Table of Contents

[1. System Overview 2](#_Toc429140977)

[2. Project Development Management 3](#_Toc429140978)

[2.1. Organization and Resources 3](#_Toc429140979)

[2.2. Personnel 3](#_Toc429140980)

[3. Schedule and Milestones 4](#_Toc429140981)

[3.1. Scheduled Activities, Tasks, and Assignments 4](#_Toc429140982)

[3.2. Delivery Milestones and Baselines 4](#_Toc429140983)

[4. Risk Analysis 7](#_Toc429140984)

[5. Software Engineering 8](#_Toc429140985)

[5.1. Standards and Procedures 8](#_Toc429140986)

[5.2. Development Methodology 8](#_Toc429140987)

[5.3. Development Resources 9](#_Toc429140988)

[6. Testing Procedures 10](#_Toc429140989)

[7. Configuration Management 11](#_Toc429140990)

|  |  |
| --- | --- |
| **Team:** | Team F |
| **Team Members:** | Taylor Benner |
| **Date:** | 2015-08-31 |
| **Project Tile** | Simulated Evolutionary Environment |

# System Overview

This system will perform the task of simulating evolution on a localized scale. In order to do so, an environment will be created that will be populated with a varying amount of entities and obstacles. The entities will be composed of sensory input, mechanical output, simulated brain, and survival needs that must be fulfilled. The obstacles will consist of hazards and benefits such as toxins or food. The entities will breed, creating subsequent generations, subject to optimized selection and mutation.

The primary mechanisms of the entities mentioned herein are grouped into 3 primary collections. Input, Output, and Needs. Inputs will be composed of sensory input in the form of vision, food intake, and damage intake. The vision of an entity will be a conic ray that judges distance to an interception point. An additional layer of vision will translate the properties of the interception point. The output of the entity will be a hyperbolic tangent between -1 and 1 indicating the magnitude of motion on left and right tracks. The needs will be described as hunger and health.

The primary mechanisms of the environment will be limited to food and toxins at first. Food will be an interception point that decreases the hunger need. Toxins will be an interception point that decrease the health need. Additional environmental mechanisms will be added, time and scope permitting.

The purpose of this system is to generate a report on the progression of optimal weights and structure of a neural network given a problem domain. For the purpose of this project, the problem domain is described as survival in a simulated environment for an entity emulating basic biological mechanisms. The key metric of success of the project will be measured by the ability to produce a report from stored data indicating optimization over time.

In terms of team composition, since the team is only composed of one member, subsequent sections will completed to the best of my ability however concessions will need to be made for Personnel, Management, and Task allocation.

# Project Development Management

The project management process selected for this project will be in the traditional style. This is in order to determine the best course of action for the development and implementation phases given the limited resources available, as discussed in the next few sections. The initial specifications of the project will be outlined in the sections to come, and the team should strive to accomplish the key deliverables within the allocated timeframe to ensure the project is completed within budget and schedule. This is also in an effort to reduce the impact of change within the lifetime of the project to ensure dependencies remain fixed, and work can progress in a linear fashion.

## Organization and Resources

Since the personnel of the team is limited in scope, the work to be done will be broken down into primary phases of development focusing on individual development phases. These phases will be discussed more in the coming sections, but a cursory overview is as follows:

1. Development Environment
2. Rendering Engine
3. Creature Development using Neural Networks
4. Simple Simulated Environment
5. Genetics
6. Simulation and Reporting

The resources available are limited in scope to 4 hours development per weekday, and 6-8 hours development per weekend. The language used will be Python incorporating abstracted libraries to reduce the amount of new code to be written. This section will be expanded upon on in Section 5.

## Personnel

The personnel assigned to this team is composed of one member. This member will create the documentation, software, tests, and planning to the best of their ability within the lifetime of the project. Since the team is working on limited resource availability, a baseline of key features will be established in Section 3 that details the critical path of the project. The Work Breakdown Structure covered in Section 3.2 will identify the key features the system needs in order to accomplish the initial specifications as outlined in the System Overview, and to make concessions for features that are not along the critical path but may be added, time permitting.

