**Sinister Transistor**

**Software Requirements Specification**

**COP 4331, Spring 2016**

Team Name: The Mega Bytes

Team Members:

* Greg Kelso
* Mark Boutwell
* Joel Gardyasz

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 02/18/16 | Greg | Template |
| v1.0 | <date here> | <who> | <put comment to summarize the changes made in this version> |
| ... |  |  |  |

Contents of this Document

Introduction

* Software to be Produced
* Reference Documents
* Applicable Standards

Definition, Acronyms, and Abbreviations

Product Overview

* Assumptions
* Stakeholders
* Event Table
* Use Case Diagram
* Use Case Descriptions

Specific Requirements

* Functional Requirements
* Interface Requirements
* Physical Environment Requirements
* Users and Human Factors Requirements
* Documentation Requirements
* Data Requirements
* Resource Requirements
* Security Requirements
* Quality Assurance Requirements

Supporting Material

**Section 1: Introduction**

Software to be Produced:

* <Include one paragraph to identify the software product(s) to be produced. Refer the reader to the reference documents for more information.>

Reference Documents:

* Concept of Operations
* Project Plan
* <any other relevant documents>

Applicable Standards

* <You do not have to repeat the standards included in the project plan. Instead, cite any standards that are specific to the system requirements.>

Definitions, Acronyms, and Abbreviations

* <Include any that are needed to read this document or "none" if document is self-explanatory and no acronyms or abbreviations will be used>

**Section 2: Product Overview**

Assumptions:

* <List all the assumptions the developers are making. For example: assumptions about other systems this product will interface with; assumptions about the technological environment in which the product will operate (how much memory, what type of processor, ...); assumptions about availability and capability of COTS, GOTS, or other re-used products, ...>

Stakeholders:

* <A stakeholder is anyone who has an interest in the software to be developed. For example, the customer, the various classes of users, applicable regulatory agencies, ... List each category of stakeholder and give a phrase or a sentence to describe their interest or concerns>

Event Table:

* <An event table identifies all the external events to which the software must respond. This is a first step in determining the required overall system functionality. The event list should be consistent with the context diagram and the interest of each stakeholder. Make sure that exceptions are considered.>
* <Use the following table format:>

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Use Case Diagram

* <Include a use case diagram here. It should be consistent with all the above work. >

Use Case Descriptions:

* <Briefly describe each use case included in the above diagram. >

**Section 3: Specific Requirements**

<Use the following template for each requirement. >

|  |
| --- |
| No: <unique requirement number> |
| Statement: <the "shall" statement of the requirement> |
| Source: <source of the requirement> |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: <list any supporting diagrams, lists, memos, etc.> |
| Evaluation Method: <How can you tell if the completed system satisfies this requirement? > |
| Revision History: <who, when, what> |

3.1 Functional Requirements

|  |
| --- |
| No: 311 |
| Statement: The player shall be able to move up, down, left, and right, unless stopped by a wall. |
| Source: Player movement |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Move the player in the world and verify that the player is moving with the inputs from the keyboard and stopping at walls. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 312 |
| Statement: The software shall feature music and sound effects. |
| Source: Game design |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: While the game is running, verify that the background music is playing and that interactions between NPCs or other game entities produce the proper sound effects. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 313 |
| Statement: The player shall be able to navigate into other areas and rooms in their environment. These rooms shall load and open once the player has indicated their desire to enter. |
| Source: Player movement |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Verify that the next room loads and opens once the player has properly moved to go into that room. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 314 |
| Statement: The player and enemies shall have a set amount of health. If this is depleted, the player/enemy shall die. If the player dies the game shall end. |
| Source: Player health |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Verify that if the player dies that the game ends. Verify that if an enemy dies its sprite is removed from the game. |
| Revision History: Created by Joel on 2/26/16 |

3.2 Interface Requirements

|  |
| --- |
| No: 321 |
| Statement: The user shall be able to start a new game. |
| Source: Game requirement |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Verify that a new game can be created after a game has already been started. Make sure this new game is a brand new game with no prior user input or interaction. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 322 |
| Statement: The player shall be able to pause the current game to pull up a pause menu. |
| Source: Interface |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: While a game is in progress, verify that the pause menu can be brought up and that it stops all movement and progress in the current game. Verify that the options on the menu are functional. |
| Revision History: Created by Joel on 2/26/16 |

