

In the development of UniSim, our testing strategy was meticulously crafted to ensure the application's reliability, functionality, and overall user experience. Drawing inspiration from the foundational principles outlined in "Introduction to Software Testing" by Offutt et al. and adhering to the ISO/IEC 25010:2011 standards, we established a comprehensive approach that effectively balances thoroughness with efficiency.

Central to our methodology was an exhaustive manual testing regimen, wherein every single project requirement was meticulously validated through direct interaction with the game. This black-box testing approach allowed our team to evaluate UniSim's functionalities from an end-user perspective, ensuring that each feature performed as intended without delving into the underlying code structure. By designing test cases derived explicitly from the project specifications, we guaranteed that all intended functionalities were rigorously examined, thereby affirming the application's functional completeness and correctness.

To complement our manual efforts, we integrated automated testing to enhance the efficiency and consistency of our quality assurance processes. Automated tests were strategically developed to target critical components and repetitive tasks, facilitating rapid regression testing and ensuring that core functionalities remained stable throughout the development lifecycle. This white-box testing approach enabled us to assess the internal workings of UniSim, utilizing coverage tools to monitor statement and branch coverage. By identifying untested code segments, we were able to iteratively refine our test suite, addressing potential vulnerabilities and optimizing the application's performance in alignment with the pesticide paradox concept discussed in Offutt et al.'s work.

Recognizing the inherent complexity and variability of interactive systems like UniSim, our layered testing strategy also incorporated manual exploratory testing. This qualitative approach empowered our testers to intuitively navigate the game, uncovering subtle issues related to user interface responsiveness, gameplay mechanics, and emergent behaviors that structured test cases might overlook. Such exploratory efforts were instrumental in enhancing the application's usability and ensuring a seamless and engaging user experience, thereby validating not only the functional aspects but also the user-centric attributes of UniSim.

Furthermore, our testing framework was designed to uphold the principles of verification and validation. Verification activities focused on ensuring that the game was built in accordance with the specified requirements and design documents, thereby confirming the technical accuracy and reliability of the implemented features. Concurrently, validation efforts were directed toward assessing whether UniSim met user expectations and provided an enjoyable gaming experience, thereby bridging the gap between technical specifications and user satisfaction.

In alignment with the ISO/IEC 25010:2011 standards, our approach also emphasized non-functional requirements such as performance, usability, and accessibility. By incorporating both manual and automated testing techniques, we were able to evaluate these aspects comprehensively, ensuring that UniSim not only functioned correctly but also delivered a high-quality and inclusive user experience.

B)

## **Manual Testing Brief:**

A comprehensive array of manual tests were done to validate the functional and non-functional requirements of the game. A total of 30 manual tests were executed. Each requirement was tested to ensure consistency with the product brief.

### **Testing Statistics**

- Total Tests Run: 30
- Tests Passed: 30
- Tests Failed: 0

### **Test Results**

All conducted tests successfully met their expected outcomes. The game performed consistently across different operating systems (Windows, Mac, Linux) and hardware configurations, maintaining stable frame rates and seamless UI scaling. Audio functionalities, including background music and sound effects, operated without glitches, and the mute feature functioned as intended.

Feature-specific tests, such as building placement, movement, deletion, and upgrades, were executed flawlessly, ensuring that players could interact with the game environment without encountering errors. The introduction of achievements and leaderboards was validated, confirming that these features accurately tracked and displayed player progress and rankings.

Additionally, accessibility features for colorblind users were effectively implemented, enhancing the game's inclusivity. Student satisfaction metrics responded correctly to player actions and in-game events, providing real-time feedback that aligned with user interactions.

### **Completeness and Correctness of Tests**

The testing process was thorough, covering all functional requirements and user requirements as specified. Each requirement was linked to its corresponding functional or non-functional counterpart, ensuring complete traceability and comprehensive coverage. The manual testing approach, which involved actively playing the game and interacting with all features, provided reliable insights into the game's performance and user experience.

The absence of failed tests indicates that the current implementation is robust and adheres closely to the defined requirements. The tests were designed to be both exhaustive and precise, verifying not only the presence of features but also their correct behavior under various scenarios.

## **Automated Testing Brief**

A comprehensive array of automated tests were done to validate the functional and non-functional requirements of the game. A total of 22 automated tests were executed. Each requirement was tested to ensure consistency with the product brief.

