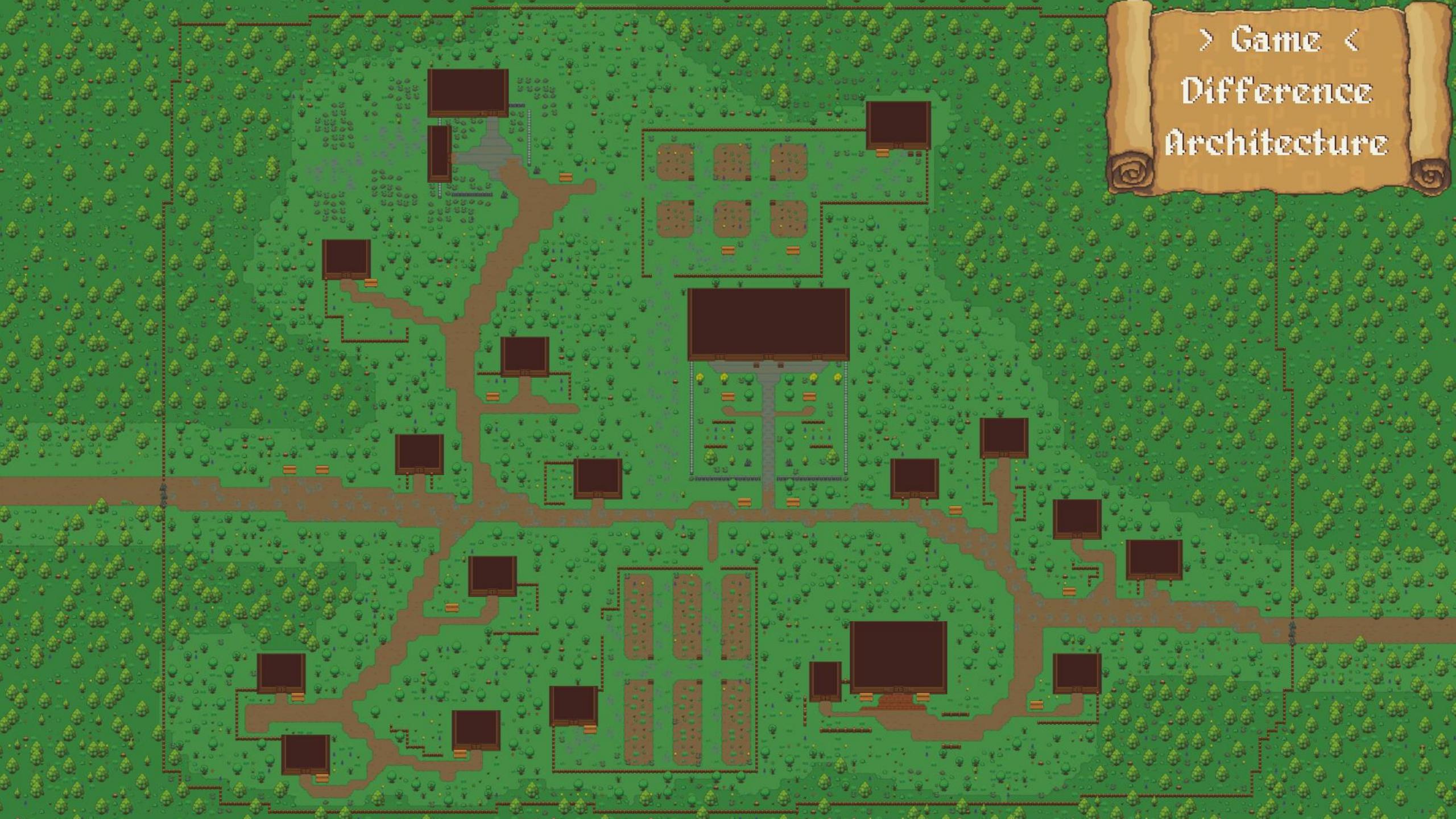


Wittgenst^t



a game by Henrik Weidner,
Oliver Beisacher, Mark Zelenka,
Marvin Pucknat and Alexander Flager

