Sean McManes
CS Senior Design II: Attack of the Beat
Professor Annexstein
22 January 2020

Assignment 1: Test Plans

Part I: Description of Overall Test Plan

The approach taken to test this project will take place on an individual parts basis and then an overall complete project basis. Throughout each test, particularly with the individual parts, depending on what results need to be examined, the project will produce data necessary for the tests. This data will be analyzed and compared to the intended results, and from that conclusions will be drawn, and further improvements/changes will be determined from the findings. After these parts are working as intended, they will be put together towards the overall project and then final tests can be conducted to make sure that the parts work together as intended, and the overall project is working as intended. Any improvements/changes would be made if needed depending on the outcomes.

Part II: Test Case Descriptions

1.1 Game Map and Texture Test 1
1.2 This test ensures that the first round of map textures and map generators work as intended
1.3 This test will run the map generator with the first round of textures to show that the generator is picking the correct textures for the given test cases, and that the textures are correctly designed to work with intended situations.
1.4 Inputs: First round textures and test map layout
1.5 Outputs: Map made from textures given based on map layout
1.6 Normal
1.7 Whitebox
1.8 Functional
1.9 Unit Test

2.1 Player Texture and Animation Test 1
2.2 This test ensures that the initial player textures and animations work as intended
2.3 This test will have the player character complete various test actions to show that the textures flow together to make the animations look correct. This will also test the timing between each part of the animation, and add/remove time if needed
2.4 Inputs: Player textures and test actions
2.5 Output: Player animation matching the given actions
2.6 Normal
2.7 Whitebox
2.8 Functional
2.9 Unit

3.1 Game Map and Texture Test 2
3.2 This test ensures that the game map generator handles exceptions as intended
3.3 This test will run the map generator with the first round of textures to show that the generator correctly identifies errors in the given map design, displays proper warnings, and displays proper textures in the areas that have errors.
3.4 Inputs: First round textures and test map layout with errors
3.5 Output: Map made from textures given based on map layout with proper error messages and error textures
3.6 Abnormal
3.7 Whitebox
3.8 Functional
3.9 Unit

4.1 Music Analyzer Test 1
4.2 Run the music analyzer over test music
4.3 This test will run the music analyzer over the first set of test music and see if the output from the analyzer match what was expected. The major part of this is making sure that the analyzer is picking good points in the music for specific actions, and any issues would be adjusted.
4.4 Inputs: Test music (Set 1)
4.5 Outputs: Action codes with timestamps from trigger points
4.6 Normal
4.7 Whitebox
4.8 Functional
4.9 Unit

5.1 Music Analyzer Test 2
5.2 Run the music analyzer after modifications over different test music
5.3 This test will run the music analyzer over another set of test music and see if the output from the analyzer match what was expected. This is to see if the adjustments made from the prior test cause incorrect outputs from other songs, and depending on the outcome, some tweaks to the design may be done. Generally, the type of music will be the same.
5.4 Inputs: Test music (Set 2)
5.5 Outputs: Action codes with timestamps from trigger points
5.6 Normal
5.7 Whitebox
5.8 Functional
5.9 Unit

6.1 Alternative Music Analyzer Test
6.2 Run the music analyzer over different genre test music
6.3 This test will run the analyzer over another set of test music from a different genre from the music from the other tests. This is to see if the settings made in the prior tests work well with other genres of music, and if not, then what adjustments can be made to get the analyzer working
6.4 Inputs: Alternative test music
6.5 Outputs: Action codes with timestamps from trigger points
6.6 Normal
6.7 Whitebox
6.8 Functional
6.9 Unit

7.1 Player/Object Movement and Actions Test 1

7.2 Test the various movements and actions of the player character and other objects

7.3 This test will ensure that the player movement and actions are working as expected and that there are no disconnect or lag with them. This test will also allow us to adjust settings, such as how far characters move per frame and so on.

7.4 Inputs: Test user input (movements and actions)

7.5 Outputs: Displaying player movements and actions

7.6 Normal

7.7 Whitebox

7.8 Functional

7.9 Unit

8.1 Gameplay and Music General Test 1

8.2 Test the game loop with music implementation

8.3 This test will ensure that the music playing with the game is outputting the action codes, as tested before (as a separate part), at the correct times and does not run into issues over time. Major points for investigation are desync from the game music, stuttering, incorrect actions, and more.

8.4 Inputs: Test music and generic game inputs

8.5 Outputs: Game display, music, and corresponding action codes

8.6 Normal

8.7 Whitebox

8.8 Functional

8.9 Unit

9.1 Player/Object Movement and Actions Test 2

9.2 Test the various movements and actions of the player character and other objects with music

9.3 This test will ensure that the music, action points gained from the music, and object movements from these action points are working in sync and as expected. Any issues that arise with the implementation of these action points and actual object movement can be addressed at this point.

9.4 Inputs: Test music

9.5 Outputs: Displaying object movements and actions, music, and action codes with timestamps

9.6 Normal

9.7 Whitebox

9.8 Functional

9.9 Unit

10.1 Gameplay and Music General Test 2
10.2 Test the game loop with music implementation and actions from the music
10.3 This test will ensure that the music playing with the game is outputting the action codes, as tested before (as a separate part), at the correct times and does not run into issues over time. This will add a player-controlled character, and several computer/music controlled characters to mainly stress test the game and make sure that these elements do not create conflicts.
10.4 Inputs: Test music and player test input
10.5 Outputs: Game display, music, and corresponding action codes
10.6 Normal
10.7 Whitebox
10.8 Performance
10.9 Unit

Part III: Test Case Matrix

	Normal/Abnormal	**Blackbox/Whitebox**	**Functional/Performance**	**Unit/Integration**
1	Normal	Whitebox	Functional	Unit
2	Normal	Whitebox	Functional	Unit
3	Abnormal	Whitebox	Functional	Unit
4	Normal	Whitebox	Functional	Unit
5	Normal	Whitebox	Functional	Unit
6	Normal	Whitebox	Functional	Unit
7	Normal	Whitebox	Functional	Unit
8	Normal	Whitebox	Functional	Unit
9	Normal	Whitebox	Functional	Unit
10	Normal	Whitebox	Performance	Unit