### **Testing Statistics**

A total of 22 automated tests were executed to validate various aspects of UniSim. These tests encompassed critical functionalities, including building placement mechanics, audio functionalities, achievement tracking, and leaderboard accuracy. The testing process yielded the following statistics:

- Total Automated Tests Run: 22
- Tests Passed: 22
- Tests Failed: 0

Code Coverage Summary:

- Classes Covered: 11.3% (7 out of 62 classes)
- Methods Covered: 16.6% (46 out of 277 methods)
- Lines Covered: 9.1% (114 out of 1253 lines)

### **Completeness and Correctness of Tests**

The automated testing suite successfully passed all executed tests, indicating that the functionalities they were designed to validate operate correctly within their tested scopes. This impeccable pass rate suggests a high level of correctness in our test implementations, ensuring that the covered features function as intended without introducing defects.

However, the code coverage analysis reveals that the overall coverage is relatively low, with only 11.3% of classes, 16.6% of methods, and 9.1% of lines being tested. While specific classes such as Building, BuildingInstance, BuildingPlacementManager, BuildingType, Coord, and InGameTimer achieved full coverage, a significant number of critical classes remain entirely untested. These include BuildingManager, BuildingRenderer, Events, LeaderBoard, and several others essential to core game functionalities.

The partial coverage in the Achievements class, at 50%, indicates that only half of its methods and lines have been validated, suggesting room for improvement to ensure complete reliability. The absence of coverage in classes responsible for managing buildings, rendering graphics, handling events, and maintaining leaderboards is particularly concerning, as these areas are pivotal to the game's operational integrity and user engagement.

Despite the absence of failed tests, the low overall code coverage points to a lack of completeness in our testing approach. This gap implies that numerous functionalities remain unvalidated through automated means, increasing the risk of undetected bugs and performance issues that could adversely affect the game's stability and user experience.

C)

## **Manual Tests:**

Testing report format

**Requirement:** UR\_OS\_COMPATIBILITY

**Author:** Daron Lepejian

**Related Requirement:** N/A

**Testing method:** Manual

**Expected Outcome:** We expected the game to run on Windows, Mac, and Linux operating systems without running into problems

**Actual Outcome:** Our game ran well on Windows, Mac, and Linux operating systems and did not encounter any problems

**Status:** Pass

---

**Requirement:** UR\_HARDWARE\_COMPATABILITY

**Author:** Daron Lepejian

**Related Requirement:** N/A

**Testing method:** Manual

**Expected Outcome:** We anticipate that the game will operate smoothly on systems with minimal hardware specifications

**Actual Outcome:** We tested the game on different entry-level laptops and low-performance computers and did not experience a drop in frame rate or

**Status:** Pass

---

**Requirement:** UR\_DISPLAY\_SCALABILITY

**Author:** Daron Lepejian

**Related Requirement:** N/A

**Testing method:** Manual

**Expected Outcome:** The game should display correctly on different screen resolutions and scales without UI issues.

**Actual Outcome:** The game UI adapted seamlessly to multiple screen resolutions and scaling settings without any display problems.

**Status:** Pass

---

**Requirement:** UR\_AUDIO

**Author:** Daron Lepejian

**Related Requirement:** FR\_AUDIO\_MUTE, FR\_SOUNDTRACK

**Testing method:** Manual

**Expected Outcome:** Audio should play correctly, with no glitches or delays across all game scenarios.

**Actual Outcome:** All game sounds and music played smoothly without any audio issues during testing.

**Status:** Pass

---

**Requirement:** UR\_COUNTER

**Author:** Daron Lepejian

**Related Requirement:** FR\_COUNT\_DISPLAY

**Testing method:** Manual

**Expected Outcome:** In-game counters should accurately reflect the current game state in real-time.

**Actual Outcome:** All counters updated correctly in response to game events without discrepancies.

**Status:** Pass

---

**Requirement:** UR\_LOCATIONS

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_PLACEABILITY, FR\_LOCATIONS\_SIZE, FR\_MAP

