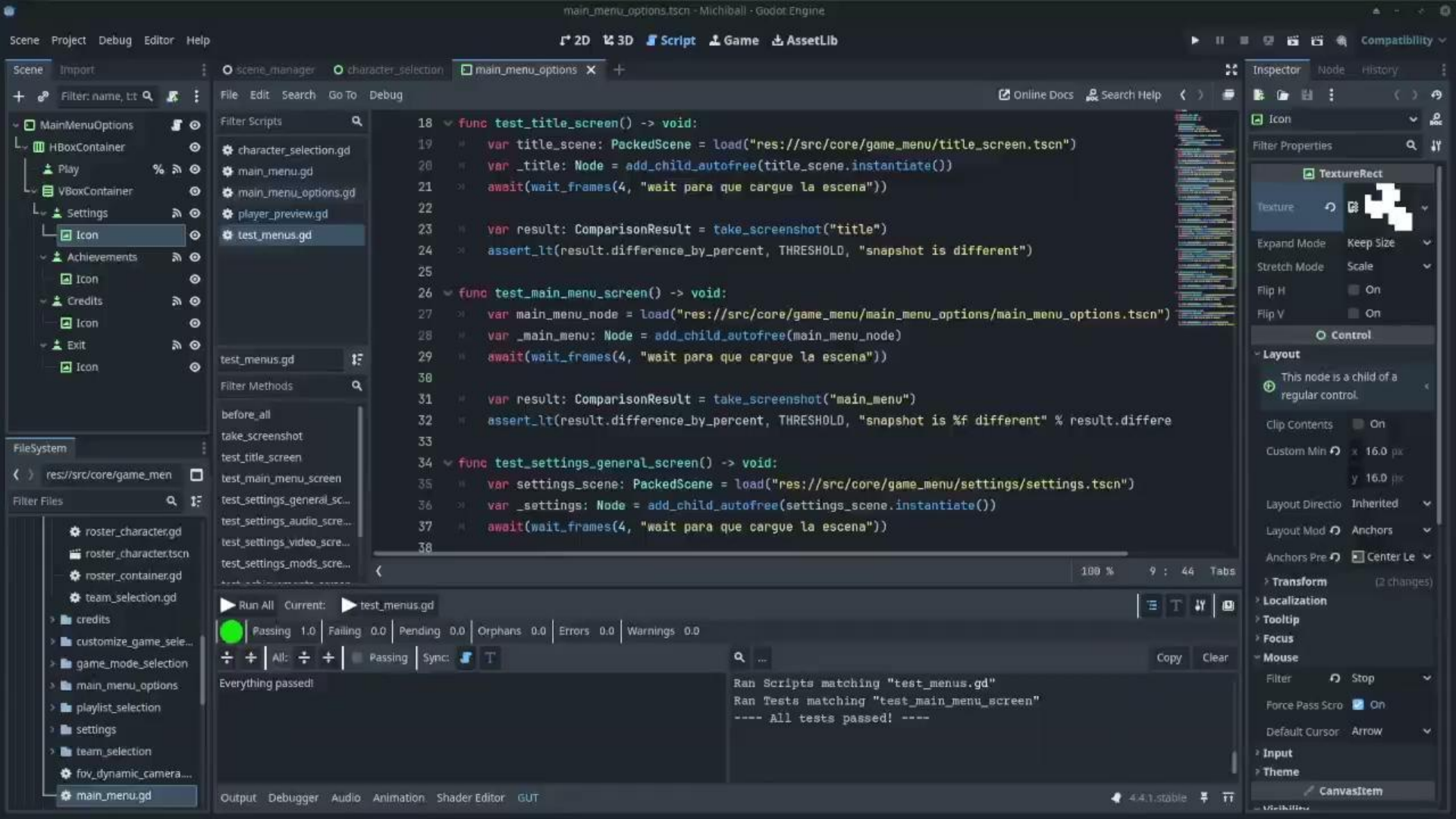
Character selection screen, Michiball

Screenshot test



Main menu, Michiball



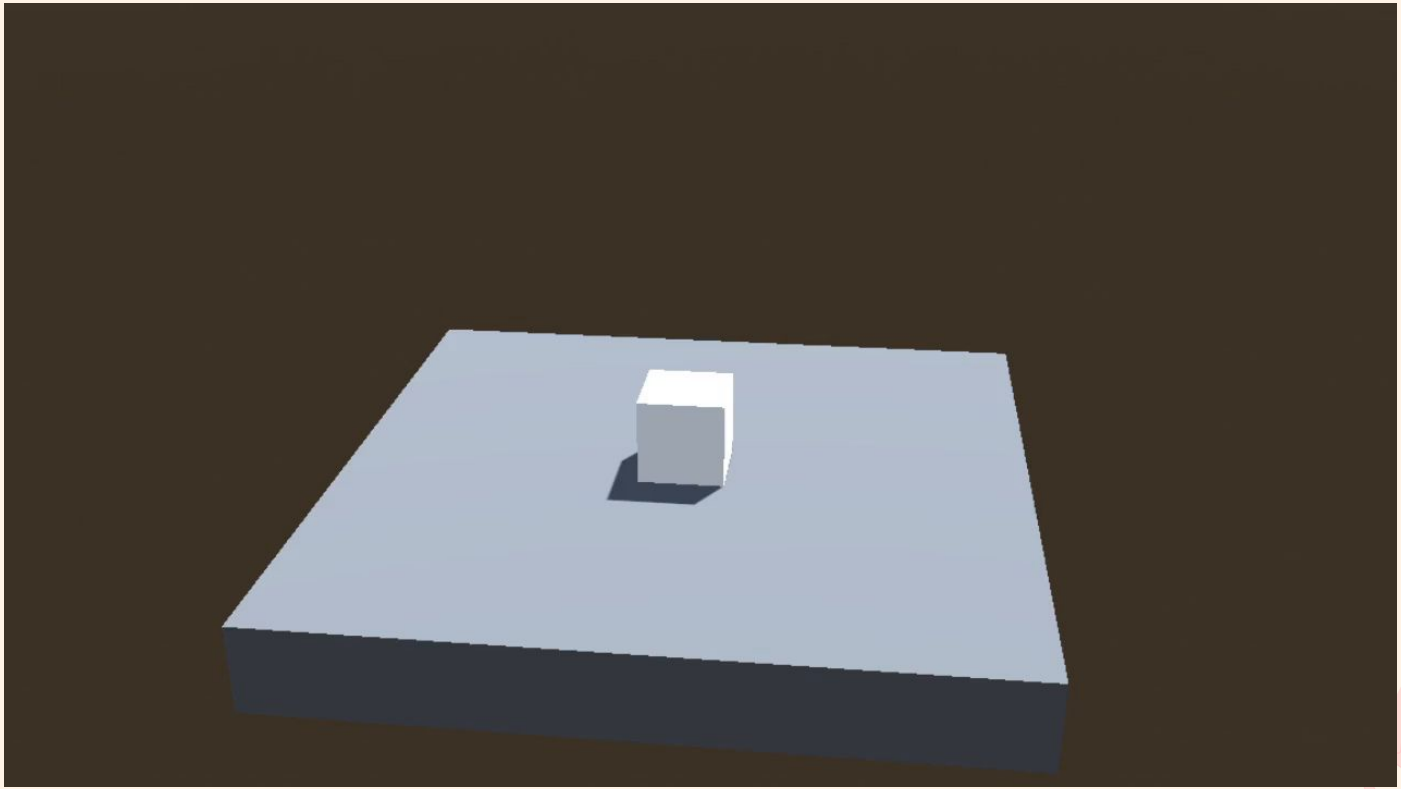

main_menu_options.tscn - Michiball - Godot Engine
Scene Project Debug Editor Help
2D 3D Script Game AssetLib
Compatibility
Scene Import
MainMenuOptions
HBoxContainer
Play
VBoxContainer
Settings
Icon
Achievements
Icon
Credits
Icon
Exit
Icon
FileSystem
res://src/core/game_men
Filter Files
roster_character.gd
roster_character.tscn
roster_container.gd
team_selection.gd
credits
customize_game_sele...
game_mode_selection
main_menu_options
playlist_selection
settings
team_selection
fov_dynamic_camera....
main_menu.gd
scene_manager
character_selection
main_menu_options
File Edit Search Go To Debug
Online Docs
Search Help
Filter Scripts
character_selection.gd
main_menu.gd
main_menu_options.gd
player_preview.gd
test_menus.gd
test_menus.gd
Filter Methods
before_all
take_screenshot
test_title_screen
test_main_menu_screen
test_settings_general_sc...
test_settings_audio_scre...
test_settings_video_scre...
test_settings_mods_scre...
18 func test_title_screen() -> void:
19 var title_scene: PackedScene = load("res://src/core/game_menu/title_screen.tscn")
20 var _title: Node = add_child_autofree(title_scene.instantiate())
21 await(wait_frames(4, "wait para que cargue la escena"))
22
23 var result: ComparisonResult = take_screenshot("title")
24 assert_lt(result.difference_by_percent, THRESHOLD, "snapshot is different")
25
26 func test_main_menu_screen() -> void:
