Test Cases and Test Coverage Report

Introduction and Purpose

This document serves to highlight and demonstrate several test cases for the CastleSky application. This document will include several test cases deemed especially important to the development process and lifecycle of the CastleSky game development. This list of test cases is not exhaustive, and ideally, a more comprehensive and detailed set of test cases should be specified.

Intended Audience

This document is intended for the current and future developers of CastleSky, in particular, the Quality Assurance Teams of any developer team

Document Flow

We shall be separating the test cases into unit tests, integration tests and finally system tests. These three categories of tests were deemed necessary for the development process of CastleSky.

Unit testing is important as it ensures the robustness and functionality of each individual component in the application. This is especially important as the entire game is composed of multiple components, and failures/unexpected behaviours occurring in even a single component could have potentially disastrous implications on the functionality of the rest of the application.

Integration testing is important in the development of CastleSky because of our heavy focus on the involvement of various minigames in the game. Integration testing is necessary to firstly, establish that the minigames are able to work within the CastleSky environment, as well as to initialize and set up the interface allowing future developers to add their own unique games to CastleSky's repertoire.

System testing is important to the development of CastleSky because the target audience are students, who tend to be a less mature audience. As such, we can expect them to have a very short attention span, and system failures resulting in a nonresponsive user interface is likely to turn them away from the game. This would counter the original goal of CastleSky - which is the process of making learning fun for students.

Naming Convention

The test cases are named using the following convention:

[developer_team_abbreviation]-[test_type]-[unique_test_number], where:

developer_team_abbreviation refers to the initials of the developer team adding this test case. For instance, the developer team 'CastleInTheSky' would have the initials 'CITS'

test_type refers to the category of test. Examples of three such categories are Unit Tests (UT), Integration Tests (IT) and System Tests (ST). Future developers may add other categories to this list.

unique_test_number refers to the unique ID for the test case. Essentially, this acts as a global counter for the total number of test cases. Note that this ID does not refresh for different categories of test cases.
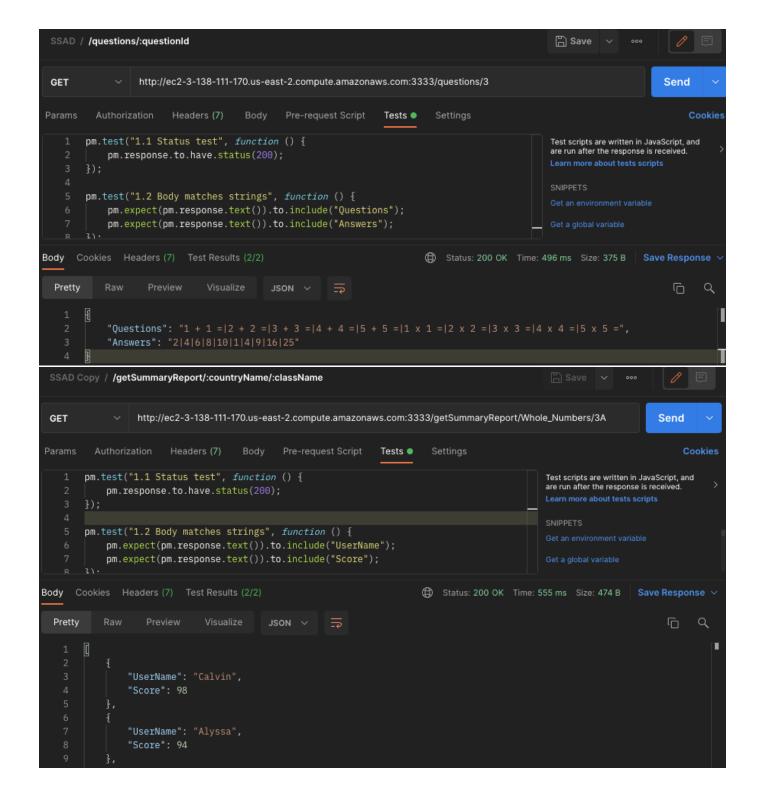
Unit Tests

Test Case Name	Action	Description	Expected Result	Actual Result	Status
CS-UT-001	World Navigation - Left Movement	User presses 'A' key.	The user moves left.	The user moves left.	Pass
CS-UT-002	Student Menu Navigation	After logging in, the user is presented with a list of World names. The user then clicks on a world to load	The World corresponding with the World name chosen by the student is loaded, and the student is transported into the selected World	The chosen World is indeed loaded, and the student is transported into the selected World	Pass
CS-UT-003	Teacher Summary Report Generation - Choose Class	When the teacher tries to generate a summary report, they have to first choose the class.	After choosing the class, the corresponding worlds open to the class should be in the World dropdown.	The Worlds that the class has access to are in the World dropdown.	Pass

CS-UT-004	Teacher Summary Report Generation - Choose World	After choosing the class, the teacher will have to choose a World.	After choosing the world, the corresponding countries in that world that the class has access to will be in the Countries dropdown.	The Countries that the class has access to are in the Country dropdown.	Pass
CS-UT-005	Teacher Summary Report Generation - Choose Country	After choosing a country, the teacher will get the generated summary report.	The summary report for the particular class and country is generated.	The summary report for the students in the class who played the final level game in the country is generated.	Pass
CS-UT-006	Teacher Generate Access Code	After choosing the World and Country in similar steps as the previous Unit Tests, the teacher can click "Generate Access Code" to have an access code generated.	An access code will appear on the screen. The teacher can click "Share to Whatsapp", which will open Whatsapp Web.	An access code appears on the screen. The teacher clicks "Share to Whatsapp", which opens Whatsapp Web with a pre-generated message.	Pass

