**Senior Project – Master Test Plan**

**Student:** Jonathan Flum

**Degree and Major:** Bachelor of Arts in Applied Computing, Business Concentration

**Project Advisor:** Prof. O’Neill

**Expected Graduation Date:** Indefinite

**Project Title:** Cognitive Trainer for Computer Science Professionals

**Test Plan Identifier:** Master Test Plan for “Cog\_Trainer” (v0.230411a)

**Introduction:**

The purpose of this test plan is to ensure satisfactory programmatic and design implementation of the functional, appearance, style, usability, and performance fit criterion identified within this project’s Requirements document. In addition, deployment of the application via GitHub pages will be conducted to a practical extent, ensuring reasonable user access and availability. Because a high level of system compatibility is expected to be achieved, permutations of specific hardware and/or software combinations other than the test configuration will not be evaluated and is out of scope.

**References:**

* Project Repository: <https://github.com/jflum/CSU-Senior-Project>
* Proposal: […/jflum/CSU-Senior-Project/blob/master/docs/Proposal.docx](https://github.com/jflum/CSU-Senior-Project/blob/master/docs/Proposal.docx)
* Requirements: […/jflum/CSU-Senior-Project/blob/master/docs/Requirements.pptx](https://github.com/jflum/CSU-Senior-Project/blob/master/docs/Requirements.pptx)

**Test Items:**

* “Cog\_Trainer” (v0.230411a)

**Features to be Tested:**

* Please refer to Volere Snow Cards within Requirements document.

**Item Pass/Fail Criteria:**

* Please refer to Volere Snow Cards within Requirements document.
* Note that for academic purposes, fit criteria requiring representative sample sizes and/or target populations may be simulated or extrapolated from similar data sets.

**Features Not to Be Tested:**

* Hardware, software, and network configurations other than that specified as the de facto testing environment. Due to the utilization of a universal development platform, a WebGL/HTML5 build target, and hosting via GitHub Pages, compatibility of all modern browsers is expected, irrespective of most system specifications.
* Potential for security vulnerabilities, specifically data manipulation by unauthorized access/penetration. Due to the non-critical and non-personally identifiable type of information the application will store, this type of testing has a low value proposition.

**Approach:**

Testing will inherently take place with a white box methodology, as the application’s designer and programmer will also serve as the tester. Functional requirements will be evaluated in an automated manner, by way of a program-specific validation script—the results of which will output to console. This modality will be limited to development builds. Conversely, performance testing will be conducted manually on the live (published), commercial build of the application. Appropriate logs/test results will be recorded and made available for review.

**Testing Levels:**

* Unit Test Plan: not indicated as a separate testing measure outside of IDE debug
* Integration Test Plan: not indicated as a separate testing measure outside of IDE debug
* System Test Plan: not indicated provided current project applicability/scope
* Acceptance Test Plan: shall be validated against functional Requirements
* Deployment/Hosting Test Plan: shall be validated against performance Requirements

**Suspension Criteria and Resumption Requirements:**

* Should the testing of Functional Requirements 1.01-1.06 result in a less than 100% pass rate, subsequent testing will not commence; further development is required.
* Testing of Requirement Sections 2-4 will resume only upon the absolute qualification of Section 1. Previously passed fit criteria must be revalidated during each iteration.