27 var main_menu_node = load("res://src/core/game_menu/main_menu_options/main_menu_options.tscn")
28 var _main_menu: Node = add_child_autofree(main_menu_node)
29 await(wait_frames(4, "wait para que cargue la escena"))
30
31 var result: ComparisonResult = take_screenshot("main_menu")
32 assert_lt(result.difference_by_percent, THRESHOLD, "snapshot is %f different" % result.differe
33
34 func test_settings_general_screen() -> void:
35 var settings_scene: PackedScene = load("res://src/core/game_menu/settings/settings.tscn")
36 var _settings: Node = add_child_autofree(settings_scene.instantiate())
37 await(wait_frames(4, "wait para que cargue la escena"))
38
100 % 9 : 44 Tabs
Run All Current: test_menus.gd
Passing 1.0 Failing 0.0 Pending 0.0 Orphans 0.0 Errors 0.0 Warnings 0.0
All: Passing Sync:
Copy Clear
Everything passed!
Ran Scripts matching "test_menus.gd"
Ran Tests matching "test_main_menu_screen"
---- All tests passed! ----
Output Debugger Audio Animation Shader Editor GUT
4.4.1.stable
Inspector Node History
Icon
Filter Properties
TextureRect
Texture
Expand Mode Keep Size
Stretch Mode Scale
Flip H On
Flip V On
Control
Layout
This node is a child of a regular control.
Clip Contents On
Custom Min x 16.0 px
y 16.0 px
Layout Directio Inherited
Layout Mod Anchors
Anchors Pre Center Le
Transform (2 changes)
Localization
Tooltip
Focus
Mouse
Filter Stop
Force Pass Scro On
Default Cursor Arrow
Input
Theme
CanvasItem

Screenshot test

Available as an open source Godot plugin!

