Civilization Wars: Advanced

Final Test Report

B4

Alex Bor

Mattie432

Luke Richardson

Dhruvil Tank

Robert Zlatarski

# 

[Introduction](#h.6cpeedn6goth)

[Test Plan](#h.9vsluiwsxh1a)

[High-Level Functionality Test Cases](#h.vipv42b6usom)

[Usability Testing](#h.d0ku3751tpa)

[Unit Testing](#h.rp47zqw171mj)

[Appendix](#h.w62yi62brffq)

# 

# Introduction

This is the documentation of the final test report for the strategy game, Civilization Wars: Advanced. This test report is split into four major sections. This report includes a test plan, which outlines how the testing was organised and how the testing was carried out, a table of test cases which test for high-level functionality of Civilization Wars: Advanced, a usability test report which outlines usability testing that was carried out during the development of Civilization Wars: Advanced and what we learnt from it, and a table of unit tests that were generated to test the various methods of modules which made up Civilization Wars: Advanced.

# Test Plan

Testing was carried out during the implementation of a functionality of the game, at the end of the implementation, and at the end of the implementation of the full game, both testing for high-level functionality and low-level functionality. Testing was carried out during the implementation of a functionality and if a bug was found, then this was fixed during the implementation and therefore, this was not documented. However, testing was documented at the end of the implementation of the functionality as a functionality that was meant to be complete, may not be as intended and contain bugs which will need to be fixed when possible. See Appendix.

As well as testing during the development of Civilization Wars: Advanced, testing was carried out at the end of the development of Civilization Wars: Advanced, which tested for high-level functionality of the game.

The testing included the following high-level functionality:

* Units & Buildings
  + User can train units when it is their turn and they have enough money
  + User cannot train units when it is not their turn
  + User cannot train units if they do not have enough money
  + User can build structures when it is their turn and they have enough money and is in the designated area
  + User cannot build structures that is outside the designated area
  + User cannot build structures when it is not their turn
  + User cannot build structures if they do not have enough money
* Mechanics
  + User can move units if it is their turn
  + User can attack enemy units if it is their turn
  + User cannot move a unit if it has already attacked
  + User can end turn when it is their turn
  + Ending the turn will reset the movement range and attack status of their units
  + Ending the turn will advance the turn onto the next player in the queue
  + Ending the turn will add money to the next player’s money
  + Eliminating an enemy unit or structure will add a reward to the player’s money
  + Money is deducted when the player trains a unit
  + Money is deducted when a player builds a structure
  + Player loses when their Base is destroyed (Free For All)
  + Player wins when they are left standing (Free For All)
  + Team loses if all their players are defeated (Cooperative)
  + Team wins if at least one member is left standing (Cooperative)
  + User can only attack once per turn, per unit
  + User has a time limit to finish their turn
  + Terrain buffs and debuffs are applied accordingly
* Single Player
  + User can play a 1v1 (Human vs AI)
  + User can play a 2v2 against AI (Human & AI vs AI & AI)
  + User can play a Free For All (Human vs AI vs AI vs AI)
* Multiplayer
  + Users can play 1v1 (Human vs Human)
  + Users can play 3 Player Free For All (Human vs Human vs Human)
  + Users can play 4 Player Free For All (Human vs Human vs Human vs Human)
* AI (Artificial Intelligence)
  + AI will build structures if it is their turn and it has enough money
  + AI will train units if it is their turn and it has enough money
  + AI will move units if it is their turn
  + AI will attack enemy units if it is their turn
* Interaction
  + Users can select units using the mouse
  + Users can end turn with a button
  + Users can attack units with the mouse
  + Users can attack units with a button
  + Users can move units with the mouse
  + Users can move units with a button
  + Users can pan the map with the mouse
  + Users can pan the map with the keyboard
  + Users can zoom in and zoom out with the keyboard
* UI (User Interface)
  + Minimap updates when a unit is trained, a structure is built, a unit has moved, a unit has been eliminated, a structure has been eliminated
  + Displays current player name
  + Displays player’s colour
  + Displays player’s ally
  + Displays player’s money
  + Displays the number of day (number of turns)
  + Displays information of the selection
  + Displays picture of the selection
  + Displays health bar of units and buildings
  + Displays buttons accordingly
  + Displays moveable cells
  + Displays attackable cells
  + Displays cells accordingly
* Main Menu
  + User can start a single player game
  + User can host a game
  + User can join a hosted game
  + User can show instructions

