PrePARED BY: TEAM 21

Jacob Cerda, Enos Franceschini, Maxwell Heller, ThongPanh DuangBoudda

Project Plan

Project 21: Wildlife Refuge

September 30, 2018

Table of Contents

[Project Description 2](#_Toc493594714)

[Overview (CO-5): 2](#_Toc493594715)

[Key Requirements (CO-5): 2](#_Toc493594716)

[Deliverables (CO-5, CO-6): 2](#_Toc493594717)

[Acronyms and Abbreviations (CO-7): 2](#_Toc493594718)

[Design and Architecture (CO-1, CO-3) 3](#_Toc493594719)

[Implementation Strategy 3](#_Toc493594720)

[High-level Work Breakdown Structure (CO-2): 3](#_Toc493594721)

[Schedule / Timeline (CO-2): 3](#_Toc493594722)

[Required Hardware (CO-2): 3](#_Toc493594723)

[Third party content (CO-2): 3](#_Toc493594724)

[Quality (CO-2): 3](#_Toc493594725)

[Other Special considerations (CO-7, CO-3): 4](#_Toc493594726)

[Process 4](#_Toc493594727)

[Process Description and Justification (CO-2) 4](#_Toc493594728)

[Tools (CO-2): 4](#_Toc493594729)

[Roles and Responsibilities (CO-2): 4](#_Toc493594730)

[Location of Project Artifacts (CO-2): 4](#_Toc493594731)

[Sponsor Communications (CO-7): 4](#_Toc493594732)

[Risk management 4](#_Toc493594733)

[Identified Potential Risks (CO-2): 4](#_Toc493594734)

[Mitigation Strategies (CO-2, CO-3): 5](#_Toc493594735)

# Project Description

## Overview (CO-5):

The client’s application is a web simulation allowing users to populate a wildlife refuge and to let them observe and interact with the result. The application will demonstrate to users what animals do in their free time using artificial intelligence. An HTML5 web application that integrates the SmartSparrow platform will be provided. The client (ETX) wishes to expand their lists of authors and offerings of digital coursework by creating new simulations outside their current content domains. ETX is also aiming to teach students through simulations. The users will include members of the ETX Center (for testing and suggestions), and visitors of the Infiniscope website when the app has reached completion.

## Key Requirements (CO-5):

The platform requirement is SmartSparrow, an application to create interactive lessons. For coding and graphics, we will invest in WebGL and WebAssembly. HTML5 is also an essential part of the application and some team members already have experience with this language. The members with no web programming experience will focus on the graphical and non-web programming in WebAssembly. Some functional requirements include artificial intelligence-based animal behavior to control the animals that are the key features of the application as well as the environmental landscape and location of the refuge.