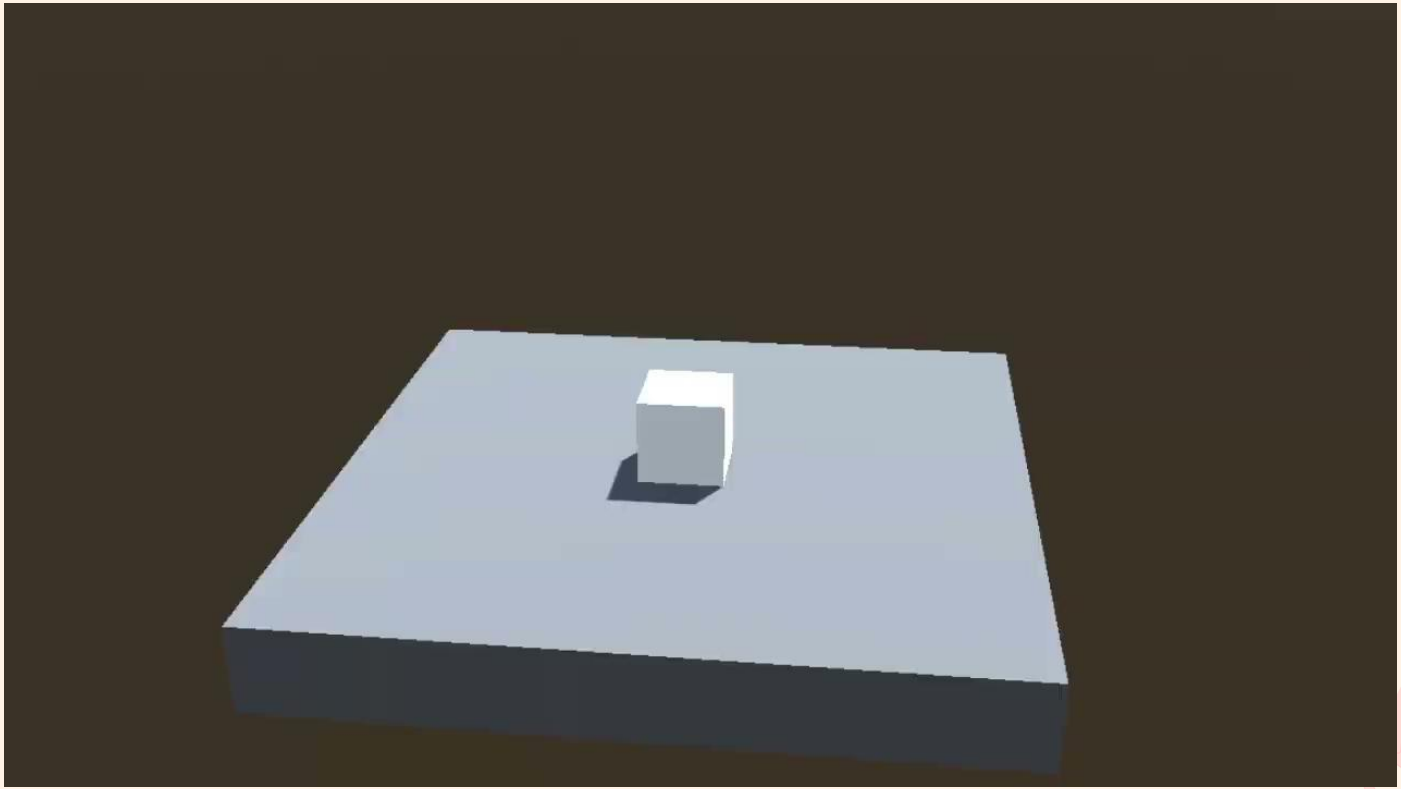
https://github.com/Nokorpo/GDsnap

Performance test/benchmark



https://github.com/Nokorpo/GDsnap

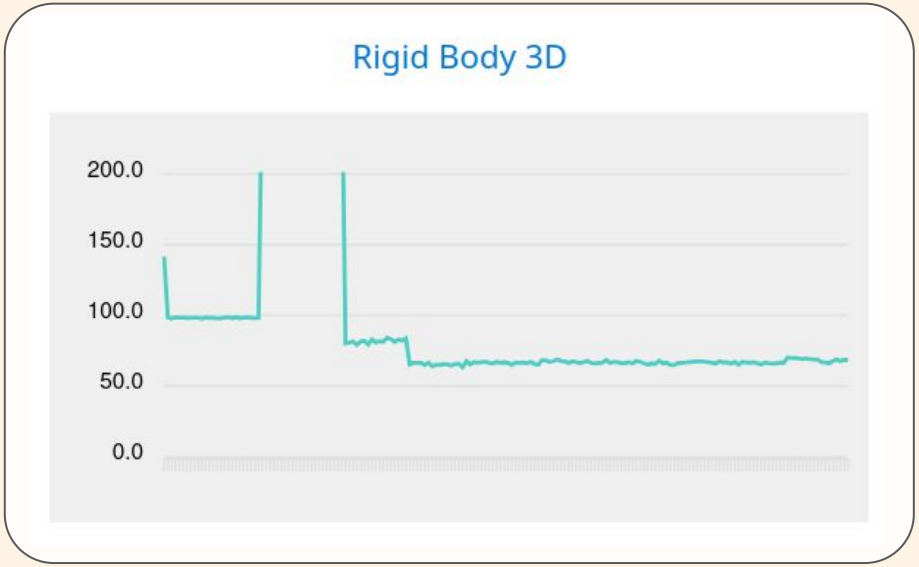
Performance test/benchmark



https://github.com/Nokorpo/GDsnap

Performance test/benchmark

Example: https://benchmarks.godotengine.org/



Godot Engine Benchmarks

Performance test/benchmark

What for?