The testing excluded the following functionality:

* Available research for upgradeable units and structures
  + Was not implemented (not enough time)
* More than one map
  + Was not implemented (not enough time)
* Training units and building structures takes a number of days

Was not implemented (not enough time)

# High-Level Functionality Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Input | Output | Result (Pass or Fail) |
| TC001 | User left-clicks on Barracks and left-clicks on Marine button | A Marine spawned if the user had enough money and it was their turn, money is deducted | Pass |
| TC002 | User left-clicks on a Factory and left-clicks on Tank button | A Tank spawned if the user had enough money and it was their turn, money is deducted | Pass |
| TC003 | User left-clicks on their Base and left-clicks on the Barracks button and left-clicks in a highlighted cell | A Barracks spawned if the user has enough money and it was their turn, money is deducted | Pass |
| TC004 | User left-clicks on their Base and left-clicks on the Barracks button and left-clicks in a non-highlighted cell | A Barracks did not spawn in the left-clicked cell, money was not deducted | Pass |
| TC005 | User left-clicks on their Base and left-clicks on the Factory button and left-clicks in a highlighted cell | A Factory spawned if the user has enough money and it was their turn, money was deducted | Pass |
| TC006 | User left-clicks on the Base and left-clicks on the Factory button and left-clicks in a non-highlighted cell | A Factory did not spawn in the left-clicked cell, money was not deducted | Pass |
| TC007 | User left-clicks on a unit and right-clicks on a highlighted cell | The unit moved to the cell | Pass |
| TC008 | User left-clicks on a unit and right-clicks on a non-highlighted cell | The unit did not move to the cell | Pass |
| TC009 | User left-clicks on a unit that has attacked and right-clicks on a highlighted cell | The unit did not move to the cell | Pass |
| TC010 | User left-clicks on the end day button | The turn was ended and the turn of the next player began, resetting the movement range and attack status of units and added money to the next player | Pass |
| TC011 | User left-clicks on a unit and right-clicks on a red highlighted enemy unit | The unit moved and attacked the unit | Pass |
| TC012 | User left-clicks on a unit and right-clicks on a non-highlighted enemy unit | The unit did not move and attack the unit | Pass |
| TC013 | User left-clicks on a unit and right-clicks on a red highlighted enemy structure | The unit moved and attacked the structure | Pass |
| TC014 | User left-clicks on a unit and right-clicks on a non-highlighted enemy structure | The unit did not move and attack the structure | Pass |
| TC015 | User left-clicks on a unit and right-clicks on a highlighted enemy unit which will be eliminated | The unit moved and attacked and eliminated the unit, giving the player money | Pass |
| TC016 | User left-clicks on a unit and right-clicks on a highlighted enemy structure which will be eliminated | The unit moved and attacked and eliminated the structure, giving the player money | Pass |
| TC017 | User left-clicks on a unit and right-clicks on a highlighted enemy Base which will be eliminated | The unit move and attacked and eliminated the Base, eliminated all that player’s units and structures | Pass |
| TC018 | User left-clicks on the Single Player button on the Main Menu and creates a 1v1 game | A 1v1 game was created as Human vs AI | Pass |
| TC019 | User left-clicks on the Single Player button on the Main Menu and creates a 2v2 game | A 2v2 game was created as Human & AI vs AI & AI | Pass |
| TC020 | User left-clicks on the Single Player button on the Main Menu and creates a Free For All game | A Free For All game was created as Human vs AI vs AI vs AI | Pass |
| TC021 | User left-clicks on the Host Game button, waits for one player and starts the game | A 1v1 game was created as Human vs Human | Pass |
| TC022 | User left-clicks on the Host Game button, waits for two players and starts the game | A 3 Player FFA game was created as Human vs Human vs Human | Pass |
| TC023 | User left-clicks on the Host Game button, waits for three players and starts the game | A 4 Player FFA game was created as Human vs Human vs Human vs Human | Pass |