3.3 Physical Environment Requirements

* < Describe the environment in which the software must run. Physical environment requirements include:
  + type of equipment on which the software must run
  + location of the equipment
  + environmental considerations: temperature, humidity, ...
  + ...>
  1. User and Human Factors Requirements

|  |
| --- |
| No: 341 |
| Statement: The game shall support users of all skill levels and of all ages. |
| Source: User demographic |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Have people of varying ages and skill levels play the game and verify that it can be played by everyone with the proper introduction and learning of the game. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 342 |
| Statement: The game shall accommodate all players by having an ability to increase or decrease audio and font size. |
| Source: User accommodation |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Verify that the increase in font works in game and that there is a way to increase or decrease audio that is easy for the user to access. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 343 |
| Statement: The software should be bug free. The game should not let users exploit bugs to cheat and achieve a higher score. |
| Source: Player integrity |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Test the game thoroughly before releasing the game to everyone. If any bugs or exploits are found, then fix them. |
| Revision History: Created by Joel on 2/26/16 |

* 1. Documentation Requirements

|  |
| --- |
| No: 351 |
| Statement: The player shall be able view a manual consisting of game features and mechanics. |
| Source: Player manual |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Have an inexperienced player read the manual and determine what they’ve learned about the game. |
| Revision History: Created by Joel on 2/26/16 |

|  |
| --- |
| No: 352 |
| Statement: All game documentation shall be put online for the users to explore all aspects of the game and its design and implementation. |
| Source: Documentation |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Verify that all documentation is put online and is accessible. |
| Revision History: Created by Joel on 2/26/16 |

3.6 Data Requirements

* <Describe any data calculations: what formula will be used? to what degree of precision must the calculations be made? >
* <Describe any retained data requirements: exactly what must be retained?
* ...>

|  |
| --- |
| No: 361 |
| Statement: Entity Controllers shall store all relevant data to the object it is attached to. |
| Source: Implementing the Encapsulation and Abstraction principles of Object Oriented Design. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: UML Diagrams in High Level Design. |
| Evaluation Method: Verify all other classes only manipulate data, and Entity Controllers are the only classes that store data. |
| Revision History: Created by Greg on 02/25/16 |

|  |
| --- |
| No: 362 |
| Statement: Unity built-in functions shall be used when relevant. |
| Source: Minimizing development time. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: For each calculation necessary, the software engineer will review whether there is already a pre-existing function able to perform the calculation. If there is no pre-existing function, and the calculation is required multiple times, the developer will create a script for the calculation. |
| Revision History: Created by Greg on 02/25/16 |

3.7 Resource Requirements

|  |
| --- |
| No: 371 |
| Statement: The system shall require no more than three software engineers to complete the project. |
| Source: Group Size |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: |
| Evaluation Method: Ensure the work is evenly spread out between group members in order to ensure the individual workload does not increase to a level that compromises the project. |
| Revision History: Created by Greg on 02/25/16 |

|  |
| --- |
| No: 372 |
| Statement: The project shall be completed with Unity2D, coded in C#, and managed with Git version control. |
| Source: Proposed in CONOP. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: CONOP. |
| Evaluation Method: If any other software engineering tools or programming languages are required for this project, the overall scope and requirements will have to be reevaluated in order to determine the necessity for such a large change. |
| Revision History: Created by Greg on 02/25/16 |

|  |
| --- |
| No: 373 |
| Statement: The project, along with its required documentation and presentation brief, shall be completed no later than 11 April 2016. |
| Source: Course Requirement. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: COP 4331 Syllabus. |
| Evaluation Method: If at any point during the development process, the project does not appear to be completed by 11 April 2016, the overall scope and requirements will be reevaluated to determine the need to make changes, in order to ensure the product is delivered on time. |
| Revision History: Created by Greg on 02/25/16 |