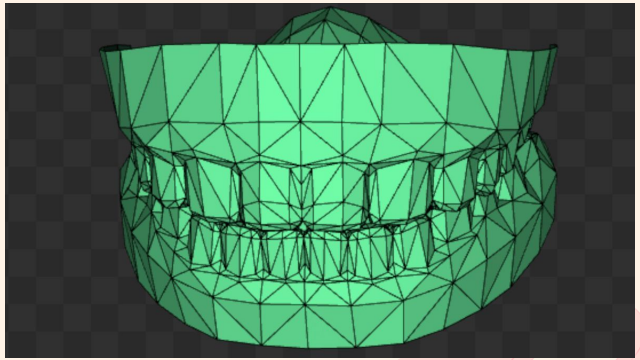
- Track resource usage evolution during a level’s development

Blighttown,
Dark Souls

Performance test/benchmark

What for?

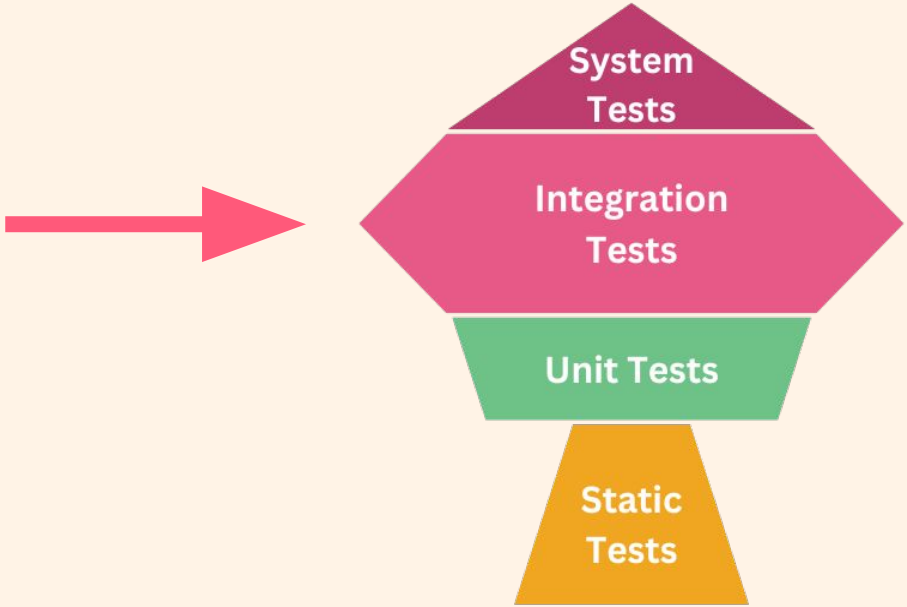
- Prevent commonly repeated objects from causing trouble



↑ NPC teeth Mesh, Cities Skylines 2

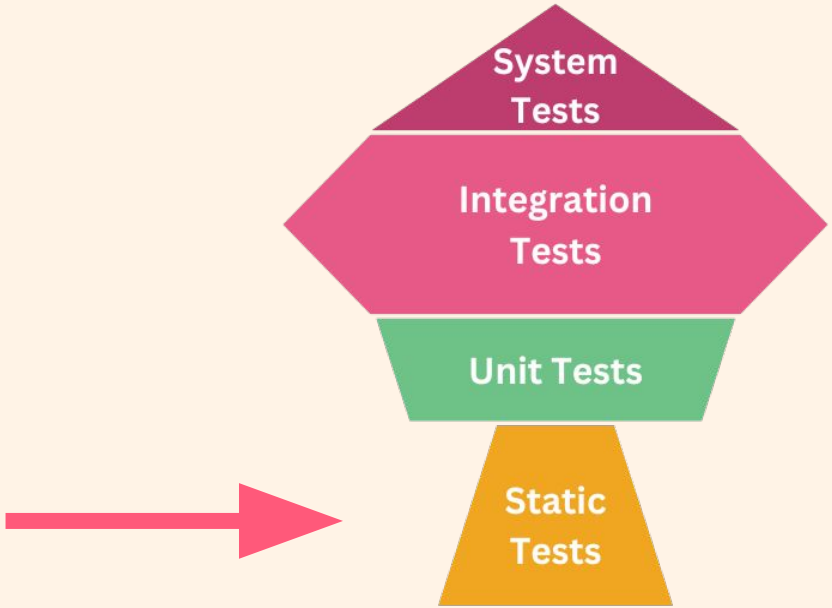
← Cities Skylines 2 gameplay

More test types!



Trophy testing model, Kent C. Dodds

Static analysis



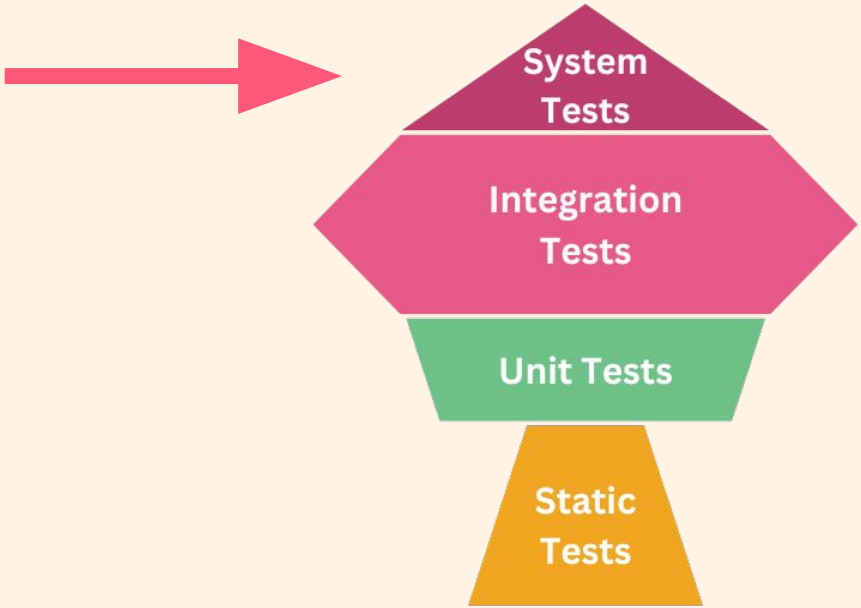
Trophy testing model, Kent C. Dodds

Static analysis

- Game isn’t executed, the test just reads its files
- Very useful and **almost free**. Examples:
  - Compilation errors and linters
  - Script that checks file naming, config files...

```
Icon2.svg: wrong file or folder name. Please, keep everything in lower case.
Icon2.svg.import: wrong file or folder name. Please, keep everything in lower case.
src/GameManager/game.tscn: wrong file or folder name. Please, keep everything in lower case.
src/input/WiiMoteInput.gd: wrong file or folder name. Please, keep everything in lower case.
```