**Test Deliverables:**

* Test Case Log: [...jflum/CSU-Senior-Project/blob/master/tests/debug\_log\_230914.txt](https://github.com/jflum/CSU-Senior-Project/blob/master/tests/debug_log_230914.txt) Consists of five iterations, one for each level of difficulty, in an effort to test the most amount of gameplay variation in the fewest number of runs, with each instance consisting of a random permutation.
* Defect/Enhancement Logs: Due to the additional implementation of scripted fail state clauses for the test cases specified below, defect logs were not enduringly captured through the development process. Further enhancement or revision to the product that results in the loss of any passing status will generate an automated, pertinent console log, which may be maintained if desired.
* Performance & Usability Testing: Detailed within the Requirements document where applicable, and validated/substantiated by survey response of a sufficient sample size of the target audience. Further analysis of these results to be incorporated into final defense documentation and presentation.
* Test Case Summary (abridged to a single iteration):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Script/Test Location** | **Parent Call** | **Output** | **Status** |
| 1 | **main.cs** |  |  |  |
| 1.1 | main:SetDifficulty (at Assets/Scripts/main.cs:483) | main:Awake () (at Assets/Scripts/main.cs:87)\* | difficulty of ("novice") set, gameplay variables assigned | Pass |
| 1.2 | main:SetMode (at Assets/Scripts/main.cs:513) | main:Awake () (at Assets/Scripts/main.cs:89)\* | game mode of ("progressive") set, timer operation modified | Pass |
|  |  |  |  |  |
| 2 | **field.cs** |  |  |  |
| 2.1 | field:BuildField (at Assets/Scripts/field.cs:155) | main:Start () (at Assets/Scripts/main.cs:97)\* | a field with (2) tiles is valid | Pass |
| 2.1 | field:BuildField (at Assets/Scripts/field.cs:166) | main:Start () (at Assets/Scripts/main.cs:97)\* | qty (2) tiles of scale (3) with spacing (1) will fit on current field size (18) | Pass |
| 2.2 | field:BuildField (at Assets/Scripts/field.cs:178) | main:Start () (at Assets/Scripts/main.cs:97)\* | a field with (2) tiles can be drawn with (8) colors | Pass |
| 2.3 | field:BuildField (at Assets/Scripts/field.cs:189) | main:Start () (at Assets/Scripts/main.cs:97)\* | a field with (2) tiles can be drawn with (4) shapes | Pass |
| 2.4 | field:BuildField (at Assets/Scripts/field.cs:200) | main:Start () (at Assets/Scripts/main.cs:97)\* | a field with tile complexity of (2) is valid | Pass |
| 2.5 | field:BuildField (at Assets/Scripts/field.cs:213) | main:Start () (at Assets/Scripts/main.cs:97)\* | availColors: black, blue, green, orange, purple, red, yellow, white | Info |
| 2.6 | field:BuildField (at Assets/Scripts/field.cs:214) | main:Start () (at Assets/Scripts/main.cs:97)\* | availShapes: circle, rectangle, square, triangle | Info |
|  |  |  |  |  |
| 3 | **tile.cs** |  |  |  |
| 3.1 | tile:Populate (at Assets/Scripts/tile.cs:350) | field:BuildField (at Assets/Scripts/field.cs:269) | tile1 with id (#1) and complexity of (2) successfully populated | Pass |
| 3.2 | tile:Populate (at Assets/Scripts/tile.cs:350) | field:BuildField (at Assets/Scripts/field.cs:269) | tile2 with id (#2) and complexity of (2) successfully populated | Pass |
|  |  |  |  |  |
| 4 | **field.cs (cont.)** |  |  |  |
| 4.1 | field:GenerateQA (at Assets/Scripts/field.cs:307) | main:Start () (at Assets/Scripts/main.cs:99)\* | qty (1) unique queries is possible with (2) tiles of (3) attributes | Pass |
| 4.2 | field:GenerateQA (at Assets/Scripts/field.cs:321) | main:Start () (at Assets/Scripts/main.cs:99)\* | selected tileID for query: 1 | Info |
| 4.3 | field:GenerateQA (at Assets/Scripts/field.cs:322) | main:Start () (at Assets/Scripts/main.cs:99)\* | selected attribute for query: 1 | Info |
| 4.4 | field:GenerateQA (at Assets/Scripts/field.cs:373) | main:Start () (at Assets/Scripts/main.cs:99)\* | valid query set for selected tile & attribute: "What is the background color of tile #1? | Pass |
| 4.5 | field:GenerateQA (at Assets/Scripts/field.cs:374) | main:Start () (at Assets/Scripts/main.cs:99)\* | valid answer set for selected query: "green" | Pass |
|  |  |  |  |  |
| 5 | **main.cs (cont.)** |  |  |  |
| 5.1 | main:Update () (at Assets/Scripts/main.cs:127) | none; called as a factor of frame update behavior | correct answer submitted | Info |

\*Note: functions in subsequent iterations are called from Update() within the respective script.

**Test Environment:**

* Hardware: Dell Precision-5550
  + CPU: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
  + GPU: NVIDIA Quadro T1000 4.0 GB
  + RAM: 32.0 GB
* Operating System: Windows 10 Pro
  + Version: 22H2
  + Build: 19045.3348
  + WFE Pack: 1000.19044.1000.0
* Integrated Development Environment (IDE): Unity Professional
  + Version: LTS 2021.3.16f1
* Internet Browser: Google Chrome
  + Version: 117.0.5938.89 (64-bit)

**Estimate:**

Execution of this test plan, including plan construction is estimated to require (13) total manhours. Cost forecasting is not applicable to this project. Overages will be documented, however no mitigating actions are indicated. Approximate allocations of effort are as follows:

* Planning – 3 hrs.
* Test Construction and Integration – 4 hrs.
* Evaluation – 2 hrs.
* Record, Reporting, & Documentation – 2 hrs.
* Consolidation and Review – 1 hr.
* Approvals – 1 hr.