# Schedule and Milestones

This project’s lifespan will be from Monday, August 31st 2015 to Saturday, October 24th 2015. The weekly allocation for this project is 8 Weeks. The primary milestones of the project are identified in Section 3.1 and assigned as Work Packages in Section 3.2.

## Scheduled Activities, Tasks, and Assignments

As indicated in the personnel section of this paper, all assignments are to be completed by the same individual. As such, the “Assigned To” portion of the Work Breakdown Structure in Section 3.2 will be removed. The milestones identified below will be assigned as work packages with specified duration and estimated delivery date via Section 3.2.

## Delivery Milestones and Baselines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Activity  Name** | **Description** | **Start Date** | **End Date** | **Dependencies** |
| 1 | Management | The ongoing processes and documents needed to manage the project | 8/31/2015 | 10/24/2015 | -- |
| 1.1 | Project Plan | The initial project document | 8/31/2015 | 9/6/2015 | -- |
| 1.2 | Team Charter | Detail responsibilities of team member | 9/6/2015 | 9/13/2015 | 1.1 |
| 1.3 | Requirements Specification | Documentation detailing business needs and schedule composition | 9/6/2015 | 9/13/2015 | 1.1, 1.2 |
| 1.4 | Design Specifications | Document detailing the design and specifications of the software | 9/13/2015 | 9/20/2015 | 1.3 |
| 1.5 | Test Plan | Create implementation documentation and user guides | 10/11/2015 | 10/18/2015 | 1.4, 2, 3 |
| 1.6 | Final Deliverable | Organize and revise documentation pertaining to project for delivery | 10/18/2015 | 10/24/2015 | 1, 2, 3 |
| 2 | Environment Setup | The initial environment set up | 9/6/2015 | 9/20/2015 | 1 |
| 2.1 | Development Environment | Select needed technologies based on requirements and create the development environment | 9/13/2015 | 9/20/2015 | -- |
| 2.2 | Library Installation | Acquire and Install software dependencies | 9/13/2015 | 9/20/2015 | -- |
| 2.3 | Code Repo | Create subversion repository for code on GitHub | 9/6/2015 | 9/13/2015 | -- |
| 2.4 | Development Environment | Install Sublime Text 2 and Python 2.7 | 9/6/2015 | 9/13/2015 | -- |
| 3 | Software Development | The creation of the software packages | 9/20/2015 | 10/11/2015 | 1, 2 |
| 3.1 | Rendering | Using PyGame to create a modular rendering system | 9/20/2015 | 9/24/2015 | - |
| 3.2 | Entity Development | Using PyBrain to create the entity class | 9/24/2015 | 9/27/2015 | 3.1 |
| 3.3 | Environment Development | Creating simple environment and data gathering | 9/27/2015 | 9/30/2015 | 3.2 |
| 3.4 | Genetics Implamen. | Implement genetic algorithm to optimize entity | 9/30/2015 | 10/06/2015 | 3.3 |
| 3.5 | Reporting | Create reports showing optimization over time | 10/06/2015 | 10/11/2015 | 3.4 |

# Risk Analysis

* Limited Personnel Available  
  Strategy – Mitigation  
    
  This is the project’s greatest risk. Due to the lack of team members, if the involved member falls behind schedule, the rest of the project’s quality or implicit budget may suffer. The method employed to deal with this is to ensure that scope is kept under control in terms of applicable features of the system and preemptive documentation efforts. This risk will require active mitigation efforts to ensure it does not pose a problem over the lifetime of the project. Risk mitigation will be monitored via the baseline established in Section 3.2 to ensure the project does not fall behind schedule
* Scope Creep  
  Strategy – Avoidance   
    
  Given the nature of this project, scope creep is a very real risk that needs to be avoided. In conjunction with limited personnel, feature bloat would cause the project to fall behind schedule and may contribute to not being able to complete. As such, the team must adhere to the baseline established in Section 3.2 and the specifications determined in the Design Plan to be completed at a later time. Preemptive efforts have already been made to ensure that the development is set up prior to the specified date of 9/6/2015. This ensures that any potential problems with environment are addressed early.
* Dependency Failure  
  Strategy – Mitigation  
    
  Since the project relies heavily on external dependencies, each one provides a single point of failure for the project. As such, in conjunction with the Scope Creep Mitigation Plan, the dependencies have already been tested on the development platform and ensured that they offer the features needed to complete the project. This is elaborated on in Section 5.