Smoke test



Trophy testing model, Kent C. Dodds

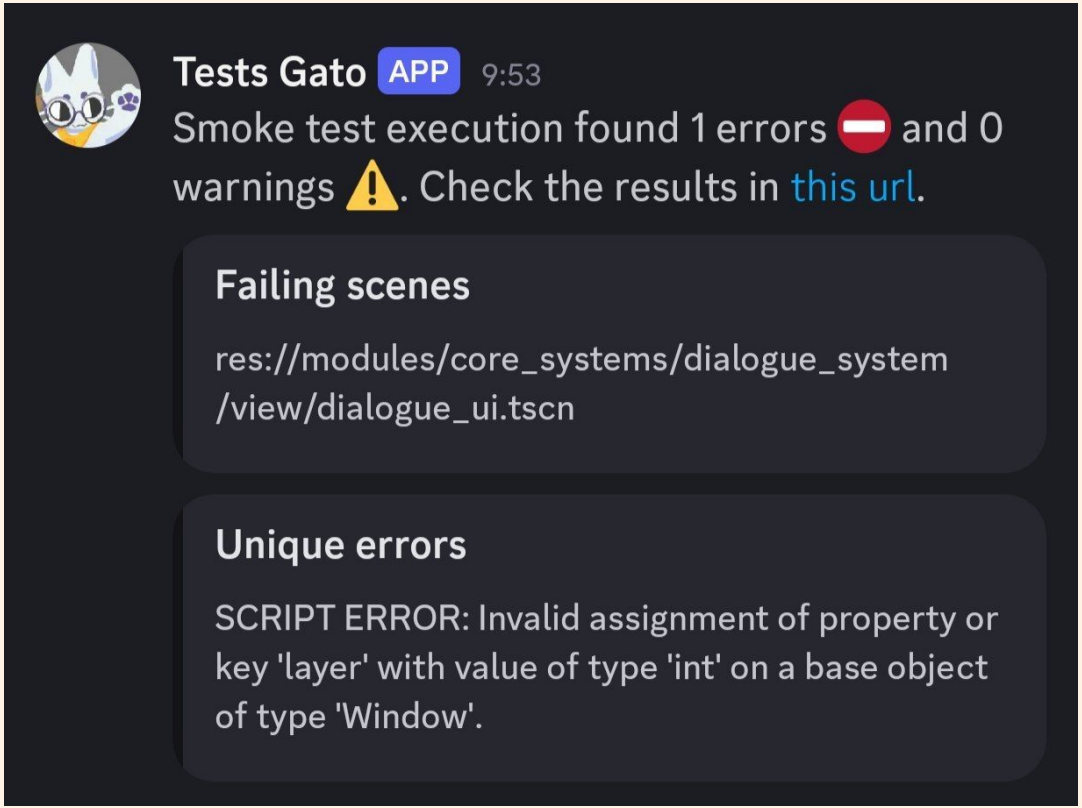
Smoke test

Smoke test

Open game scenes and check whether they have errors

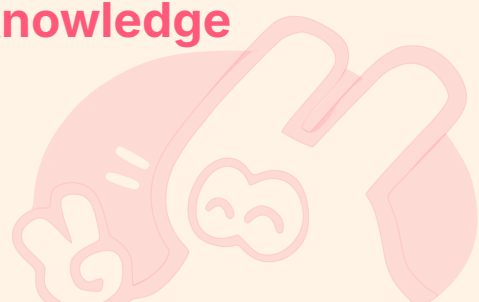
```
74   - Testing res://src/core/game_menu/character_selection/character_selection.tscn...
75   - Testing res://src/core/game_menu/character_selection/player_selection_preview.tscn...
76   - Testing res://src/core/game_menu/character_selection/roster_character.tscn...
77   - Testing res://src/core/game_menu/credits/credits.tscn...
78   - Testing res://src/core/game_menu/customize_game_selection/game_mode_selection.tscn...
79 SCRIPT ERROR: Parse Error: Invalid argument for "new()" function: argument 4 should be "bool" but is "Resource".
80           at: GDScript::reload (res://src/core/game_menu/game_mode_selection/game_mode_selection.gd:22)
81 ERROR: Failed to load script "res://src/core/game_menu/game_mode_selection/game_mode_selection.gd" with error "Parse
   error".
82     at: load (modules/gdscript/gdscript.cpp:3022)
83   - Testing res://src/core/game_menu/game_mode_selection/game_mode_selection.tscn...
```

Smoke test


Tests Gato APP 9:53
Smoke test execution found 1 errors ⛔ and 0 warnings ⚠️. Check the results in this url.
Failing scenes
res://modules/core_systems/dialogue_system/view/dialogue_ui.tscn
Unique errors
SCRIPT ERROR: Invalid assignment of property or key 'layer' with value of type 'int' on a base object of type 'Window'.