**Testing method:** Manual

**Expected Outcome:** All game locations should load correctly and be accessible to the player.

**Actual Outcome:** Each location loaded without issues and was fully accessible during manual playtesting.

**Status:** Pass

---

**Requirement:** UR\_LOCATIONS\_MOVE

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_PLACEABILITY, FR\_LOCATION\_PLACEABILITY, FR\_LEADERBOARD

**Testing method:** Manual

**Expected Outcome:** Players should be able to move between locations seamlessly.

**Actual Outcome:** Movement between all locations was smooth with no glitches or delays.

**Status:** Pass

---

**Requirement:** UR\_LOCATIONS\_SIZES

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_SIZE

**Testing method:** Manual

**Expected Outcome:** Locations should maintain consistent sizes and proportions across different devices.

**Actual Outcome:** All locations retained their intended sizes and proportions during testing on various devices.

**Status:** Pass

---

**Requirement:** UR\_MAP

**Author:** Daron Lepejian

**Related Requirement:** FR\_MAP

**Testing method:** Manual

**Expected Outcome:** The in-game map should accurately represent all locations and be interactive.

**Actual Outcome:** The map displayed all locations correctly and responded to user interactions as expected.

**Status:** Pass

---

**Requirement:** UR\_LOCATIONS\_UPGRADE

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_UPGRADE

**Testing method:** Manual

**Expected Outcome:** Players should be able to upgrade locations without errors.

**Actual Outcome:** Upgrading locations functioned correctly without any issues during manual testing.

**Status:** Pass

---

**Requirement:** UR\_MENU

**Author:** Daron Lepejian

**Related Requirement:** FR\_SOUNDTRACK

**Testing method:** Manual

**Expected Outcome:** Game menus should be responsive and navigate correctly.

**Actual Outcome:** All menus operated smoothly and navigated to the correct sections without problems.

**Status:** Pass

---

**Requirement:** UR\_LOCATION\_PLACEABILITY

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_PLACEABILITY, FR\_LOCATION\_PLACEABILITY, FR\_MAP

**Testing method:** Manual

**Expected Outcome:** Players should be able to place buildings or objects in designated locations without issues.

**Actual Outcome:** The placement of buildings and objects worked flawlessly during manual testing.

**Status:** Pass.

---

**Requirement:** UR\_EVENTS

**Author:** Daron Lepejian

**Related Requirement:** FR\_EVENTS\_MANAGEMENT

**Testing method:** Manual

**Expected Outcome:** In-game events should trigger correctly and affect the game state as intended.

**Actual Outcome:** All events triggered appropriately and impacted the game state as expected during testing.

**Status:** Pass

---

**Requirement:** UR\_STUDENT\_SATISFACTION

**Author:** Daron Lepejian

**Related Requirement:** FR\_SATISFACTION\_UPDATE

**Testing method:** Manual

**Expected Outcome:** Student satisfaction metrics should accurately reflect player actions and choices during events

**Actual Outcome:** Satisfaction scores updated correctly based on player interactions during manual testing.

**Status:** Pass

---

**Requirement:** UR\_RECREATION\_X2

**Author:** Daron Lepejian

**Related Requirement:** FR\_RECREATION\_MULTIPLICITY

**Testing method:** Manual

**Expected Outcome:** Recreation features should function correctly and enhance player experience.

**Actual Outcome:** Recreation features operated smoothly and positively impacted the gameplay experience during testing.

**Status:** Pass

---

**Requirement:** UR\_ACHIEVEMENTS

**Author:** Daron Lepejian

**Related Requirement:** FR\_ACHIEVEMENTS

**Testing method:** Manual

**Expected Outcome:** Achievements should be displayed correctly and track player progress accurately.

**Actual Outcome:** Achievements were displayed and updated correctly as players met the criteria during manual testing.

**Status:** Pass

---

**Requirement:** UR\_LEADERBOARD

**Author:** Daron Lepejian

**Related Requirement:** FR\_LEADERBOARD

**Testing method:** Manual

**Expected Outcome:** The leaderboard should display player rankings accurately based on scores or other metrics.

**Actual Outcome:** The leaderboard displayed correctly and updated rankings in real-time during manual testing.

**Status:** Pass

---

**Requirement:** FR\_LOCATIONS\_PLACEABILITY

**Author:** Daron Lepejian

**Related Requirement:** UR\_LOCATIONS, UR\_LOCATION\_PLACEABILITY

**Testing method:** Manual

**Expected Outcome:** The system shall clearly display on the grid whether a building can be placed.

**Actual Outcome:** Placement indicators accurately showed valid and invalid placement areas during testing.

**Status:** Pass

---

**Requirement:** FR\_LOCATION\_PLACEABILITY

**Author:** Daron Lepejian

**Related Requirement:** UR\_LOCATION\_PLACEABILITY, UR\_LOCATIONS\_PLACEABILITY, UR\_MAP

**Testing method:** Manual

**Expected Outcome:** The system shall show to the user whether they can place a building in a location on a grid or not.

**Actual Outcome:** Placement availability indicators worked correctly, preventing buildings from being placed on restricted areas.

**Status:** Pass

---

**Requirement:** FR\_LOCATIONS\_SIZE

**Author:** Daron Lepejian

**Related Requirement:** UR\_LOCATIONS\_SIZES

**Testing method:** Manual

**Expected Outcome:** Different locations should take up different amounts of squares on the grid.

**Actual Outcome:** Locations occupied the correct number of grid squares as designed during testing.

**Status:** Pass

---

**Requirement:** FR\_EVENTS\_MANAGEMENT

**Author:** Daron Lepejian

**Related Requirement:** UR\_EVENTS

**Testing method:** Manual

**Expected Outcome:** The system shall schedule and trigger random events at predetermined times.

**Actual Outcome:** Multiple events were scheduled and triggered correctly at the specified intervals during manual testing.