| TC024 | User creates a Single Player game with AI and left-clicks on the end turn button | AI built structures and trained units and attacked enemy units and structures when in range | Pass |
| TC025 | User left-clicks on a unit and left-clicks on the Move button and left-click on a highlighted cell | The unit moved to the cell | Pass |
| TC026 | User left-clicks on a unit and left-clicks on the Move button and left-clicks on a non-highlighted cell | The unit did not move to the cell | Pass |
| TC027 | User left-clicks on a unit and left-clicks on the Attack button and left-clicks on a red highlighted enemy unit | The unit moved and attacked the unit | Pass |
| TC028 | User left-clicks on a unit and left-clicks on the Attack button and left-clicks on a non-highlighted enemy unit | The unit did not move and attack the unit | Pass |
| TC029 | User moves mouse to the north edge | The view moved north | Pass |
| TC030 | User moves mouse to the south edge | The view moved south | Pass |
| TC030 | User moves mouse to the east edge | The view moved east | Pass |
| TC031 | User moves mouse to the west edge | The view moved west | Pass |
| TC032 | User presses the up arrow on the keyboard | The view moved north | Pass |
| TC033 | User presses the down arrow on the keyboard | The view moved south | Pass |
| TC034 | User presses the right arrow on the keyboard | The view moved east | Pass |
| TC035 | User presses the left arrow on the keyboard | The view moved west | Pass |
| TC036 | User presses the - key on the keyboard | The view zooms out one increment | Pass |
| TC037 | User presses the + key on the keyboard | The view zooms in one increment | Pass |
| TC038 | User trains a unit, builds a structure, moves a unit, eliminates a unit and eliminates a structure | The minimap was updated accordingly | Pass |
| TC039 | User creates a Single Player game with the name “Player” | “Player” was shown as the current player in the correct colour (blue) | Pass |
| TC040 | User creates a Single Player game | Player’s colour was shown correctly (blue) | Pass |
| TC041 | User creates a Single Player 2v2 game | Player’s ally was shown correctly (AI Player 1) | Pass |
| TC042 | User creates and plays a Single Player game | Player’s money began with the initial amount (500) and was deducted accordingly when units were trained and structures were built and money was added when the player eliminated enemy units and structures | Pass |
| TC043 | User creates a Single Player game and clicks the end turn button | The number of days (number of turns) was incremented with each press | Pass |
| TC045 | User selects a unit or structure or terrain | The correct information and image was shown depending on the selection | Pass |
| TC046 | User creates a Single Player game | A health bar was shown for every unit and building on the map | Pass |
| TC047 | User attacks an enemy unit or structure | The stats of the attacked unit or structure was adjusted accordingly, as well as the health bar | Pass |
| TC048 | User clicks on a unit | Attack, Move, End Day and Help buttons are shown | Pass |
| TC049 | User clicks on a Barracks | Marine, End Day and Help buttons are shown | Pass |
| TC050 | User clicks on a Factory | Tank, End Day and Help buttons are shown | Pass |
| TC051 | User clicks on a Base | Barracks, Factory, End Day and Help buttons are shown | Pass |
| TC052 | User clicks an enemy structure | Barracks and Factory buttons are disabled | Pass |
| TC053 | User clicks an enemy unit | Attack and Move buttons are disabled | Pass |
| TC054 | User left-clicks on the Host Game button, waits for three players and starts the game | A 4 Player FFA game was created as Human vs Human vs Human vs Human, with the correct colours on each client | Pass |
| TC055 | User left-clicks on the Host Game button, waits for three players and starts the game | A 4 Player FFA game was created as Human vs Human vs Human vs Human and only the Help button was enabled for players waiting for their turn | Pass |
| TC056 | User clicks on a unit | Shows moveable cells and attackable cells | Pass |
| TC057 | User clicks on the Instructions button on the Main Menu | Instructions are shown | Pass |