> Game <
Difference
Architecture



> Game <
Difference
Architecture

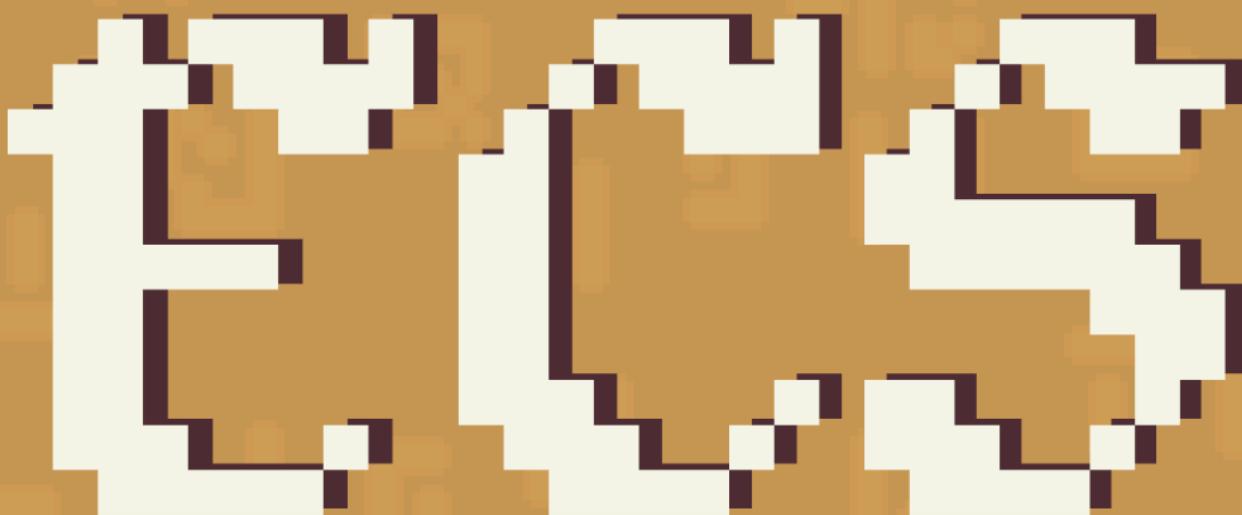


Game

> Difference <
Architecture

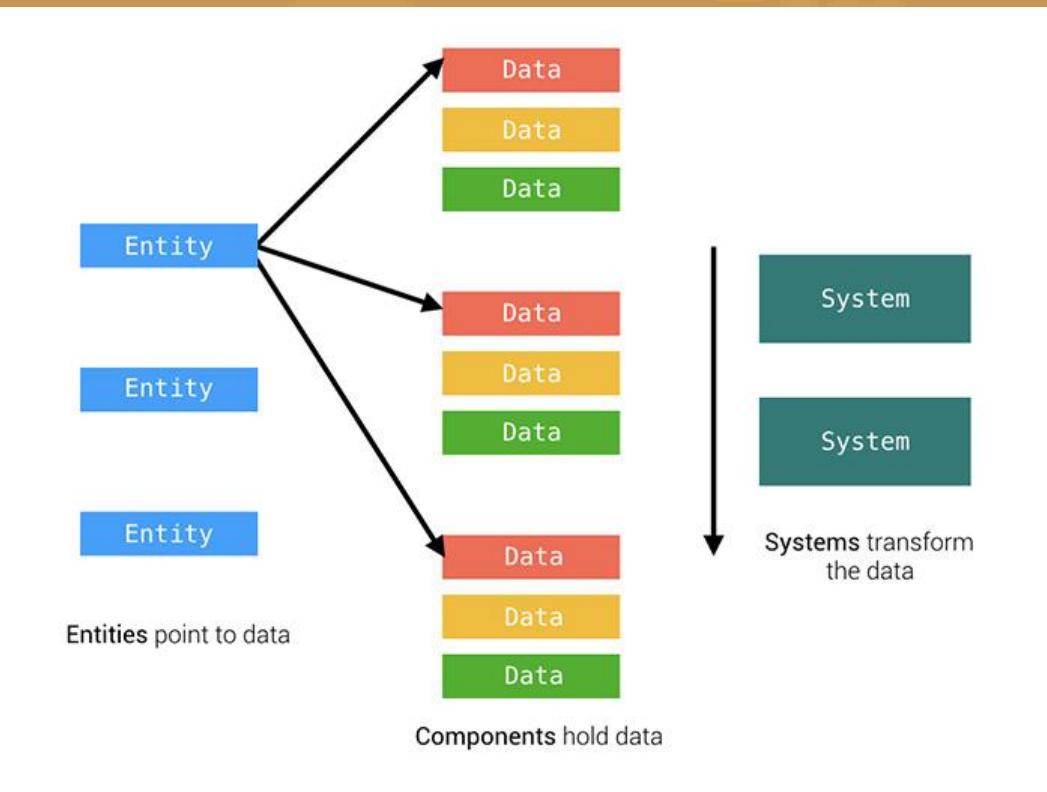
- mehr Arbeit als gedacht
- Keine endgegner

Game
Difference
Architecture <



Game Difference Architecture

ECS





level 1

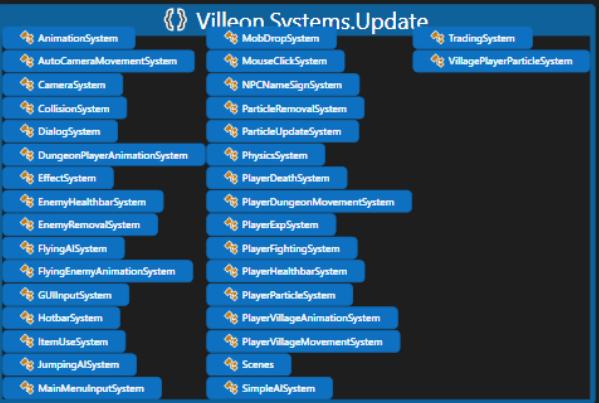


Game Difference Architecture

ECS

System

Entity



Component

Architecture
→ Testing <
Retrospektive

UNIT TEST

Physics

Collision

Physic Tests

```
private Scene _testScene = new Scene("Testing");

[TestInitialize]
public void SetUp()
{
    TypeRegistry.SetupTypes();

    _testScene.AddSystem(new PhysicsSystem("Physics"));
    _testScene.AddSystem(new CollisionSystem("Collision"));
}
```

Physic Tests

```
[TestMethod]
public void EntityHasGravitation()
{
    // Instanciate physics entity
    Entity physicEntity = new Entity("Physic");
    physicEntity.AddComponent(new Physics());
    physicEntity.AddComponent(
        new DynamicCollider(new Vector2(0, 0), new Vector2(0, 0), 1, 1));
```