# Software Engineering

The following section details the methodologies, resources, and procedures the team will use in the production of the project system.

## Standards and Procedures

A cut down version of IEEE’s Process Tree development methodology will be used to ensure that the elements of the project are completed correctly and in an acceptable time frame. The project will be broken down into 3 subsequent branches containing Primary, Supporting, and Organizational structures.

The Primary Branch will focus on the development, deployment, and operation of the project program. All development will be completed under this classification and further documents will reference the Primary Branch when referring to the technical development.

The Supporting Branch will consist primarily of the documentation and configuration management portions of the project. Any potential issues that arise during the lifetime of the project such as conflict or additional risk assessment and management will be handled through this branch.

Finally, the Organizational Branch will house the management initiatives, process control and management, and user guide specifications.

## Development Methodology

The software development employed over this project will follow the IEC 12207 Standards of Development. Initially, a high level overview will be drafted that explores the specifications of the project and how this relates to the system. Modular design will be an important piece of the development lifecycle. The system will be broken down into the following coding classes:

* Rendering
* Entity
* Environment
* Genetics
* Reporting

The methods and members of each will be explored as each class is designed via the Design Specifications document. After the design plan has been created for the above modules / classes, coding will commence. Each class will have Unit Tests created for the primary routines that ensure the module works correctly. Finally, a systems test will be drafted to ensure the system meets the minimum requirements of the project specification.

## Development Resources

* Languages
  + Python
* Libraries and Dependencies
  + PyGame
  + PyBrain
  + Numpy
  + Shelve
* Software Resources
  + Sublime Text 2
  + Python 2.7
  + GitHub Online Repo
  + Microsoft Word
  + Microsoft Excel
  + Microsoft Project
* Online Information Resources
  + Stack Overflow
  + Python Documentation
  + Wikipedia
  + PyBrain.org
  + PyGame.org

# Testing Procedures

In order to assess the project’s health over time, weekly status reports and time sheets will be provided by the team leader. These weekly reports will indicate how the project is progressing over the lifespan, and if additional risks are identified along the critical path. Using the Work Breakdown Structure as a baseline metric, we will be able to monitor the project to ensure it remains on schedule and accomplishes the project requirements.

Software will routinely be tested for bugs in development. The design phase of the project will flesh out specifications for each individual module or class. These specifications will provide a more granular approach to testing the program as we can actively monitor for bugs in the code that prevent specification competition, as well as ensure that implementation with additional modules will happen in a way that produces the desired output.

Both static and dynamic testing method will be employed based on the deliverable in question for that period of time. Time permitted, documentation will be submitted for review prior to final submission to ensure all areas of the document are covered and are in line with project specification.

Dynamic Testing will involve creating test cases for a method or member of a class to ensure it performs the desired function. An example may be that if an entity’s’ coordinates intersect that of a toxin, that entity’s health must decrease proportionally to the strength of the toxin and the longevity of interception.

As each class is designed, these test cases will be created for the primary methods that class must perform.

# Configuration Management

Every day, changes to the code base and documentation base will be submitted to a GitHub Repository online. This is to provide versioning control over time for use if a version needs to be rolled back. The version number of the project will increment by 1 for each major revision which includes: documentation, class introduction, milestones, or weekly rollover. The secondary version number will increment for each commit to the repository which tracks change over version.

The versioning standard will be Project X.YY. As the project progresses the version number will be tracked along with the corresponding baseline package as determined in Section 3.2. This will provide a means of monitoring progress as a definable metric and correlate it to the actual development of the project over time.

In this way, the team will be able to track change over time in the form of definable change logs that may be used in future implementation phases. This also provides the ability for any additional team members that may be added in the future to follow how the project has progressed, as well as be able to routinely obtain the newest copy of the project documentation and code bases.

In congruence with section 3.2, the GitHub Repository will be created and linked to in the second week.