## Deliverables (CO-5, CO-6):

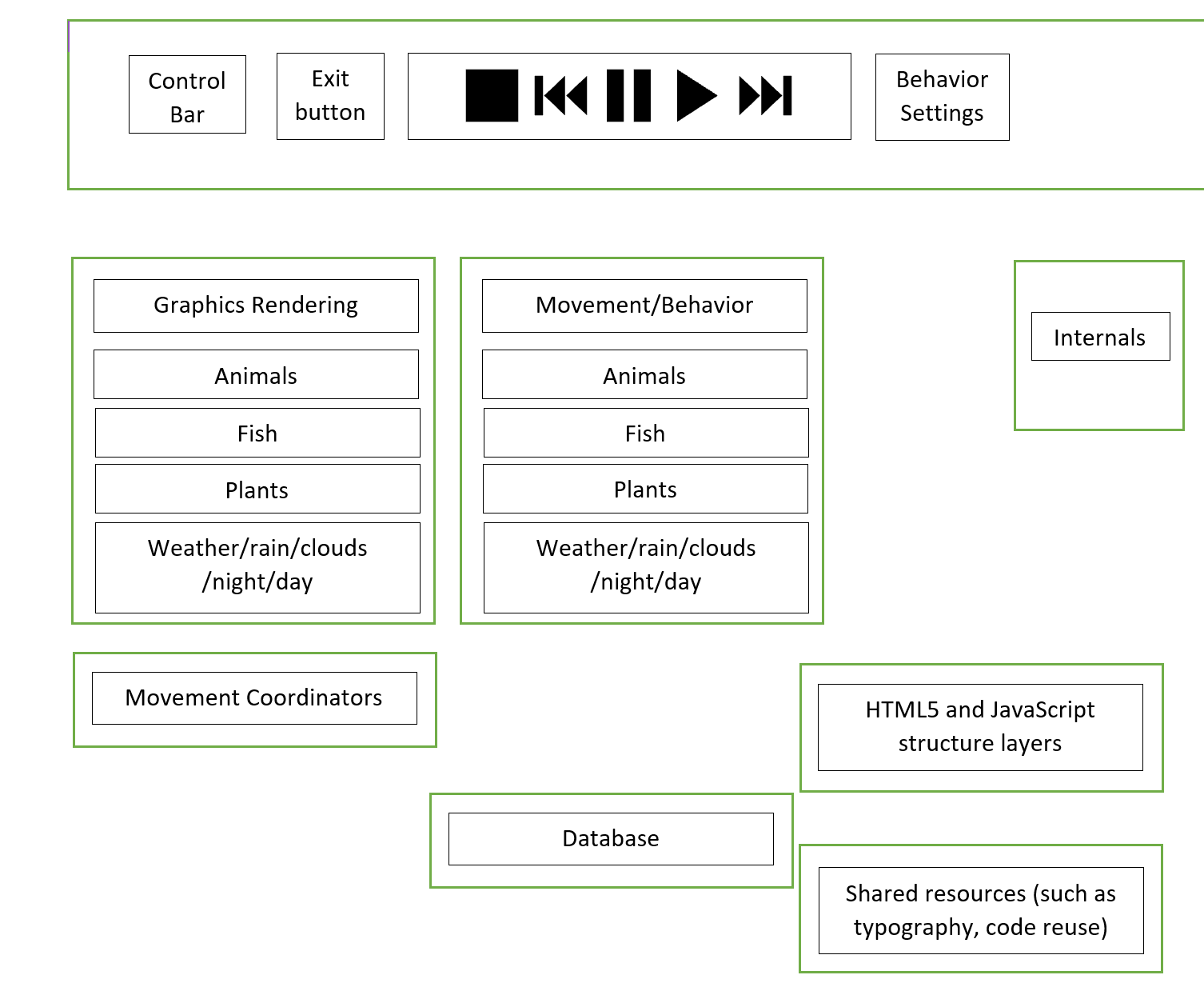
The team will provide an SRS document detailing all the software requirements along with an MTP document detailing the testing requirement to verify proper functionality of the software. Once development commences all functional source code will be provided with respective unit tests. The source code will provide functionality for the simulation of a wildlife refuge. The simulation will encompass the behaviors of animals and environment in the refuge. The behaviors will cover activities such as hunting, exploring, foraging, hibernating, interacting with other animals, and interacting with surrounding environment. The simulation will also have environment characteristics specific to the chosen refuge such as local flora, local climate, etc. Finally, the complete project deliverable will be a static web page, deployed to basic hosting, that will be functional on Microsoft IE, Microsoft Edge, Google Chrome, and Mozilla Firefox.

## Acronyms and abbreviations (CO-7):

|  |  |
| --- | --- |
| Acronyms/Abbreviations | Definition |
| ETX | Education Through eXploration |
| SPR | SmartSparrow |
| WR | Wildlife Refuge |
| SRS | Software Requirement Specification |
| MTP | Master Test Plan |
| IE | Internet Explorer |
| FF | Firefox |
| GC | Google Chrome |
| ME | Microsoft Edge |

# Design and Architecture (CO-1, CO-3)

Architecture Diagram:



The architecture will require the modules shown in the above diagram. There will be a ribbon with placement TBD, that will allow users to control the animation, add animals and other features like wooded areas or swamp area. The ability to modify the behavior will also be included in that ribbon.

The structure will be a combination of HTML5 and JavaScript.

Using movement coordinators to model interaction, the goal is to make the movement and behavior of each animal lifelike.

The database will hold all wildlife and botany, as well as the overall landscape and behavior models.