Physic Tests

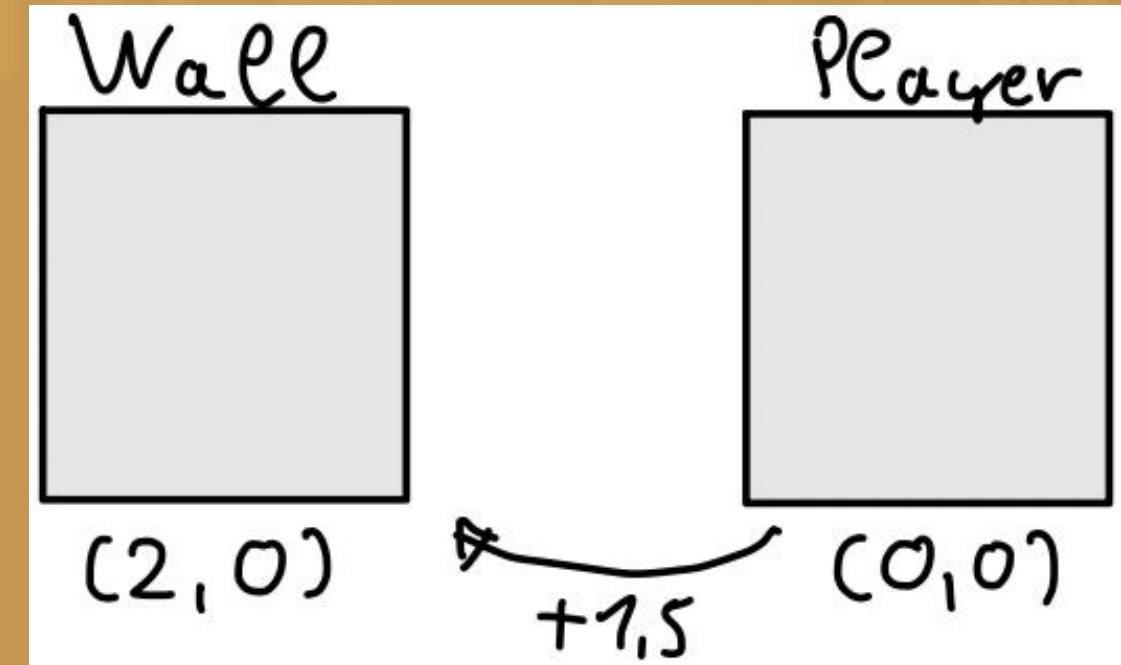
```
// Setup Test-Scene
_testScene.AddEntity(physicEntity);

// Update the PhysicsSystem 5 times with 0.1f time between updates
_testScene.Update(0.5f);

// Free Fall:  $1/2 * g * t^2 = 0.5f * -6 * 0.5^2 = -7.5f$ 
Assert.AreEqual(0.5f * (-Constants.GRAVITY * 3) * (float)Math.Pow(0.5f, 2f),
    physicEntity.GetComponent<Transform>().Position.Y);

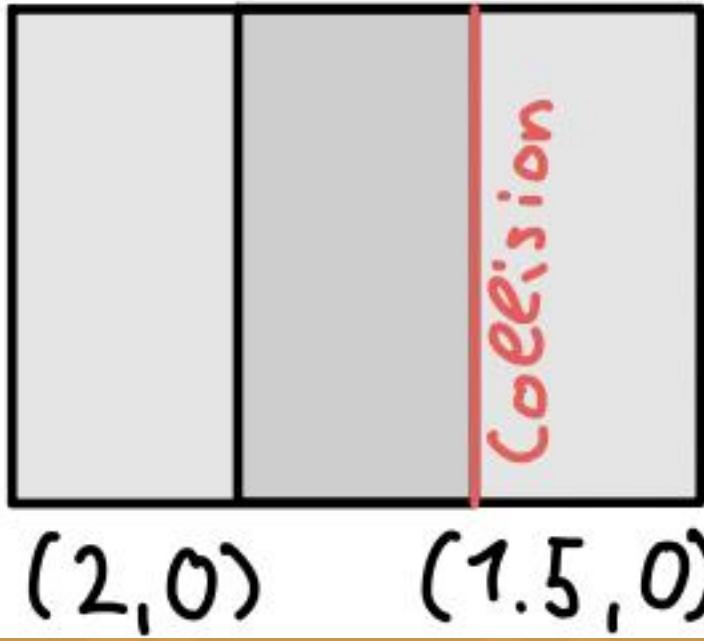
// Clean up
_testScene.RemoveEntity(physicEntity);
```

Collision Tests



Collision Tests

Wall Player



Collision Tests

Wall	Player
(2,0)	(1,0)

Collision Tests

```
[TestMethod]
public void CleanCollisionLeft()
{
    TypeRegistry.SetupTypes();
    CollisionSystem collisionSystem = new ("collisionSystem");

    Vector2 enemyStartPos = new Vector2(2, 0);
    Vector2 playerStartPos = new Vector2(0, 0);

    Entity enemy = new Entity(new Transform(enemyStartPos, 1.0f, 0.0f), "Enemy");
    enemy.AddComponent(new Collider(Vector2.Zero, enemyStartPos, 1f, 1f));

    Entity player = new Entity(new Transform(playerStartPos, 1.0f, 0.0f), "Player");
    player.AddComponent(new DynamicCollider(Vector2.Zero, playerStartPos, 1f, 1f));
```

