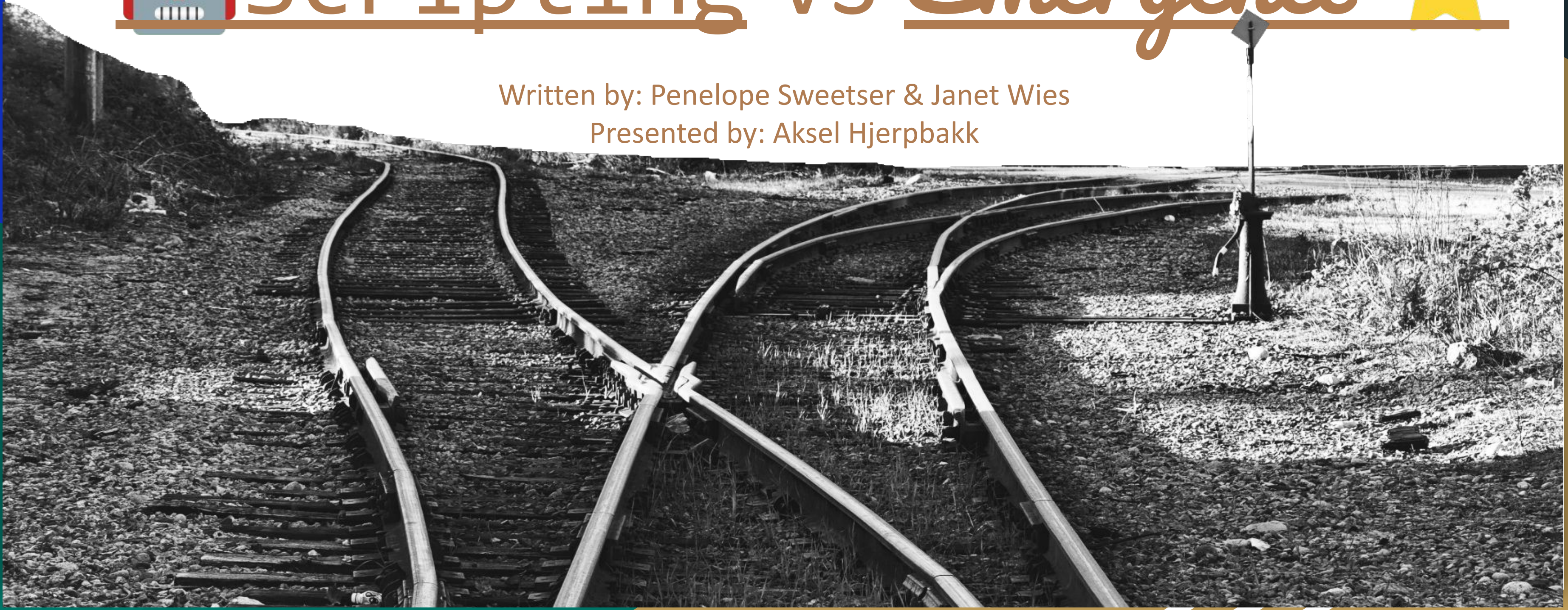
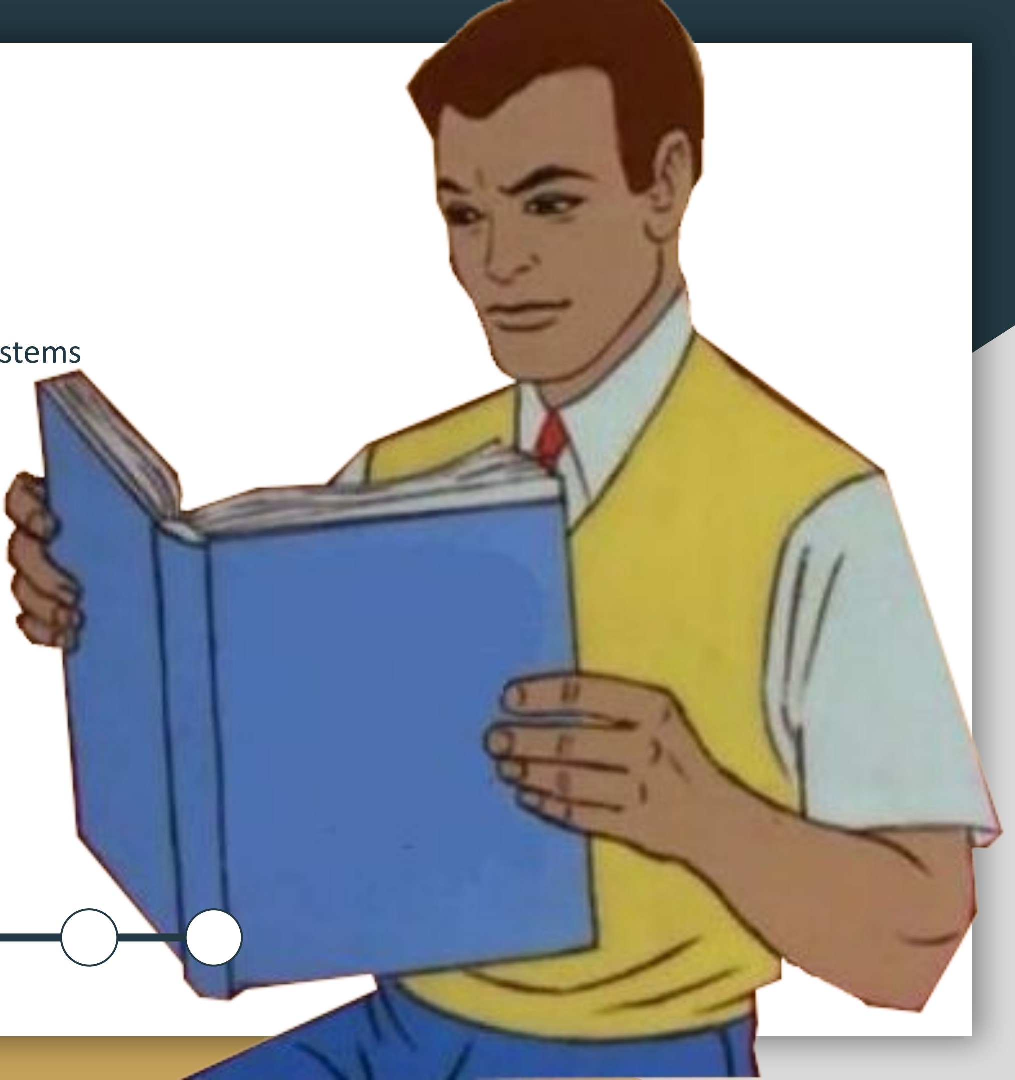
🤖Scripting VS Emergence ⭐