Summary

- Write tests. Not too many. Mostly **integration**
- Tests for **specific issues**
  - Static analysis
  - Smoke test
  - Screenshot test
  - Performance test
  - ...
- Prevent something bad from happening **without your knowledge**



Part 3

How to automate?

Hypothetical

Who will make a **more consistent** coffee?

- Pablo, college student, brews 1L every 2-3 months when he has to cram, buys whatever coffee the supermarket has in store...
- A coffee pod machine

Manual processes

- A manual process will eventually fail
  - Interruptions, forgetting a step, mistakes
  - Blue screen, power outage, program update broke compatibility
  - ... and more
- Avoid manual processes
  - Or make containment plans (eg. checklist, redundancy)

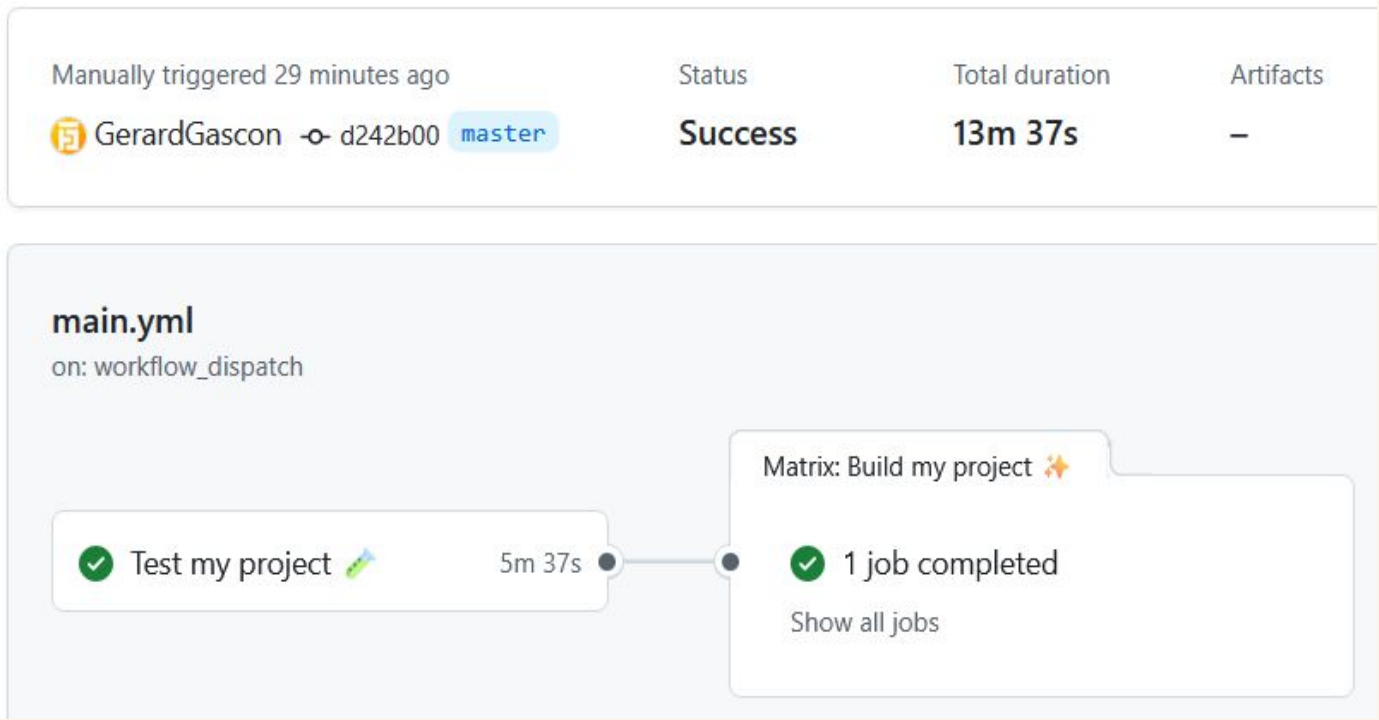
What processes can be automated?

- Running **tests**!
- **Builds** (Android/iOS, Web, Windows/Mac/Linux)
- Uploading a build to a **store** (itch.io, Steam, App Stores...)
- Tasks that only 1 person in the team knows how to do
- Tasks that must happen every X time
  - (eg. clean up a directory, generate a changelog, generate documentation)
- Image/video conversion to a specific format/resolution
- ...

Continuous Integration (CI)