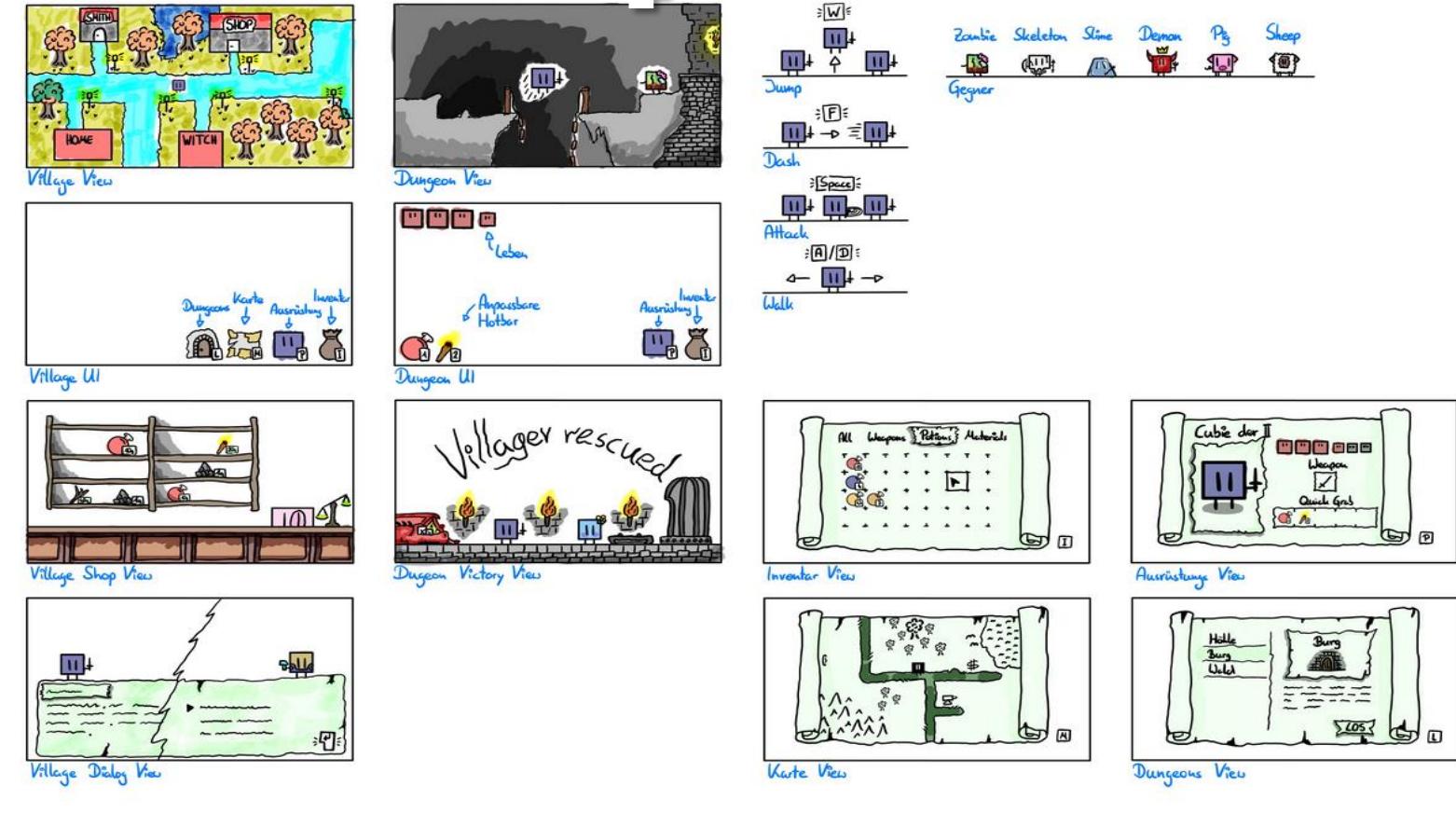
Collision Tests

```
collisionSystem.AddEntity(enemy);
collisionSystem.AddEntity(player);

player.GetComponent<Transform>().Position += new Vector2(2, 0);
collisionSystem.Update(0);

Assert.AreEqual(new Vector2(1, 0), player.GetComponent<Transform>().Position);
}
```

