MODEL-BASED TESTING OF 3D VIDEO GAMES
Madis Taimre, Elvior
Õnne Mets, Virtual Heritage
Overview

- Issues with game testing
- Available methods for automated testing
- Testing of VR (Virtual Reality) Quiz used for museums
- Created testing framework
- Test results
ISSUES WITH GAME TESTING
Alpha phase horrors

- The game is feature complete, but untested
- Systematic testing begins
- QA works overtime or hires additional staff
- Automated testing should alleviate this phase
- Tools should be usable when game is being developed
Pursuit of abstraction

- Video game
  - Built in randomness
  - Players avatar
  - Other players
  - Environment events
  - Music
  - Animations
  - Intractable objects
  - Environment objects
  - Sound effects
  - Artificial intelligence

User Conference on
Advanced Automated Testing
AVAILABLE METHODS FOR AUTOMATED TESTING OF GAMES
Available methods for automated testing of games

• Capture and replay
  • No abstraction

• Bots
  • No test oracle
  • Limited use

• Model-based testing
  • High coverage and abstraction
  • Maintainable
Testing framework

- Framework to control the behavior of a game
- Get the state of the game
- Usable with Unity®
- Can be used with methods other than MBT
- Open source
Test runner

- Test scripts
- Test data

Framework .NET library

- Connection handler
- View State

Game development platform

- Manager
- Object 1
- Object 2
- Object ..
- Object n
TestObject

- IsVisible
  - Active in hierarchy
  - Seeable by avatar

- GetParameter
  - Text
  - Indicator status

- Trigger
  - Button
  - Environment object
VR Quiz

- Virtual Reality
- Point and click interface
- 6 DOF for player
Testing with scripts

- Turn to object and check visibility
- Turn to object and click
- Checks the states of all object every time
  - Higher coverage

```java
[TestMethod]
public void TestChooseImage1()
{
    CheckViewState(m_adapter.GetViewState(), Level1Data.ViewStates);
    m_adapter.TriggerButton(Level1Data.m_estonianPlaque);
    CheckViewState(m_adapter.GetViewState(), Level2Data.ViewStatesEstonianQuestion);
    m_adapter.TriggerButton(Level2Data.m_quizObject1);
    CheckViewState(m_adapter.GetViewState(), Level2Data.JailPosViewState);
}
```
Test results

- 64% coverage
- Generated over 570 actions
- Found 4 major errors
Conclusion

• MBT offers high abstraction, which is needed for game testing
• By adding test scripts into a game, it is possible to test various game elements and interactions
• Successfully tested a 3D game using MBT and the created framework
  • https://gitlab.com/madis-taimre/TestingFramework.git
THANK YOU FOR LISTENING