Written by: Penelope Sweetser & Janet Wies
Presented by: Aksel Hjerpbakk



Introduction

- Two main approaches for designing game systems
  - Scripting
  - Emergence
- Both approaches are challenging in their own way!



Scripting and Emergence

- The majority of games are scripted
  - Predefined paths
  - The game designers dictate
  - Scripted interactions are called “emulation”
- The alternative to scripted interaction is emergent interaction
  - Not hardcoded on a local scale, but instead use global rules
  - The emergent approach to game design is called “simulation”

Considerations for game developers

- Five issues to consider with game development

    1. Implementing and testing
    2. Modifying and extending
    3. Creative control for game developers
    4. Uncertainty and QA
    5. Ease of feedback and directions to player

Considerations for scripted systems

1. **Implementing and testing**
2. Modifying and extending
3. Creative control for game developers
4. Uncertainty and QA
5. Ease of feedback and directions to player

Considerations for scripted systems

1. Implementing and testing
2. **Modifying and extending** ⇦
3. Creative control for game developers
4. Uncertainty and QA
5. Ease of feedback and directions to player

●—●—●—●—○—○—○—○—○

Considerations for scripted systems

1. Implementing and testing
2. Modifying and extending
3. **Creative control for game developers** ⇦
4. Uncertainty and QA
5. Ease of feedback and directions to player

Considerations for scripted systems

1. Implementing and testing
2. Modifying and extending
3. Creative control for game developers
4. **Uncertainty and QA**
5. Ease of feedback and directions to player

Considerations for scripted systems

1. Implementing and testing
2. Modifying and extending
3. Creative control for game developers
4. Uncertainty and QA
5. **Ease of feedback and directions to player**

Considerations for emergent systems

1. **Implementing and testing** ⇦
2. Modifying and extending
3. Creative control for game developers
4. Uncertainty and QA
5. Ease of feedback and directions to player

Considerations for emergent systems

1. Implementing and testing
2. **Modifying and extending** ⇦
3. Creative control for game developers
4. Uncertainty and QA
5. Ease of feedback and directions to player

● ● ● ● ● ○ ○ ○ ○

Considerations for emergent systems

1. Implementing and testing
2. Modifying and extending
3. **Creative control for game developers**
4. Uncertainty and QA
5. Ease of feedback and directions to player

Considerations for emergent systems

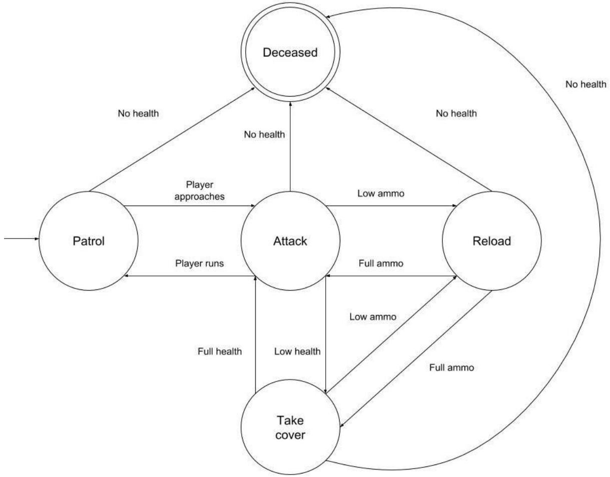
1. Implementing and testing
2. Modifying and extending
3. Creative control for game developers
4. **Uncertainty and QA**
5. Ease of feedback and directions to player