**Status:** Pass

---

**Requirement:** FR\_SATISFACTION\_UPDATE

**Author:** Daron Lepejian

**Related Requirement:** UR\_STUDENT\_SATISFACTION

**Testing method:** Manual

**Expected Outcome:** The system shall update and recalculate student satisfaction whenever a building is placed, moved, or an event occurs.

**Actual Outcome:** Student satisfaction metrics updated accurately in response to building placements, movements, and events during manual testing.

**Status:** Pass

---

**Requirement:** FR\_RECREATION\_MULTIPLICITY

**Author:** Daron Lepejian

**Related Requirement:** UR\_RECREATION\_X2

**Testing method:** Manual

**Expected Outcome:** The system shall allow placing at least two types of recreational buildings that are visually different.

**Actual Outcome:** Multiple types of recreational buildings were placed successfully, each with distinct visuals during manual testing.

**Status:** Pass

---

**Requirement:** FR\_ACHIEVEMENTS

**Author:** Daron Lepejian

**Related Requirement:** UR\_ACHIEVEMENTS

**Testing method:** Manual

**Expected Outcome:** The game should track and display player achievements accurately.

**Actual Outcome:** Achievements were tracked and displayed correctly as players met the required criteria during manual testing.

**Status:** Pass

---

**Requirement:** FR\_LEADERBOARD

**Author:** Daron Lepejian

**Related Requirement:** UR\_LEADERBOARD

**Testing method:** Manual

**Expected Outcome:** The leaderboard should accurately reflect player rankings based on their scores or other defined metrics.

**Actual Outcome:** Leaderboard accurately displayed player rankings and updated them in real-time during manual testing sessions.

**Status:** Pass

---

**Requirement:** UR\_LOCATIONS\_DELETE

**Author:** Daron Lepejian

**Related Requirement:** FR\_LOCATIONS\_DELETE

**Testing method:** Manual

**Expected Outcome:** Players should be able to delete buildings or locations without causing errors or affecting game stability.

**Actual Outcome:** Buildings and locations were successfully deleted without any errors or negative impact on game stability during manual testing.

**Status:** Pass

---

**Requirement:** FR\_LOCATIONS\_DELETE

**Author:** Daron Lepejian

**Related Requirement:** UR\_LOCATIONS\_DELETE

**Testing method:** Manual



**Expected Outcome:** The system shall allow players to delete buildings or locations, updating all relevant game metrics and ensuring no residual data remains.

**Actual Outcome:** The deletion functionality operated correctly, removing buildings and updating counters and satisfaction metrics appropriately during manual testing.

**Status:** Pass

**Automated Tests**

Coverage Summary for Package: com.backlogged.university

Package	Class, %	Method, %	Line, %
com.backlogged.university	11.3% (7/62)	16.6% (46/277)	9.1% (114/1253)

Class	Class, %	Method, %	Line, %
Achievements	50% (1/2)	35.7% (5/14)	19% (8/42)
Building	100% (1/1)	64.3% (9/14)	66.7% (14/21)
BuildingInstance	100% (1/1)	42.9% (3/7)	45.5% (5/11)
BuildingManager	0% (0/3)	0% (0/19)	0% (0/157)
BuildingPlacementManager	100% (1/1)	66.7% (6/9)	50% (18/36)
BuildingRenderer	0% (0/1)	0% (0/7)	0% (0/49)
BuildingState	0% (0/1)	0% (0/2)	0% (0/5)
BuildingType	100% (1/1)	100% (2/2)	100% (5/5)
Coord	100% (1/1)	71.4% (5/7)	60% (9/15)
Events	0% (0/1)	0% (0/5)	0% (0/15)
GameOverScreen	0% (0/3)	0% (0/12)	0% (0/42)
GamePreferences	0% (0/1)	0% (0/28)	0% (0/42)
InGameTimer	100% (1/1)	84.2% (16/19)	85.9% (55/64)
LeaderBoard	0% (0/1)	0% (0/2)	0% (0/30)
Main	0% (0/1)	0% (0/4)	0% (0/8)
MapScreen	0% (0/12)	0% (0/33)	0% (0/228)
Pair	0% (0/1)	0% (0/8)	0% (0/14)
SatisfactionScore	0% (0/2)	0% (0/13)	0% (0/82)
SettingsScreen	0% (0/22)	0% (0/53)	0% (0/279)
Soundtrack	0% (0/1)	0% (0/3)	0% (0/7)
TitleScreen	0% (0/4)	0% (0/16)	0% (0/101)