Implementation Strategy

## High-level Work Breakdown Structure (CO-2):

1. SmartSparrow Learning Platform:

The SmartSparrow learning platform makes it possible to deploy immersive and interactive web-based applications; also, it provides tools to analyze user behaviors, so that developed applications are improved. However, the team does not have prior experience deploying applications on the platform. To successfully deploy the products on the platform, some time is expected to allow the team familiarization with structures and tools.

2. HTML5:

The team members have different levels of knowledge regarding HTML5, and some are beginners. For a beginner to HTML5, it will take some time to familiarize ourselves with syntax, such as tags, attributes, colors, and others will be required.   
HTML5 will be used mainly in the design of the web site.

3. JavaScript:

JavaScript poses the same challenges as HTML5; some members have experience with JavaScript whereas others are beginners. However, JavaScript and HTML5 are logically similar. If the requirements and flow are well defined, then computation should be fairly straight forward. Nonetheless, some time is expected to become familiar with the syntax; such as, variable declaration, variable types, functions or methods, objects and others.

JavaScript will be used to control the web site’s behaviors. For instance, when users populate the Wildlife Refuge with animals or plants, JavaScript shall provide appropriate behaviors for the web pages to display, according to the data and intelligence in the model.

4. Data and Intelligence:

The team members started conducting, researching and gathering data and intelligence immediately after the first sponsor meeting. Communications have begun with a subject matter expert who is a professor at ASU, and extensive web research is ongoing to collect data about animals, plants and environments. Moreover, the sponsor at ETX has volumes of data on non-related sample simulations. Nonetheless, there will be continued research, data and intelligence gathering as we develop our application so that the application can be improved incrementally. Data collection activities will be continued until the completion of the project.

5. WebGL (Web Graphics Library)

WebGL is a JavaScript API for rendering interactive 2D and 3D graphics within modern web browsers; such as Chrome, Microsoft Edge, and Firefox, without the use of plug-ins. Therefore, WebGL is deemed necessary to provide immersive and interactive applications as envisioned in SmartSparrow. The team is still new to WebGL; however, there are resources available the team is confident that this hinderance will be overcome. WebGL use extensive JavaScript APIs, and it will take the team a while to familiarize with these APIs. Some time will be necessary to learn the basics of WebGL.

## Schedule / Timeline (CO-2):

The schedule for this project will fit into the two-semester timeline required by the Capstone Course. The team will follow the cadence of 3-week sprints. The current schedule covers the dates for the whole semester, however the milestones are only listed for this semester only. The milestones that are planned for development are listed below:

|  |  |
| --- | --- |
| Define Project Plan | 10/1 - 10/5 |
| Sprint 1  Gather refuge environment information  Gather refuge animal information  Meet with subject matter expert (dependent on availability)  Identify required refuge environment, animal, objects.  Design and Define elements for web site.  Preliminary Design and Presentations | 10/8 - 10/26 |
| Sprint 2  Finalize website layout and elements  Create initial refuge environment objects. | 10/29 - 11/16 |
| Sprint 3  Provide Project Status Report, Review Code and Demo | 11/19 - 12/7 |
| Sprint 4 (Tentative for Winter Break)  Complete unfinished features from previous sprints  Bug fix | 12/10 - 01/4 |
| Sprint 5 | 01/7 - 01/25 |
| Sprint 6 | 01/28 - 02/15 |
| Sprint 7 | 02/18 - 03/08 |
| Sprint 8 | 03/11 - 03/29 |
| Sprint 9 | 04/1 - 04/19 |

## Required Hardware (CO-2):

Any personal computers: Laptops and desktops with modern web browsers; such as, Firefox, Microsoft Edge, and Chrome can be used as developing tools. For testing tools, cell phones, tablets, laptops, and desktops with modern web browsers can be used. All tools that are used in the project are not supplied by any parties; Our team is expected to have personal computers.

## THird party content (CO-2):

ETX will provide Javascript API packages that must be integrated into the final delivery; also, source code, advisement, and resources will be provided by our sponsor to assist us in completing the necessary tasks, according to the project description. Moreover, ETX will host all final web deliverables. Therefore, if there are additional contents we need to complete our project, we will use only content that are available for public.