# Usability Testing

Some usability testing was carried out, but not as much as we had hoped as there was a lot of functionality that needed to be implemented. Usability testing was carried out during week eight of the project where our supervisor, Jay Young, organised a get together, inviting a number of project groups to try out the games developed by other groups. We took this opportunity as a chance to carry out usability testing, where we allowed other teams to try our game and took on board any feedback that they threw our way.

We received feedback related to the way units are represented. Units are actually a group, or a squad, of units. For example, a Marine unit is made up of a group of ten Marines. This means that if half of the Marine’s health is left, this means that half of the Marines are left in that group, which ultimately means less attack damage as there are less Marines. This was never shown in anyway during this stage of development and it was difficult to show the number of Marines alive in a Marine unit individually. In order to overcome this problem of showing the number of Marines in a Marine unit and it’s attack damage, we showed it as a statistic which is shown when a unit is selected.

Another feature which was criticised was the fact that when you right click on a unit to attack with the currently selected unit, the unit will move and attack at the same time, as attacking is only possible when the unit is directly next to the unit that is to be attacked. So, the unit will move automatically to the cell north, south, east, or west of the unit that is to be attacked. Users have no control over this from the way the algorithm searches to move and attack. Users would prefer to have control of the movement as there were sometimes situations which the user had advantage over next moves if they were able to choose which cell the unit ended up if a unit was attacked.

# Unit Testing

For unit testing of our game, we created JUnit classes, testing the major classes and it’s methods which made up the mechanics and playability of the game. However, various UI functionality such as mouse clicks was hard to simulate in JUnit classes and therefore, were excluded from the unit tests. The assertEquals method was used for comparing the expected output with the actual output. JUnit classes were created based on the Unit, Building, Cell and Player classes. Below is a table of the breakdown of the JUnit classes, found in the Test Package folder.