|  |
| --- |
| No: 374 |
| Statement: The project shall be completed with only free software engineering tools. |
| Source: Developer Requirements. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: Ensure software engineers do not incur any financial cost during the course of the project. |
| Revision History: Created by Greg on 02/25/16 |

|  |
| --- |
| No: 375 |
| Statement: The overall size of the project, including necessary documentation, shall take no more than 1GB of memory. |
| Source: 1GB is the maximum allowable limit for a free public Github repository. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: Supporting Material Item No. 41 |
| Evaluation Method: If the overall project is approaching the 1GB limit, then documentation files can be moved to a different file hosting source. If the project approaches the limit again, then file size optimization of the project and reevaluation of scope and requirements will be necessary. |
| Revision History: Created by Greg on 02/25/16 |

3.8 Security Requirements

* <Describe any security requirements:
  + must access to the system or information be controlled?
  + must one user's data be isolated from others?
  + how will user programs be isolated from other programs and from the operating system?
  + how often will the system be backed up?
  + must the backup copies be stored at a different location?
  + should precautions be taken against fire, water damage, theft, ...?
  + what are the recovery requirements?
  + ...>

|  |
| --- |
| No: 381 |
| Statement: The system shall remain secure during development. |
| Source: Project Needs. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: The project is stored on a private Github repository, and all group members ensure that only the developers working on the project are allowed access. |
| Revision History: Created by Greg on 02/26/16 |

|  |
| --- |
| No: 382 |
| Statement: The system shall be stored and version controlled using Git. |
| Source: Project Needs. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: Due to the nature of how Git version control works, this ensures all group members keep a working copy of the project, backed up on a private Github repository. Group members commit changes and discuss merges whenever necessary. |
| Revision History: Created by Greg on 02/26/16 |

|  |
| --- |
| No: 383 |
| Statement: The system shall execute as a standalone application for Windows, OS X, and Linux devices. |
| Source: Usability concerns. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: To ensure the software will be safe from network attacks, once the game is downloaded it will run as a standalone application. There will be no network connectivity, and everything will be stored locally on the user’s computer. |
| Revision History: Created by Greg on 02/26/16 |

3.9 Quality Assurance Requirements

* <Describe quality attributes:
  + What are the requirements for reliability, availability, maintainability, security, portability ...?
  + How must these quality attributes be demonstrated?
  + Must the system detect and isolate faults? If so, what types of faults?
  + Is there a prescribed mean time between failures?
  + Is there a prescribed time the system must be available?
  + Is there a maximum time allowed for restarting the system after a failure?
  + What are the requirements for resource usage and response times?
  + ...>

|  |
| --- |
| No: 391 |
| Statement: The system shall be readable to all speakers of the English language. |
| Source: Most common language in the class. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: The project will be in English and the text will be easily readable and understandable. Play testers will determine the effectiveness of this requirement. |
| Revision History: Created by Greg on 02/26/16 |

|  |
| --- |
| No: 392 |
| Statement: The system shall minimize the required time to test. |
| Source: Common KPI (Key Performance Indicator) |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: None. |
| Evaluation Method: All testing will be timed, and the average testing time per component will be assessed. Outliers will require further re-evaluation for the component, possibly requiring a redesign. |
| Revision History: Created by Greg on 02/26/16 |

|  |
| --- |
| No: 393 |
| Statement: The user shall meet the minimum Unity System Requirements in order to play the video game. |
| Source: Recommended settings provided by Unity. |
| Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")> |
| Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")> |
| Supporting Materials: Supporting Material Item No. 41 |
| Evaluation Method: When downloading the application, users will be reminded of the minimum system requirements, and given a hyperlink to the Unity webpage that displays these requirements. |
| Revision History: Created by Greg on 02/26/16 |

**Section 4: Supporting Material**

* <Here is where you put all your analysis work from which you derived the above requirements. It may include UML or other diagrams, notes, memos, etc.)
* **Item No. 41:** [Github Repository File Restrictions](https://help.github.com/articles/what-is-my-disk-quota/)
* **Item No. 42:** [Unity System Requirements](https://unity3d.com/unity/system-requirements)