## Quality (CO-2):

Quality of the product will be measured by validating functionality of the final product. The validation coverage will be defined in the MTP. The MTP will have tests for all the respective features that are developed as well as tests to ensure that requirements are met. The MTP will have functional, non-functional, performance, and integration tests to ensure a pleasant user experience. The metric will be provided as a percentage of implemented and validated features/requirements vs the total number of features/requirements. Another metric for measuring the quality of the product will be the number of bugs that are filed against the product which will be tracked and evaluated at each sprint. Finally, all commits will be integration tested with SPR.

To ensure readable and reusable source code the following style guides and recommendations:

* + HTML
    - [W3Schools HTML5 Style Guide](https://www.w3schools.com/htmL/html5_syntax.asp)
  + JavaScript
    - [W3Schools JavaScript Style Guide](https://www.w3schools.com/js/js_conventions.asp)
    - [W3Schools JavaScript Best Practices](https://www.w3schools.com/js/js_best_practices.asp)
    - [W3Schools JavaScript Performance](https://www.w3schools.com/js/js_performance.asp)
    - [W3Schools JavaScript Mistakes](https://www.w3schools.com/js/js_mistakes.asp)
  + CSS
    - [Google CSS Style Guide](https://google.github.io/styleguide/htmlcssguide.html#CSS)

To further ensure quality tasks will be assigned to respective team members, tracked through Trello, and each commit that is made will be tied to a Task ID. Commits will not be merged unless reviewed and approved by at least one non-submitting team member.

## Other Special considerations (CO-7, Co-3):

To a team member who is not familiar with HTML5 and JAVASCRIPT, there is a tutorial ([HTML5 Tutorial](https://www.w3schools.com/html/default.asp)) that provides most of what they need to know about HTML5 and JAVASCRIPT. Also, we will probably incorporate 3D functionality into our project, and 3D is new to us. We found these sites that we believe are good references:

1. [Rachel Smith’s WebGL page](https://codepen.io/rachsmith/post/beginning-with-3d-webgl-pt-1-the-scene)

2. [Udacity Course on 3D graphics](https://www.udacity.com/course/interactive-3d-graphics--cs291%20/)

# process

## Process Description and justification (Co-2)

The team will use the Scrum process for this project. We do not plan on adapting the process itself but are not using Taiga for our project management. Trello will be used instead, due to the cleaner interface. As discussed in the Tools section, the Kanban boards are similar to the Scrum management boards. Scrum was chosen because it allows for fewer mistakes, it provides the framework for constant feedback and exposure to make sure the product is of the highest quality possible, and because of how it increases team morale. All client requirements will be treated with equal priority. All the requirements will be input and tracked in Trello.

## Tools (CO-2):

**Office 365:** Used to generate documentations; such as meeting minutes, project plan and other documents that are required. This was preferred, because all the members can read and edit the same version of documents and eliminates transferring documents through email.

**Slack:** Used for communication rather than email. This is preferred, because of its ease of use, and real time communication. Additionally, the Slack conversation history allows members to review past conversations to remain in sync with the team. More importantly, the need to search e-mail to retrieve past conversations is eliminated.

**Github:** This was selected because it is a widely used and well supported tool for version control and collaboration. Version control provides each member to have an individual version of the sources which can be worked on without breaking another member’s code. Each member can check out any branch of code to work on, then submit it, which allows other members to checkout and test it for approval or fixing bugs. Once reviewed and approved the branches can all be merged to the master branch.

**Trello**: This will be used for project and task management. Trello was selected for process/task management as the team has more experience with the tool. While Trello is not a full Scrum tool it allows the use of a Kanban board which is very similar to the Scrum task management. In addition to the built-in functionality that Trello provides there are plugins for Scrum functionality that can be used in the project.

Trello enables the team to clearly define tasks and assign them to members and track the progress of each task. Additionally, Trello can be integrated into GitHub so that pull requests, issues, and commits are easily tracked.

## Roles and Responsibilities (CO-2):

All members will rotate through the roles of scrum masters, team and project managers. The team is required to define software requirements, customer requirements and user stories for the project. Each member will select the requirements and user stories to work on each sprint. The scrum master role will rotate every sprint and is responsible for leading stand up, capturing and submitting meeting minutes. The scrum master is also responsible for monitoring team progress using burndown charts. Finally, the scrum master helps unblock team members.