Mockup

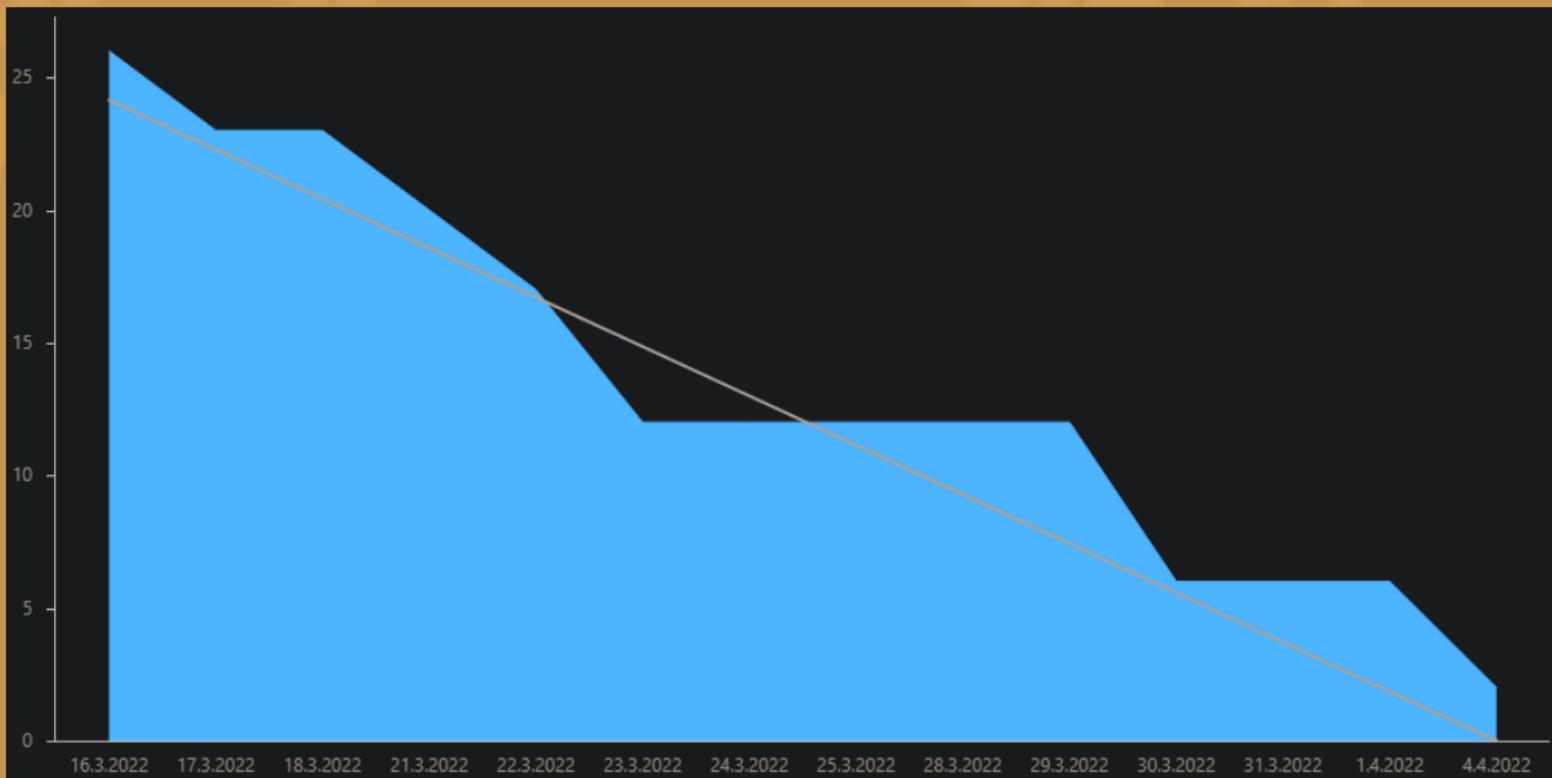


Architecture

Testing

>Retrospektive<

Sprint 1

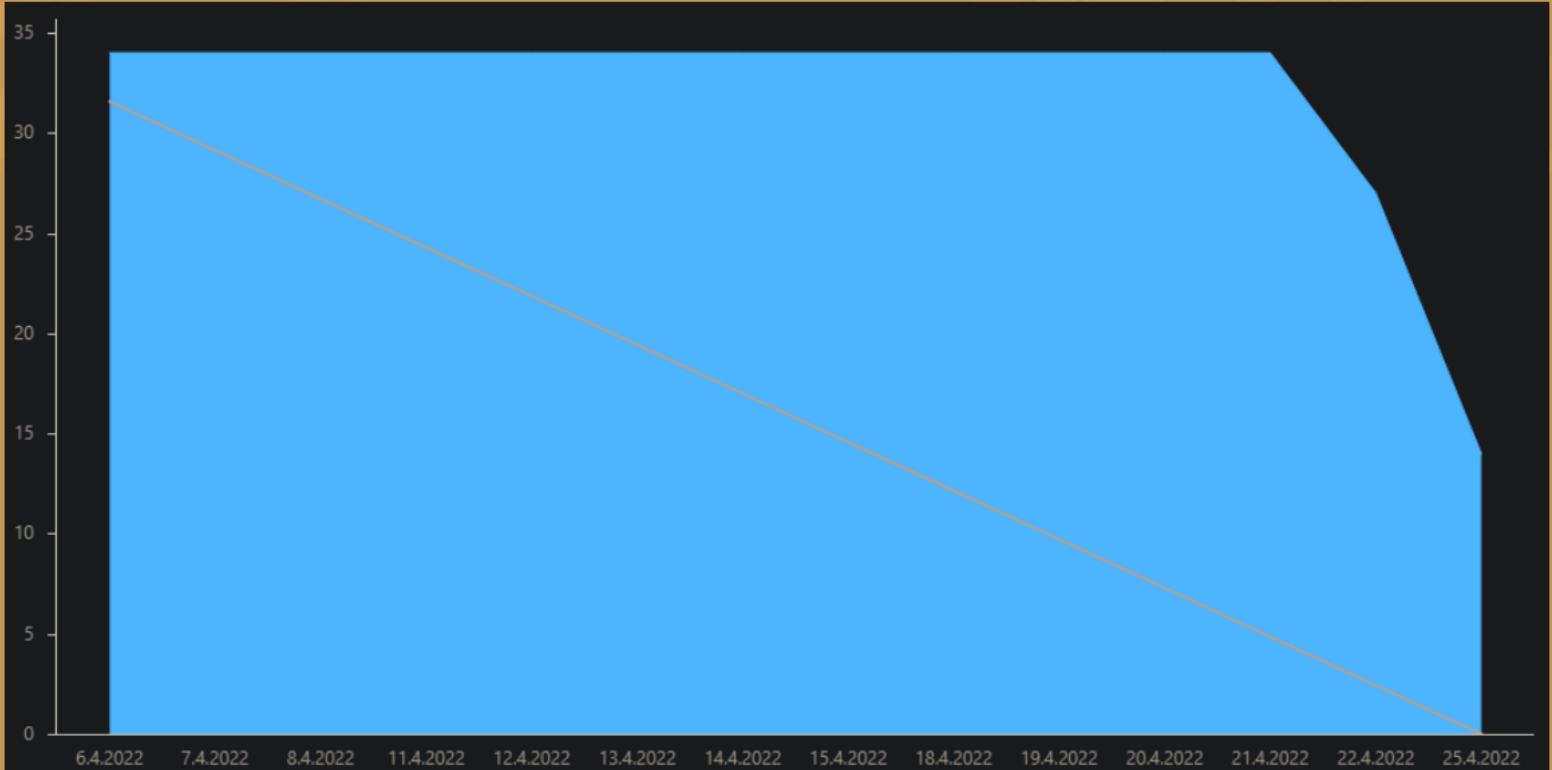


Architecture

Testing

>Retrospektive<

Sprint 2