Integration Tests

Test Case Name	Action	Description	Expected Result	Actual Result	Status
CS-IT-001	Minigame Entry	While navigating an Open World, the player must be able to easily enter a minigame and begin playing	When approaching the minigame area, the screen changes to reflect the minigame	When inside a minigame area, the screen immediately changes, allowing the player to play the game	Pass
CS-IT-002	Retrieving Mini Game Questions	When playing a game, the questions must be correct to the level that the user was selected.	When playing a game in Country, the game must retrieve the correct questions based on the City Id.	When playing a game in a country, the game retrieves the correct questions	Pass
CS-IT-003	Teacher Retrieving Leaderboard	When a teacher wants to see a leaderboard for a particular country, the teacher has to select the world and country.	The teacher should see an ordered list of student names with the scores.	The teacher sees an ordered list of student names with scores.	Pass
CS-IT-004	Update Score	When a student completes a final level game in a Country, the game updates the DB with the latest score.	The DB should be populated with the new score, together with the identifying information.	The DB is correctly updated with the new score and identifying information.	Pass
CS-IT-005	Student Registration	When a student registers for the game, a new entry is added to the student.	The DB is updated with the new student, with the password hashed, and the email, and the student is assigned a UserId.	The DB is correctly updated with the new student's information, and the student is assigned a UserId.	Pass
CS-IT-006	Country Access Code Validation	When a student tries to enter a new country, they will be met with a guard. The guard will request for an access code, and the student has to key in a correct access code for the guard to move away. The correct access code is retrieved from the DB using the countryId.	The guard should allow the user to pass with the right access code.	The guard allows the user to pass with the right access code.	Pass

For our game to database integration, we have done some Postman test scripts. To run the test scripts, install Postman and download "SSAD. postman_collection.json" from our backend repository, and import it into Postman. Here is an example of how the test script looks like:

Save

GET ∨ http://ec2-3-138-111-170.us-east-2.compute.amazonaws.com:3333/questions/3

Send

Params   Authorization   Headers (7)   Body   Pre-request Script   Tests ●   Settings   Cookies

```
1  pm.test("1.1 Status test", function () {
2      pm.response.to.have.status(200);
3  });
4
5  pm.test("1.2 Body matches strings", function () {
6      pm.expect(pm.response.text()).to.include("Questions");
7      pm.expect(pm.response.text()).to.include("Answers");
8  });
```

Test scripts are written in JavaScript, and are run after the response is received.
Learn more about tests scripts

SNIPPETS
Get an environment variable
Get a global variable

Body   Cookies   Headers (7)   Test Results (2/2)   ⊕ Status: 200 OK   Time: 496 ms   Size: 375 B   Save Response ∨

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  {
2      "Questions": "1 + 1 =|2 + 2 =|3 + 3 =|4 + 4 =|5 + 5 =|1 x 1 =|2 x 2 =|3 x 3 =|4 x 4 =|5 x 5 =",
3      "Answers": "2|4|6|8|10|1|4|9|16|25"
4  }
```

Save

GET ∨ http://ec2-3-138-111-170.us-east-2.compute.amazonaws.com:3333/getSummaryReport/Whole_Numbers/3A

Send

Params   Authorization   Headers (7)   Body   Pre-request Script   Tests ●   Settings   Cookies

```
1  pm.test("1.1 Status test", function () {
2      pm.response.to.have.status(200);
3  });
4
5  pm.test("1.2 Body matches strings", function () {
6      pm.expect(pm.response.text()).to.include("UserName");
7      pm.expect(pm.response.text()).to.include("Score");
8  });
```

Test scripts are written in JavaScript, and are run after the response is received.
Learn more about tests scripts

SNIPPETS
Get an environment variable
Get a global variable

Body   Cookies   Headers (7)   Test Results (2/2)   ⊕ Status: 200 OK   Time: 555 ms   Size: 474 B   Save Response ∨

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  [
2      {
3          "UserName": "Calvin",
4          "Score": 98
5      },
6      {
7          "UserName": "Alyssa",
8          "Score": 94
9      },
```

System Tests

Test Case Name	Action	Expected Result	Actual Result	Status
CS-ST-001	Smoke Test	The application responds to the user's action as designed.	The application responds to the user's action as designed.	Pass

CS-ST-002	Installation Test	CastleSky is successfully installed and ready to use on any computer device complying with our minimum system requirements as stated in the SRS	CastleSky is successfully installed and ready to use on any reasonable computer device.	Pass
CS-ST-003	Stress Testing	100 simultaneous users should be logged into the CastleSky system (from different machines) and be able to play minigames without encountering significant delays	100 simultaneous users are able to play the Geometry minigame without encountering user input lag issues	Pass
CS-ST-004	Scalability Test	Database must be able to process queries from multiple application without crashing.	The database is updated based on the application usage of 100 users simultaneously.	Pass
CS-ST-005	Recoverability Test	The application is able to recover the data of the user in case the application closes during its operation.	The application is able to recover the data of the user in case the application closes during its operation.	Pass
CS-ST-006	Security Test	The application is able to protect the database from any unauthorized intrusion or security breaches.	The application is able to protect the database from any unauthorized intrusion or security breaches.	Pass