Coursework Map Rehash & Interfaces

Software Engineering 1

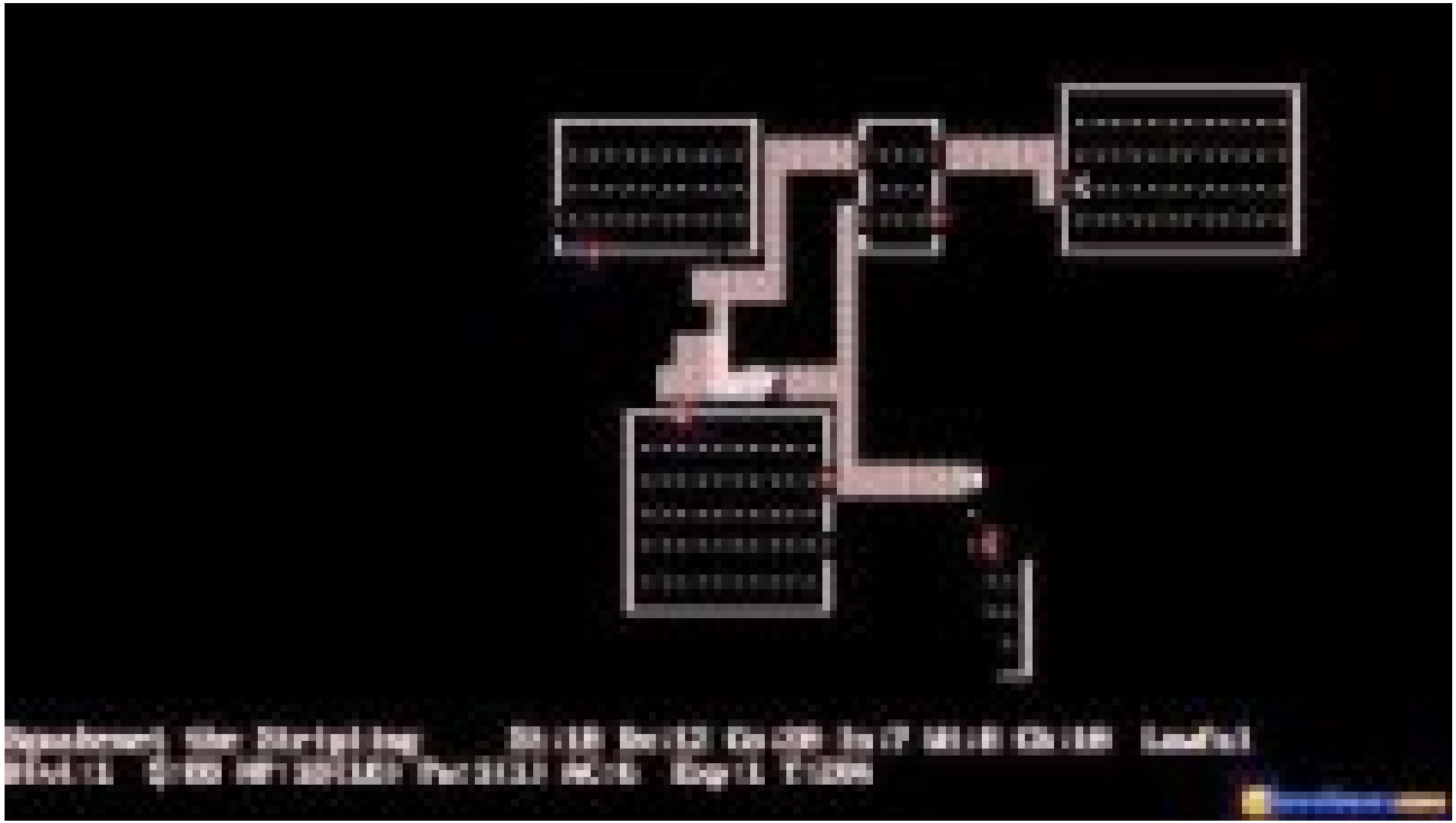
Dr Swen E. Gaudl

University of Plymouth 2022

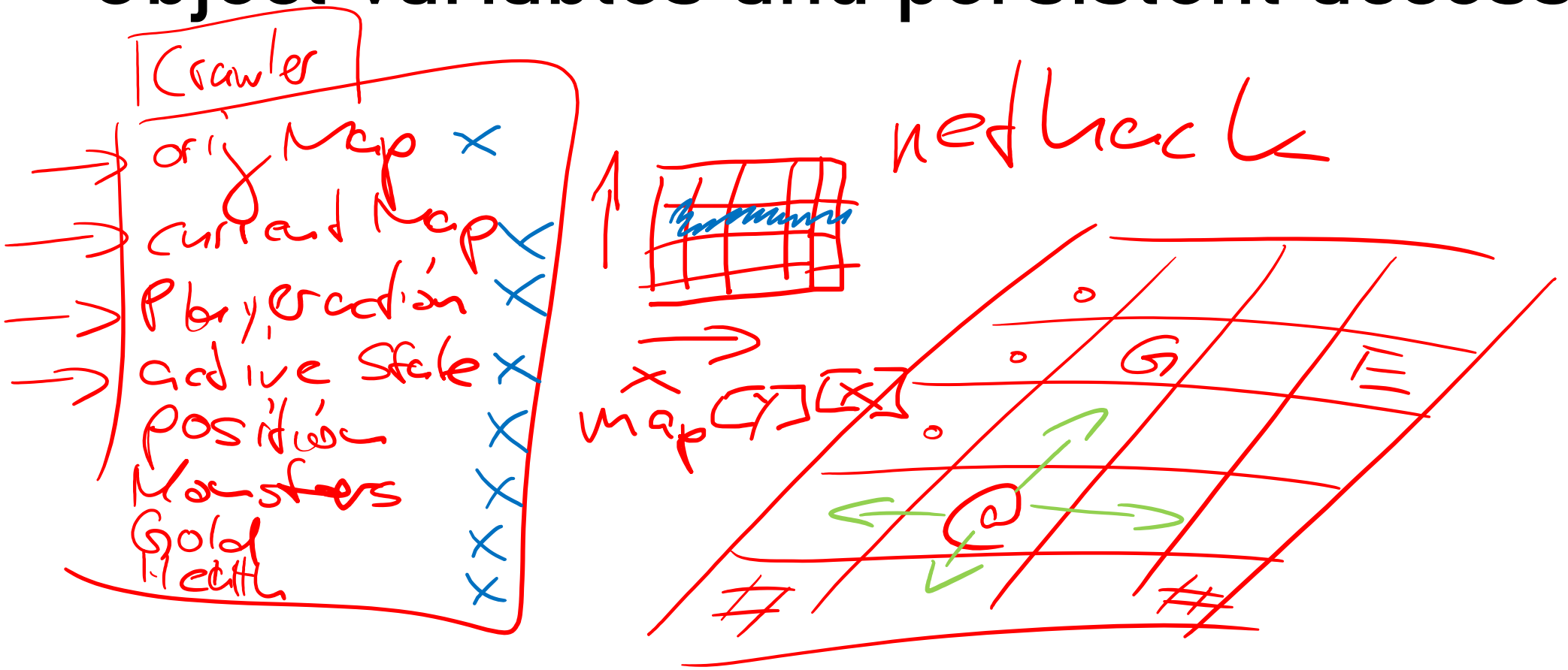
COMP10000

NetHack/ Dwarf Fortress

NetHack/ Dwarf Fortress



Object variables and persistent access



Prep For Thursday:

- Come with questions for CW & Ex5!
- No need to attend if No Questions!

- Latest Feedback possible before end of year!

COMP1000 Agenda This Week:

- Interfaces
- Q&A and EX5 Introduction

Interfaces & Abstract/ Partial Class

Features:

- Specify members
- Do not allow fields/variables
- OOD principle employed:
  - hiding information/implementation
  - Abstraction
  - Polymorphism
- Allows for Multi-Inheritance
- No Implementation of functionality
- Cannot be instantiated

```
public interface IEntity {
    // interface members
    bool doDamage(IEntity entity);
    double getHealth();
    bool isAlive();
}
```

Interfaces & Abstract/ Partial Class

```
public class Entity : IEntity
{
    private int Health;
    public bool doDamage(IEntity entity)
    {}

    public double getHealth()
    {}

    public bool isAlive()
    {}
}
```

```
public interface IEntity {
    // interface members
    bool doDamage(IEntity entity);
    double getHealth();
    bool isAlive();
}
```

Interfaces

```
interface IInterface
{
  public void Method1();
  public int Method2();
}
class ActualClass : IInterface
{
   // Explicit interface member
implementation:
   public void Method1()
   {
    // Method implementation.
   }
   public int Method2()
   {
    // Method implementation.
    return 0;
   }
```

```
static void Main()
{
   // Declare an interface instance.
   IInterface obj = new ActualClass();
   // Call the member.
   obj.Method1();
}
```

Abstract

Features:

- Specify members
- Does allow fields/variables
- OOD principle employed:
  - hiding information/implementation
  - Abstraction
  - Polymorphism
- Allows for Multi-Inheritance
- cannot be instantiated
- may contain abstract methods and accessors
- not possible to modify an abstract class with the sealed modifier

Abstract

```
public abstract class AEntity {
    // interface members
    protected int Lives = 0;
    public abstract bool doDamage(IEntity entity);
    double getLives()
     {
         return Lives;
     }

    public abstract bool isAlive();
}
```

Features:

- Specify members
- Does allow fields/variables
- OOD principle employed:
  - hiding information/implementation
  - Abstraction
  - Polymorphism
- Allows for Multi-Inheritance
- cannot be instantiated
- may contain abstract methods and accessors
- not possible to modify an abstract class with the sealed modifier

```
public interface IEntity {
    // interface members
    bool doDamage(IEntity entity);
    double getHealth();
    bool isAlive();
}
```

Abstract

```
public interface IEntity {
    // interface members
    bool doDamage(IEntity entity);
    double getHealth();
    bool isAlive();
}

public abstract class AEntity {
    // interface members
    protected int Lives = 0;
    public abstract bool doDamage(IEntity entity);
    double getLives()
     {
         return Lives;
     }

    public abstract bool isAlive();
}
```

```
public class Entity : AEntity, IEntity
{
  public override bool doDamage(IEntity entity)
  { }
  public double getHealth()
  { }
  public override bool isAlive()
  { }
}
```

Further Reading

Reading Up:

- https://docs.microsoft.com interfaces
- w3schools.com
- www.tutorialspoint.com

Further Concepts:

- abstract class
- Sealed keyword
- partial class