## Location of Project Artifacts (CO-2):

GitHub will be used for source code artifacts to enable proper collaboration while minimizing merge conflicts with other commits. Documents will be held in Office365/Sharepoint/OneDrive to enable real-time document collaboration. Trello will be used for project management.

|  |  |
| --- | --- |
| Github | [WildLife Refuge Github page](https://github.com/jcerda94/group21WildLifeRefuge) |
| Office 365 | [Sharepoint site](http://arizonastateu-my.sharepoint.com/) |
| Trello | [Trello Group Page](https://trello.com/ser401group21/home) |

## Sponsor communications (CO-7):

Sponsor communications are planned to be held on a weekly cadence (dependent on sponsor and team availability). Additional meetings are planned to review each sprint deliverable. Meetings will be held on Slack and meetings will be taken by team members on a rotating cadence for the role.

# Risk management

## identified Potential risks (CO-2):

1. Difficulty bringing up tools and existing frameworks (Risk Level: Low - Med): This team has experience with most of the tools and frameworks that will be used in the project, though the team has not had experience with this scale and scope of project. Managing this level of complexity over a long period of time is a new experience for the team.
   1. Impact: A decrease in the quality of our documentation and a reduced pace of progress on the project.
   2. Expected incident rate: Every other Sprint (beginning several sprints in)
2. Team Skill Distribution (Risk Level: Med): Properly utilizing team members and equitably sharing work could be difficult. Some team members have significantly more skill in certain areas vs. others. This creates the risk of both overworking some team members and underutilizing others based on the project’s technical makeup/design.
   1. Impact: Code quality could drop because of overworked team members or improperly allocated work based on each member’s areas of expertise.
   2. Expected Incident rate: Up to every Sprint based on the Project’s design
3. Cadence and Regularity of Communication (Risk Level: Low): Our team members are not co-located and live in several different time zones. Maintaining timely communication and coordination could be difficult.
   1. Impact: Project timelines could slip, and delays could be created in our team’s response to unexpected issues (technical challenges, team members having personal emergencies, etc.)
   2. Expected Incident rate: Weekly if a cadence is not maintained

## mitigation strategies (CO-2, Co-3):

Risk #1: To mitigate this risk the team can integrate a couple different quality standards into our sprints and sprint retrospectives. If a user story is active across more than one sprint, the team will document the root cause and create a basic mitigation plan to prevent that story from creeping into future Sprints.

Quality Standards/Metrics to track:

* + Each user story has an associated requirement or an explicit justification for not having one
  + How many user stories are later decomposed into more stories (across all Sprints)
  + Measuring velocity over time (user stories per sprint)
  + Incorporating backlog grooming into each sprint retrospective

Tracking these metrics and the overhead added to each sprint retrospective should only add 30mins – 1hr per sprint.

Risk #2: For this risk the team can ensure that during the initial design phase that the design is properly modularized. Skill distribution and project difficulty can be discussed during sprint planning and sprint retrospectives. During feature creation and project development, team skill development can be assigned as a “user story”. With this development user story, an acceptance criteria can be set of how well that skill should be understood, and maybe add a criterion for a certain number of hours of pair programming/code review with a teammate more skilled in that area. The initial overhead for this strategy is high, the team would need to be very thorough in our initial design phase and have an extensive conversation about team skill levels around said design. Despite this initial overhead, preemptively addressing team skill distribution will most likely improve the consistency of sprint velocity later in the project’s development.

Risk #3: Mitigating this risk is simple. The team will establish a weekly team meeting in advance of the weekly sponsor meeting, so any potential issues can be addressed before meeting with the sponsor. To maintain a certain level of responsiveness, team members should check the team slack channel at least once per day and respond to any team inquiries during that time. Daily meetings or semi-weekly meetings can be established based on the needs of the Team. Team members will be required to give 24 – 48hrs notice if they know they will miss a team/sponsor meeting and write a short report of what they would have discussed in the meeting. There should be no active overhead for this communications policy and this policy can be amended to meet the team’s need over the lifecycle of the project.