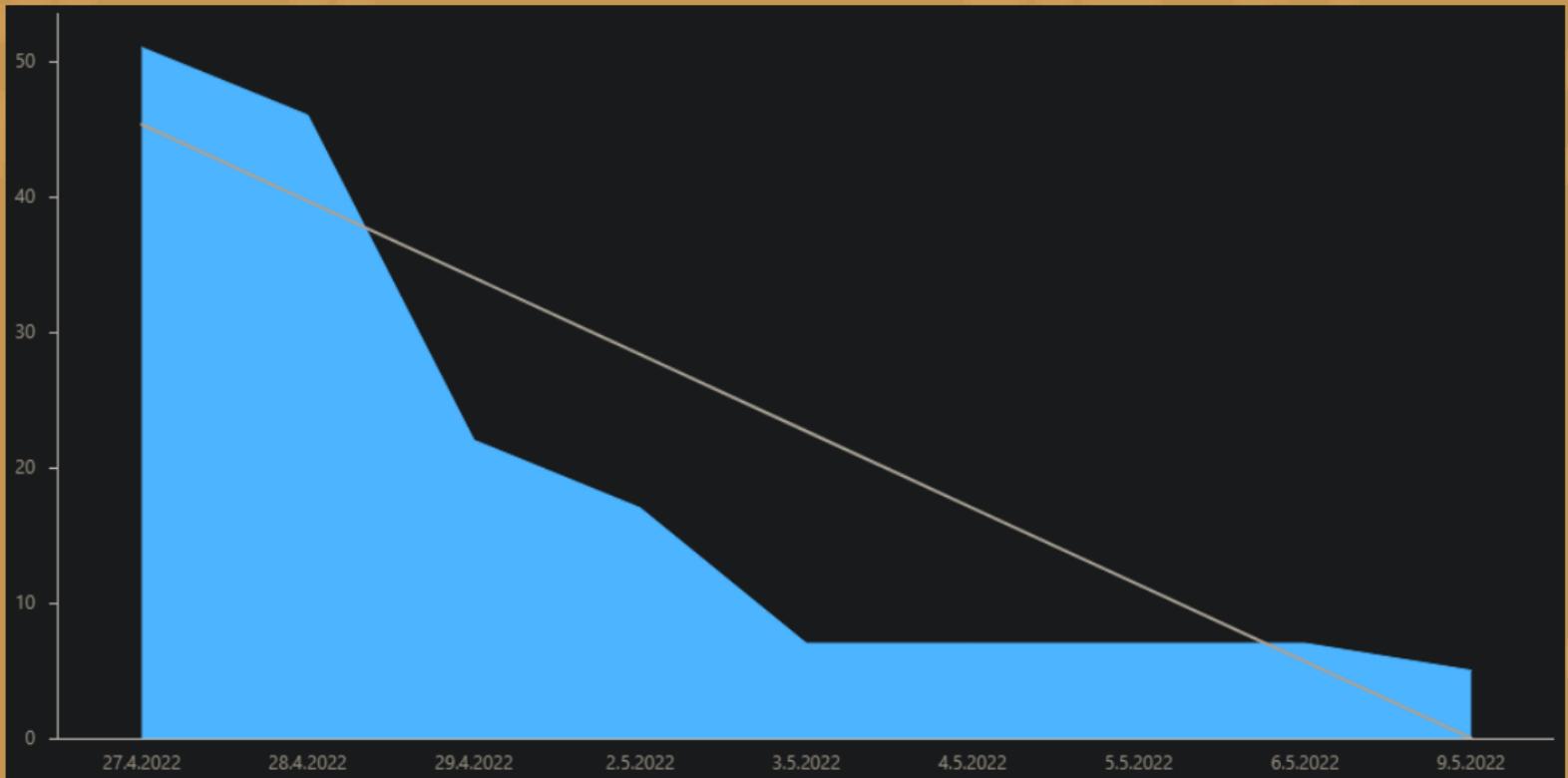


Architecture

Testing

>Retrospektive<

Sprint 3

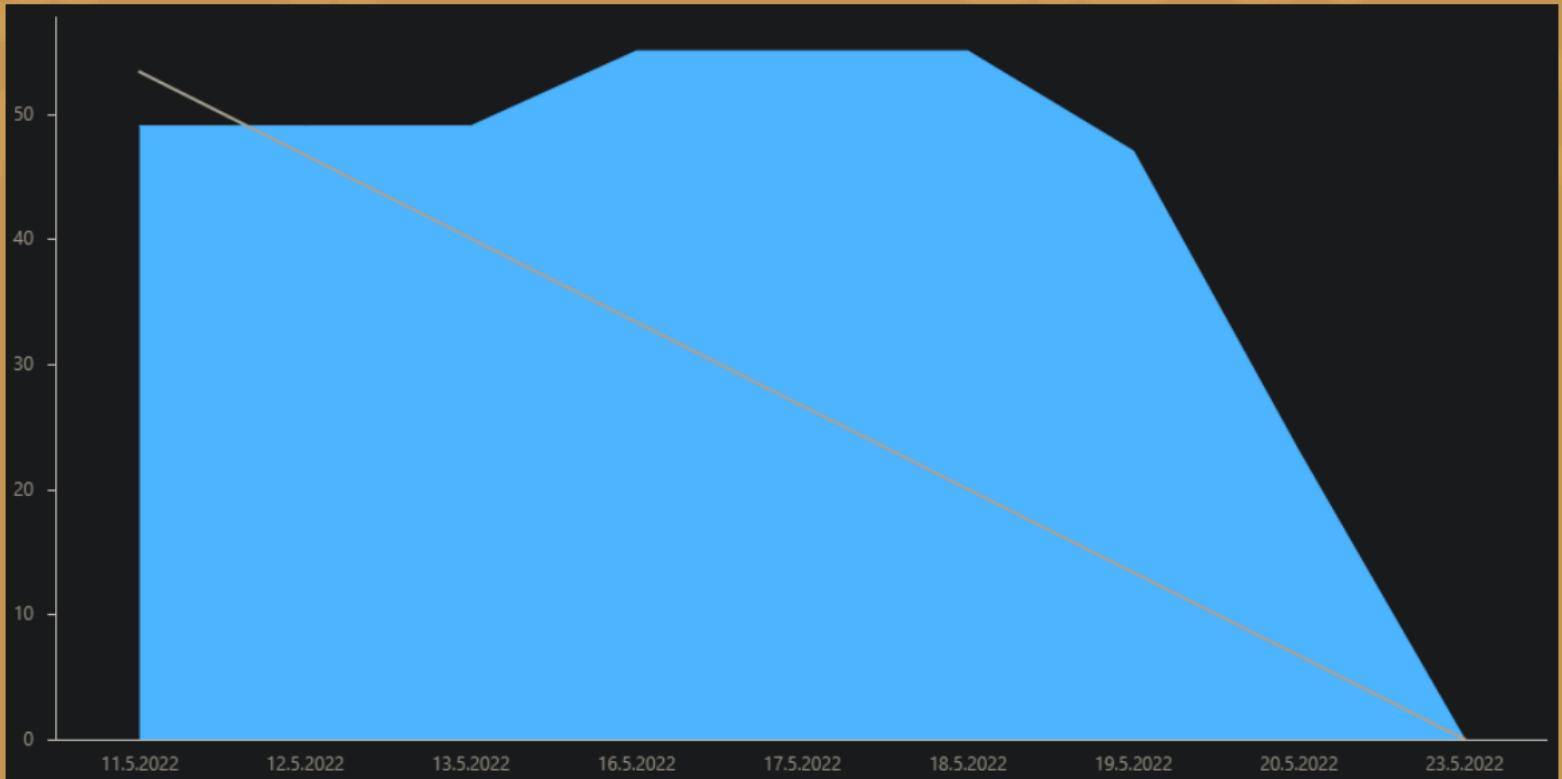


Architecture

Testing

>Retrospektive<

Sprint 4

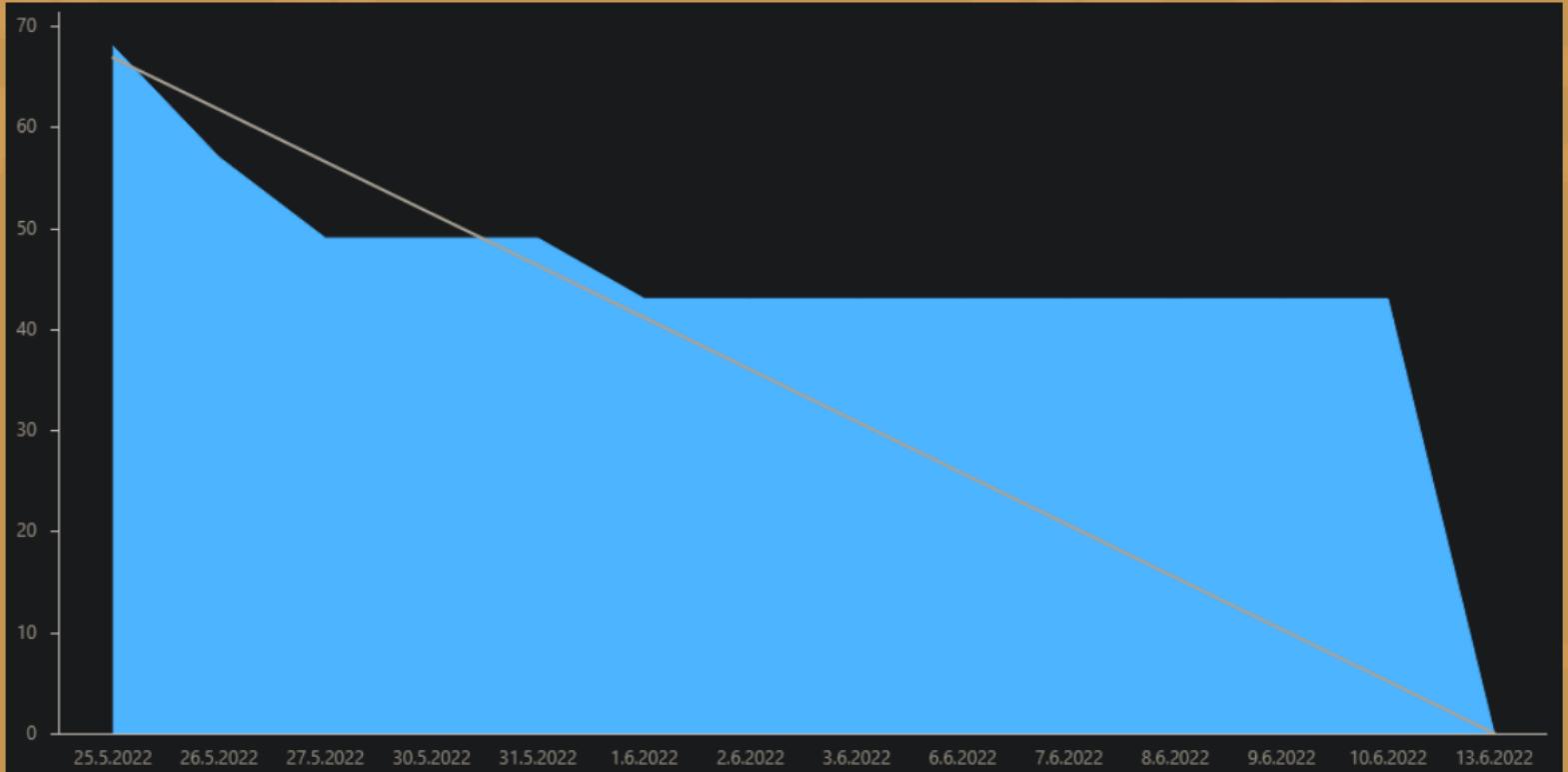