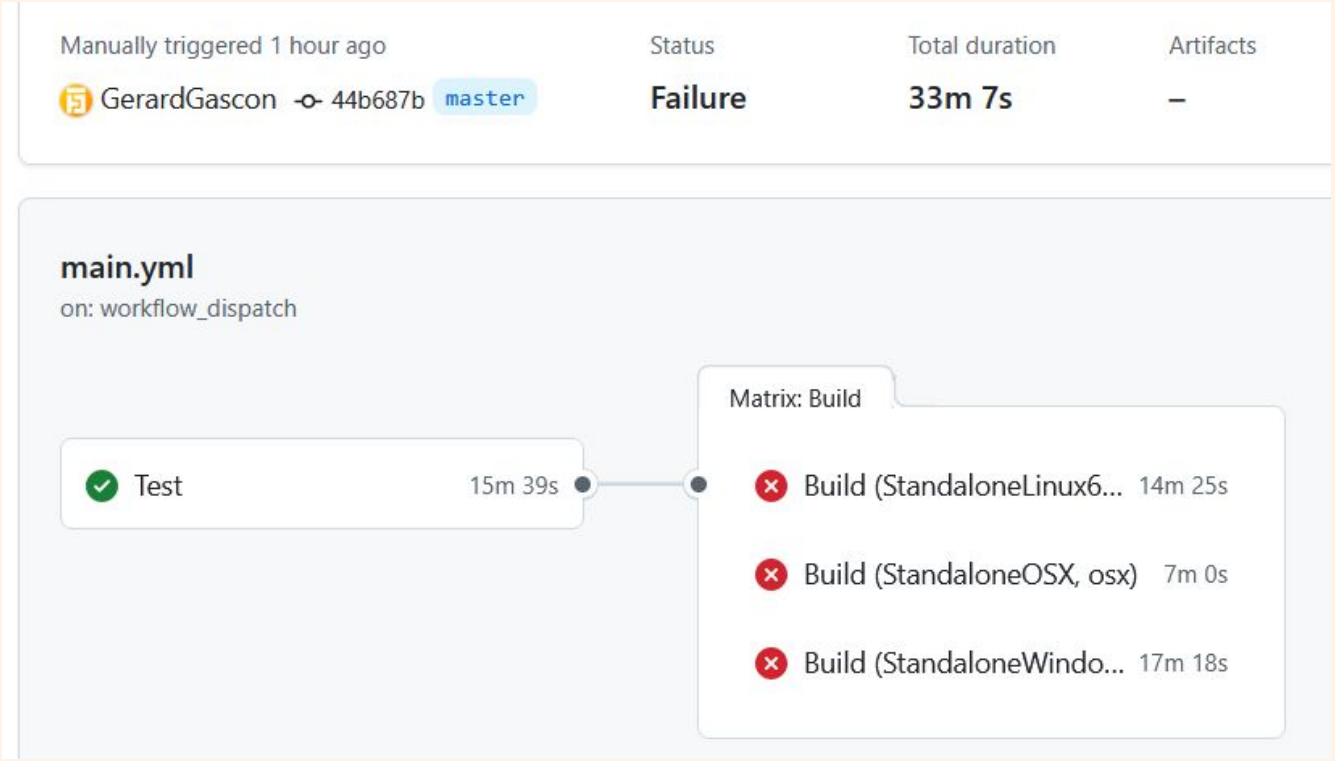
- Run scripts in **response to**:
  - Commit push
  - Pull request merge
  - Every X time (cron)
  - Press of a button
- Automation for the **whole team**
- Iterate **faster**, more **consistently**