|  |  |  |
| --- | --- | --- |
| JUnit Class | Method Name | Description |
| UnitTest.java | testGetCurrentMoveRange | Test for returning the current move range of a Unit |
| UnitTest.java | testGetBaseHealth1 | Test for returning the base health of a Marine |
| UnitTest.java | testGetBaseHealth2 | Test for returning the base health of a Tank |
| UnitTest.java | testGetCurrentHealth1 | Test for returning the current health of a Marine |
| UnitTest.java | testGetCurrentHealth2 | Test for returning the current health of a Tank |
| UnitTest.java | testGetCost1 | Test for returning the cost of a Marine |
| UnitTest.java | testGetCost2 | Test for returning the cost of a Tank |
| UnitTest.java | testGetReward1 | Test for returning the reward of a Marine |
| UnitTest.java | testGetReward2 | Test for returning the reward of a Tank |
| UnitTest.java | testGetBaseAttackDamage1 | Test for returning the base attack damage of a Marine |
| UnitTest.java | testGetBaseAttackDamage2 | Test for returning the base attack damage of a Tank |
| UnitTest.java | testGetBaseMoveRange1 | Test for returning the base move range of a Marine |
| UnitTest.java | testGetBaseMoveRange2 | Test for returning the base move range of a Tank |
| UnitTest.java | testGetBaseNumberUnitsInGroup1 | Test for returning the base number of units in a Marine |
| UnitTest.java | testGetBaseNumberUnitsInGroup2 | Test for returning the base number of units in a Tank |
| UnitTest.java | testGetSingleUnitHealth1 | Test for getting the health of a single unit of a Marine |
| UnitTest.java | testGetSingleUnitHealth2 | Test for getting the health of a single unit of a Tank |
| UnitTest.java | testGetSingleUnitAttack1 | Test for getting the attack of a single unit of a Marine |
| UnitTest.java | testGetSingleUnitAttack2 | Test for getting the attack of a single unit of a Tank |
| UnitTest.java | testGetCell | Test for getting the cell of a Marine |
| UnitTest.java | testSetCell | Test for setting the cell of a Marine |
| UnitTest.java | testGetName1 | Test for getting the name of a Marine |
| UnitTest.java | testGetName2 | Test for getting the name of a Tank |
| BuildingTest.java | testGetCurrentHealth1 | Test for getting the current health of a Base |
| BuildingTest.java | testGetCurrentHealth2 | Test for getting the current health of a Barracks |
| BuildingTest.java | testGetCurrentHealth3 | Test for getting the current health of a Factory |
| BuildingTest.java | testGetBaseHealth1 | Test for getting the base health of a Base |
| BuildingTest.java | testGetBaseHealth2 | Test for getting the base health of a Barracks |
| BuildingTest.java | testGetBaseHealth3 | Test for getting the base health of a Factory |
| BuildingTest.java | testGetCost1 | Test for getting the cost of a Base |
| BuidlingTest.java | testGetCost2 | Test for getting the cost of a Barracks |
| BuildingTest.java | testGetCost3 | Test for getting the cost of a Factory |
| BuildingTest.java | testGetReward1 | Test for getting the reward of a Base |
| BuildingTest.java | testGetReward2 | Test for getting the reward of a Barracks |
| BuildingTest.java | testGetReward3 | Test for getting the reward of a Factory |
| BuildingTest.java | testGetCell | Test for getting the cell of a Base |
| BuildingTest.java | testGetName1 | Test for getting the name of a Base |
| BuildingTest.java | testGetName2 | Test for getting the name of a Barracks |
| BuildingTest.java | testGetName3 | Test for getting the name of a Factory |
| CellTest.java | testGetTerrain1 | Test for getting the terrain of a Cell having a Flat terrain |
| CellTest.java | testGetTerrain2 | Test for getting the terrain of a Cell having a Water terrain |
| CellTest.java | testGetTerrain3 | Test for getting the terrain of a Cell having a Bridge terrain |
| CellTest.java | testGetTerrain4 | Test for getting the terrain of a Cell having a Mountain terrain |
| CellTest.java | testGetTerrain5 | Test for getting the terrain of a Cell having a Forest terrain |
| CellTest.java | testGetUnit1 | Test for getting the Unit from an occupied Cell |
| CellTest.java | testGetUnit2 | Test for getting the Unit from an empty Cell |
| CellTest.java | testSetUnit | Test for setting the Unit of a Cell |
| CellTest.java | testGetBuilding1 | Test for getting the Building from an occupied Cell |
| CellTest.java | testGetBuilding2 | Test for getting the Building from an empty Cell |
| CellTest.java | testSetBuilding | Test for getting the Building of a Cell |
| CellTest.java | testGetCellPosX | Test for getting the X coordinate of a Cell |
| CellTest.java | testGetCellPosY | Test for getting the Y coordinate of a Cell |
| CellTest.java | testCanMoveUnitTo1 | Test for moving a Unit onto a Cell occupied by a Building |
| CellTest.java | testCanMoveUnitTo2 | Test for moving a Unit onto a non-occupied Cell |
| PlayerTest.java | testGetPlayerName | Test for getting the name of the Player |
| PlayerTest.java | testSetPlayerName | Test for setting the name of the Player |
| PlayerTest.java | testGetEliminated | Test for getting the eliminated status of a Player |
| PlayerTest.java | testSetEliminated | Test for setting the eliminated status of a Player |

# Appendix

**GTC** - Game Test Case  
**CTC** - Cell Test Case

**BTC** - Building Test Case  
**UTC** - Unit Test Case  
**NTC** - Networking Test Case  
**ITC** - Interface Test Case  
**MTC** - Menu Test Case