Considerations for emergent systems

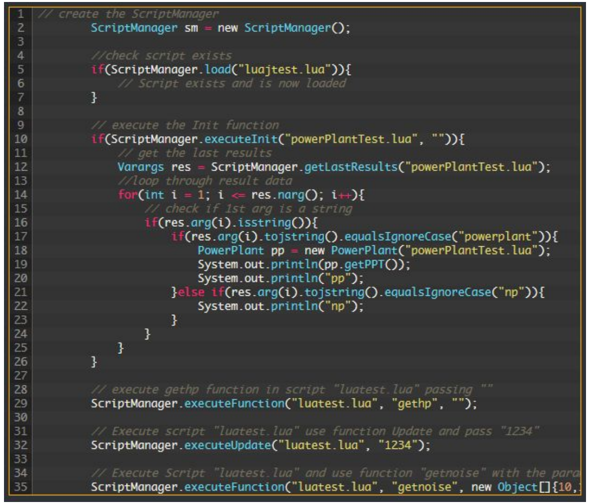
1. Implementing and testing
2. Modifying and extending
3. Creative control for game developers
4. Uncertainty and QA
5. **Ease of feedback and directions to player**

Techniques for scripting game worlds

Finite State Machine

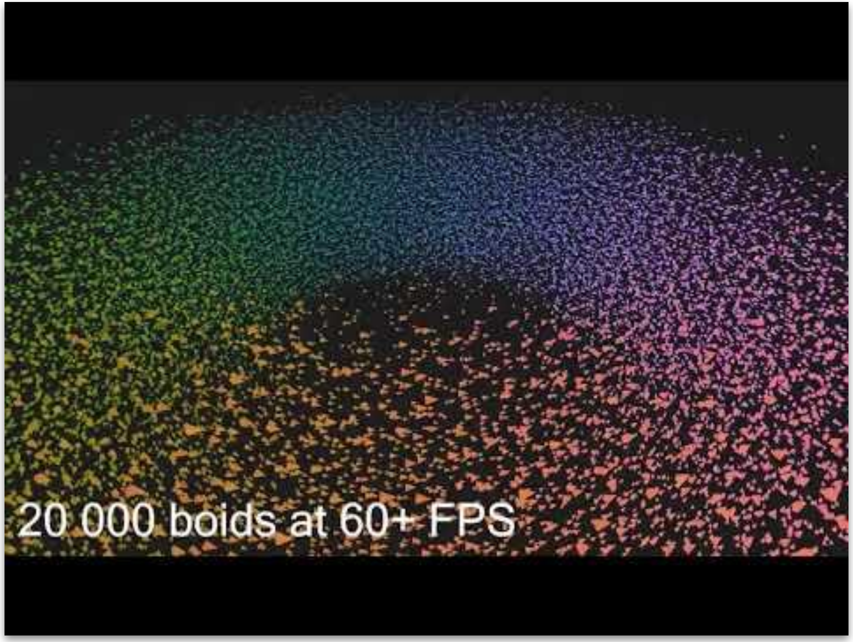


Scripting Languages



```
// create the ScriptManager
        ScriptManager sm = new ScriptManager();

        //check script exists
        if(ScriptManager.load("luajtest.lua")){
            // Script exists and is now loaded
        }

        // execute the Init function
        if(ScriptManager.executeInit("powerPlantTest.lua", "")){
            // get the last results
            Varargs res = ScriptManager.getLastResults("powerPlantTest.lua");
            //loop through result data
            for(int i = 1; i <= res.narg(); i++){
                // check if 1st arg is a string
                if(res.arg(i).isstring()){
                    if(res.arg(i).tojstring().equalsIgnoreCase("powerplant")){
                        PowerPlant pp = new PowerPlant("powerPlantTest.lua");
                        System.out.println(pp.getPPT());
                        System.out.println("pp");
                    }else if(res.arg(i).tojstring().equalsIgnoreCase("np")){
                        System.out.println("np");
                    }
                }
            }
        }

        // execute gethp function in script "luatest.lua" passing ""
        ScriptManager.executeFunction("luatest.lua", "gethp", "");

        // Execute script "luatest.lua" use function Update and pass "1234"
        ScriptManager.executeUpdate("luatest.lua", "1234");

        // Execute Script "luatest.lua" and use function "getnoise" with the para
        ScriptManager.executeFunction("luatest.lua", "getnoise", new Object[]{10,
```

Techniques for emergence in game worlds



- Flocking (boids)
- Cellular Automata
- Neural Networks
- Evolutionary Algorithms

Emergent gameplay and player expression

Enables player creativity

For more insight into designing an emergent game, you can look at this documentary about Prey (I will link it in the comments)