Architecture

Testing

>Retrospektive<

Sprint 5

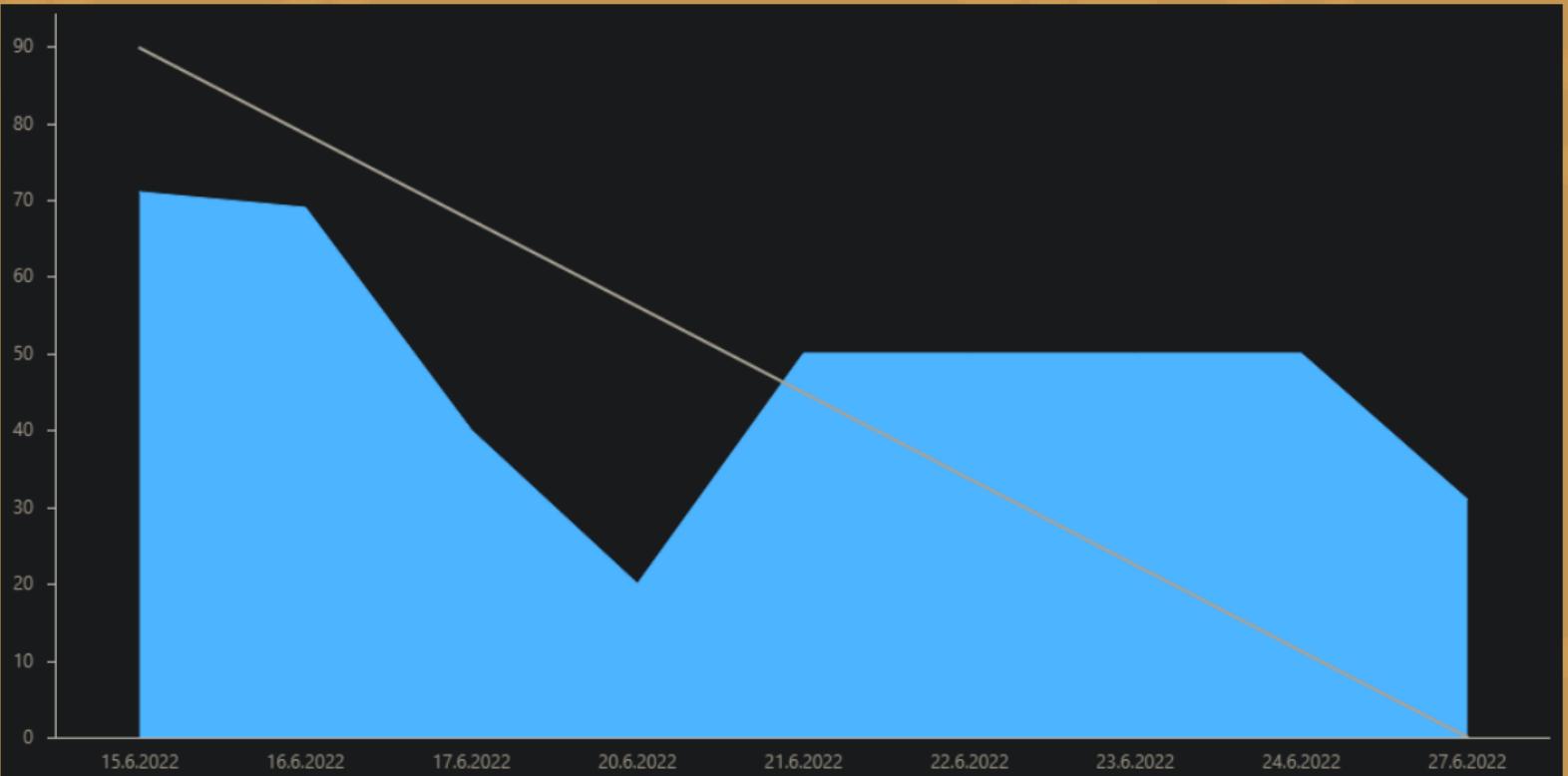


Architecture

Testing

>Retrospektive<

Sprint 6



Was hatte verbessert werden können:

- Fokus verloren
- Anfangs keine Struktur
- Wenig manuelle tests
- Wenig documentation

Was lief gut:

- Meiste geschafft
- Sinnvolle tests
- Gute programmstruktur
- Kontinuierlicher Fortschritt
- Jeder hat etliches beigetragen.

Litw2 dlett