**Staffing and Training Needs:**

* Project Manager (professor)
  + Required skills: IT Project Management, to include application Development, Integration and Deployment
  + Additional training required: none
* Software Engineer (student)
  + Required skills: Foundational CS courses including procedural programming, object-oriented programming, and data structures, e.g.
  + Additional training required: none

**Responsibilities:**

* Michael O’Neill
  + Review, advise, and approve specified project deliverables.
  + Coordinate meeting schedule.
* Jonathan Flum
  + Develop testing methodologies, prepare and conduct evaluations, document results, perform analysis, determine corrective actions, and implement programmatic/design changes, as required.

**Risks:**

* Forfeiture of effort, either partial or complete, due to hardware failure and/or data loss.
  + Severity: high
  + Probability: low
  + Mitigation: Project data shall be backed up via Google Drive on a near-synchronous basis, milestone deliverables/versions shall be pushed to GitHub
  + Contingency: Restoration of data via cloud backup once hardware environment has been stabilized. Impact: 2+ hours.
* Delay of development (and associated test measures) as a result of a major and unavoidable framework upgrade, i.e., an updated Git API or Unity Core that creates incompatibility or instability between one or more technologies.
  + Severity: medium
  + Probability: medium
  + Mitigation: As practical as possible, project software and associated tools will not be upgraded mid-development, unless of a critical nature.
  + Contingency: Validation/test deployment of new integration(s) must be completed prior to further development. Impact: 1+ hours.

**Assumptions and Dependencies:**

* Development of this application, in addition to specific test modalities, rely on the availability of Unity Pro as an integrated development environment and platform, the license for which has been granted under the Unity Student Plan. Continued access is expected to be maintained for the duration of the project, up to initial release.
* Deployment of this application, with a proposed target of web-based public access, is subject to the availability of GitHub and GitHub Pages for the hosting of pertinent build data and serving of the end-user application, respectively. A high uptime ratio (>99.9%) is presumed based on historic performance.

**Approvals:**

Michael O’Neill shall have final approval/acceptance of project deliverables.

**Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **ECD** | **STATUS** | **Comments** |
| 1 | **Plan** |  |  |  |
| 1.1 | Draft Proposal | 02/28/22 | Complete | Deliverable |
| 1.2 | Draft Proposal and Requirements | 03/31/22 | Complete | Deliverable |
| 1.3 | Completed Proposal and Requirements | 04/12/22 | Complete | Deliverable; fork Sr. Project Repo and add documents, update readme |
|  |  |  |  |  |
| 2 | **Build** |  |  |  |
| 2.1 | Data/Objects | 02/20/23 | Complete | Primary class for information sets |
| 2.2 | User Interface | 03/20/23 | Complete | Input/output considerations |
| 2.3 | Functional Loop | 04/03/23 | Complete | Cradle-to-grave sequence |
| 2.4 | Graphic Design | 04/10/23 | Complete | Display formatting, responsiveness |
| 2.5 | Sound Design | 04/10/23 | Complete | Implement background music, sfx |
| 2.6 | “Significant Progress” | 04/25/23 | Complete | Deliverable; current state of above |
|  |  |  |  |  |
| 3 | **Test** |  |  |  |
| 3.1 | Create Test Plan | 03/20/23 | Complete | Deliverable; partially identified in FRD, user task/process oriented |
| 3.2 | Test & Document | 04/10/23 | Complete | Generate specification punch list and order by priority |
| 3.3 | Clear backlog | 04/25/23 | Complete | Finalize any outstanding product implementations |
| 3.4 | Generate Final Report | 9/14/23 | Complete | Create scripted validation tool that substantiates program operation |
|  |  |  |  |  |
| 4 | **Deploy** |  |  |  |
| 4.1 | Package and deliver | 8/31/23 | Complete | Deliverable; host application on GitHub Pages |
| 4.2 | Evaluate | 10/13/23 | In progress | Identify lessons learned, best practices, extendibility, etc. |
| 4.3 | Defense Documentation | 11/17/23 (tentative) | In progress | Deliverable; Final Report |
| 4.4 | Project Presentation | 11/17/23 (tentative) | In progress | Deliverable |