



DHBW

Duale Hochschule
Baden-Württemberg



Tower Defense

Pigs vs Mushrooms

Agenda

1. Projektziel
2. Architektur
3. Entwurfsmuster
4. Technologien
5. Test Coverage
6. CI/CD Setup
7. Live-Demo
8. Lessons learned



Projektziel

- Neues Tower Defense Spiel
- Kein Spiel Imitieren
- Eigene Designs
- Lore für unser Game -> Siehe Github



Architektur

- Basiert auf Event-driven Architektur mit Mediator-Topologie
- Unity weil:
 - Populärste Game Engine für mobile Entwicklung
- Wie ist unser Unity Projekt aufgebaut?
 - Szenen für Menüs und Level
 - Event-Handler
 - Logik über Skripte



- Skripte folgen dem Single Responsibility Principle
- Vererbung und Schnittstellen
- Nutzung von Third-Party-Libraries



Technologien

- Unity



- C#



- YouTrack



- GitHub

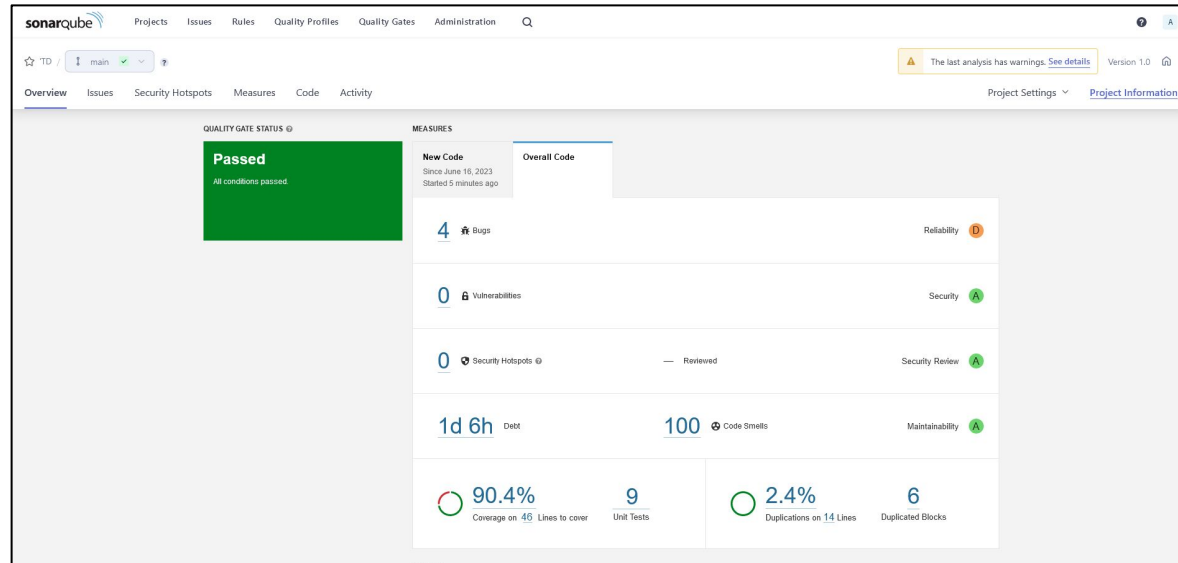


- SonarQube



Test Coverage

- C# Skripte werden in .NET Core durch Unit Tests getestet.
- SonarQube wird genutzt um die Test-Abdeckung und weitere Metriken zu überwachen.



CI/CD

- Keine richtige Lösung aufgrund von Lizenzproblemen mit Unity
- Alternative Lösung:

➡ Automatisches Bauen und Ausführen der C# Skripte bei Commits und Pull Requests über GitHub Actions



All workflows

Showing runs from all workflows

Filter workflow runs

10 workflow runs	Event	Status	Branch	Actor
<div><div></div><div>Added project management documents</div><div>.NET Core Desktop #14: Commit 679276 pushed by Freaky2164</div></div>	main	<div><div></div><div>16 hours ago</div><div>1m 11s</div></div>		
<div><div></div><div>Update Test Plan.md</div><div>.NET Core Desktop #13: Commit 521854d pushed by Freaky2164</div></div>	main	<div><div></div><div>16 hours ago</div><div>1m 54s</div></div>		
<div><div></div><div>Added multiple rounds</div><div>.NET Core Desktop #12: Commit 6b540b pushed by lber02</div></div>	main	<div><div></div><div>16 hours ago</div><div>1m 2s</div></div>		
<div><div></div><div>Update Test Evaluation Summary.md</div><div>.NET Core Desktop #11: Commit 850c44 pushed by Freaky2164</div></div>	main	<div><div></div><div>16 hours ago</div><div>1m 11s</div></div>		
<div><div></div><div>fix</div><div>.NET Core Desktop #10: Commit ba088d pushed by WeberHr</div></div>	main	<div><div></div><div>17 hours ago</div><div>1m 23s</div></div>		
<div><div></div><div>fix labels, add upgrades for tower 2</div><div>.NET Core Desktop #9: Commit cd8312 pushed by WeberHr</div></div>	main	<div><div></div><div>17 hours ago</div><div>1m 19s</div></div>		
<div><div></div><div>Updated backgrounds</div><div>.NET Core Desktop #8: Commit af9e6d pushed by Freaky2164</div></div>	main	<div><div></div><div>17 hours ago</div><div>1m 33s</div></div>		
<div><div></div><div>Update README.md</div></div>		<div><div></div><div>17 hours ago</div></div>		

build (Release)
succeeded 15 hours ago in 5m 35s

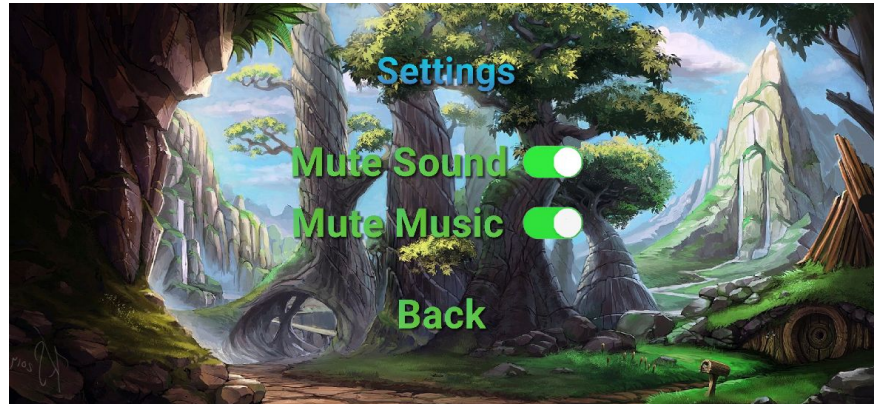
Search logs

> Set up job	1s
> Checkout	35s
> Install .NET Core	35s
> Setup MSBuild.exe	2s
> Execute unit tests	1m 40s
> Post Install .NET Core	0s
> Post Checkout	2s
> Complete job	0s



Live-Demo







Lessons learned

- Striktere GitHub-Rules
- Mehr Druck vom Team um Aufgaben schneller zu erledigen
- C# Best Practices vorher lernen und anwenden
- Task präziser erstellen, um mögliche Unklarheiten früher zu klären





Vielen Dank für eure ungeteilte
Aufmerksamkeit :)