CI with Unity

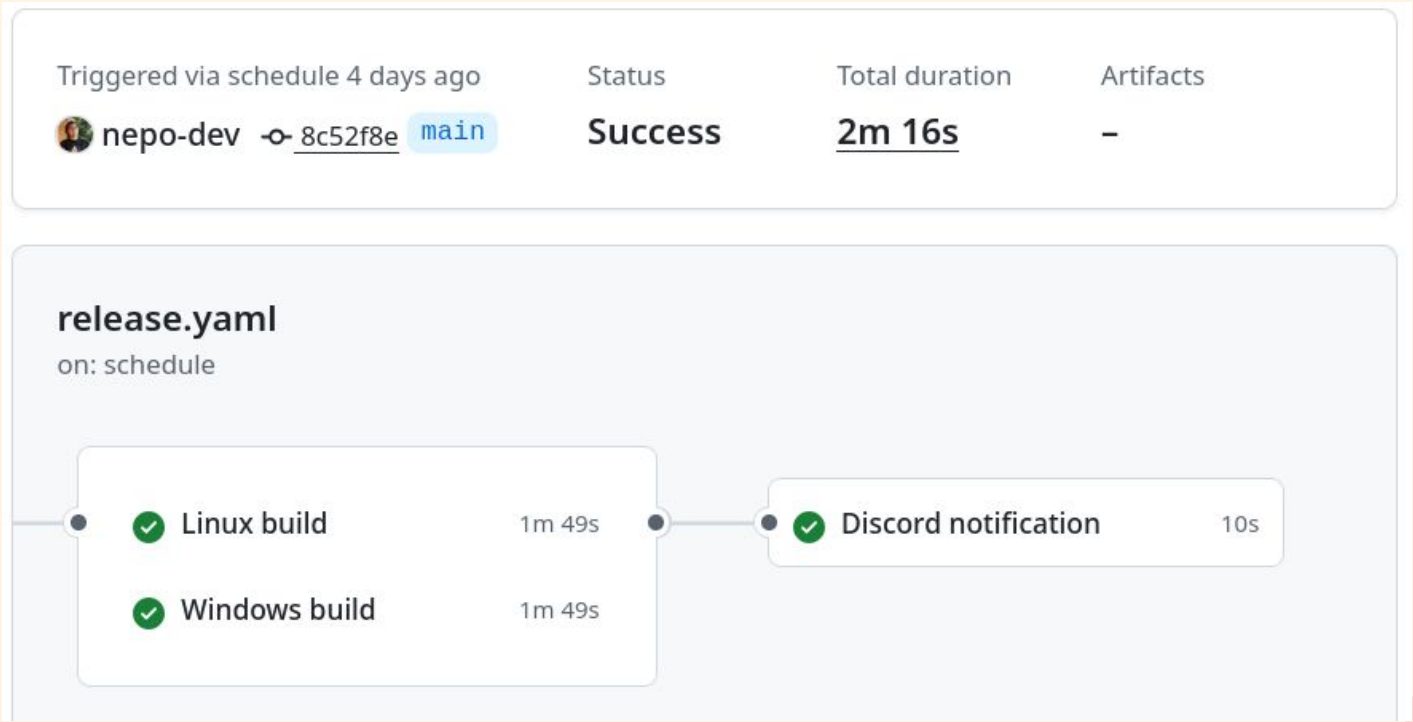


Unity

CI with Unity



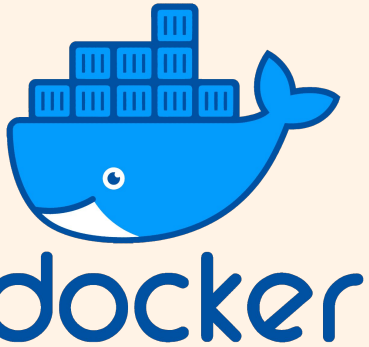
CI with Godot



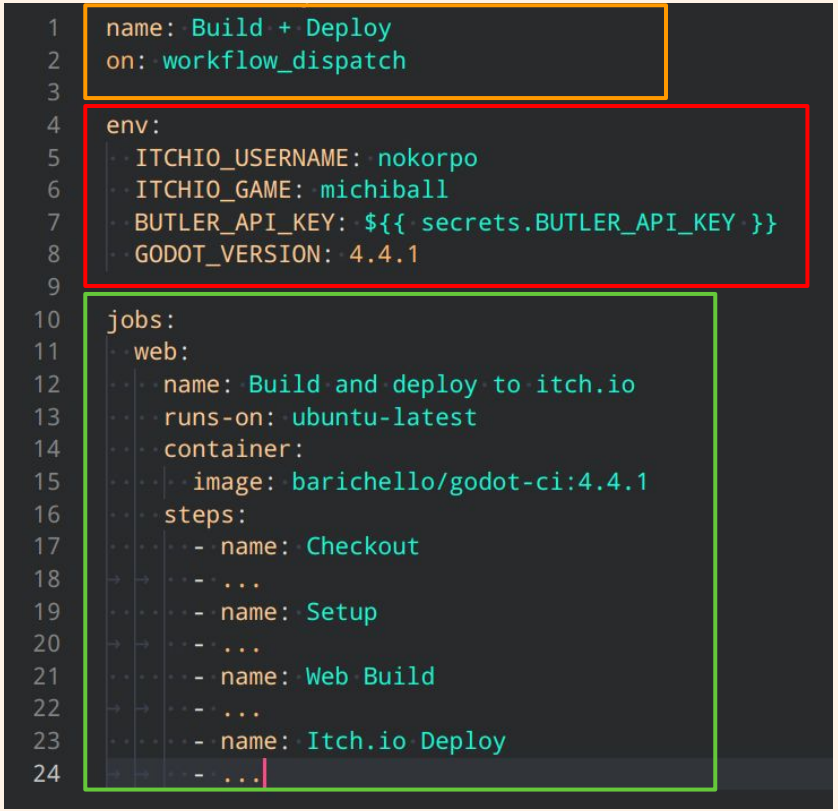
Continuous Integration (CI)

```yaml
name: Build + Deploy
on: workflow_dispatch

env:
  ITCHIO_USERNAME: nokorpo
  ITCHIO_GAME: michiball
  BUTLER_API_KEY: ${{ secrets.BUTLER_API_KEY }}
  GODOT_VERSION: 4.4.1

jobs:
  web:
    name: Build and deploy to itch.io
    runs-on: ubuntu-latest
    container:
      image: barichello/godot-ci:4.4.1
    steps:
      - name: Checkout
        uses: actions/checkout@v2
        with:
          lfs: true
      - name: Setup
        run: |
          mkdir -v -p ~/.local/share/godot/export_templates
          mv /root/.local/share/godot/templates/${GODOT_VERSION}.stable
          ~/.local/share/godot/export_templates/${GODOT_VERSION}.stable
      - name: Web Build
        run: |
          mkdir build
          godot --export-release --headless "Web" ./build/index.html
      - name: Itch.io Deploy
        run: |
          ls ./build
          butler push ./build $ITCHIO_USERNAME/$ITCHIO_GAME:html5
```

(Actions)



Continuous Integration (CI)



```yaml
name: Build + Deploy
on: workflow_dispatch

env:
  ITCHIO_USERNAME: nokorpo
  ITCHIO_GAME: michiball
  BUTLER_API_KEY: ${{ secrets.BUTLER_API_KEY }}
  GODOT_VERSION: 4.4.1

jobs:
  web:
    name: Build and deploy to itch.io
    runs-on: ubuntu-latest
    container:
      image: barichello/godot-ci:4.4.1
    steps:
      - name: Checkout
      - ...
      - name: Setup
      - ...
      - name: Web Build
      - ...
      - name: Itch.io Deploy
      - ...
```

← When does it run

← Variables:

Names, passwords, API keys...

← Steps to run

Continuous Integration (CI)

```yaml
steps:
  - name: Checkout
    uses: actions/checkout@v2
    with:
      lfs: true
  - name: Setup
    run: |
      mkdir -v -p ~/.local/share/godot/export_templates
      mv /root/.local/share/godot/templates/${GODOT_VERSION}.stable
      ~/.local/share/godot/export_templates/${GODOT_VERSION}.stable
  - name: Web Build
    run: |
      mkdir build
      godot --export-release --headless "Web" ./build/index.html
  - name: Itch.io Deploy
    run: |
      ls ./build
      butler push ./build $ITCHIO_USERNAME/$ITCHIO_GAME:html5
```

← Download project

← Setup Godot

← Generate web build

← Upload to itch.io

Continuous Integration (CI)

	Execution time	Storage
Github	1.000 min/mo	1 GB
CircleCI	2.000 min/mo	2 GB
Itch.io	–	30 GB

Summary

- Objective: find defects **as soon as possible**
- Test and **iterate** processes to achieve **continuous improvement**
- **Write tests**. Not too many. Mostly **integration**.
  - Tests for **specific issues**
  - Prevent something bad from happening **without your knowledge**
- Avoid human error and bottlenecks with **automation**

Questions?

https://nokorpo.com

- Slides
- More information
- Examples
- Relevant links