Capstone Project Plan

Project Name: Virtual Pet Simulator

Project Members: Wiktoria Biernat (W0427844)
Cassandra Curtis (W0449188)
Josh Jones (W0446531)
Mark Moulton (W0440932)
Rose Scoville (W0249159)

Date: February 16, 2022

Content

Document Control

Document Authors

- Wiktoria Biernat
- Cassandra Curtis
- Josh Jones
- Mark Moulton
- Rose Scoville

Document History

- February 3, 2022 – Rough template created.
- February 9-10, 2022 – Draft completed.
- February 13, 2022 – Draft revised.
- February 16, 2022 – Document finalized.

Document Approval

MM, WB, CC, RS, JJ

Project Charter

Executive Summary

The purpose of this project is to work as a team to produce a Virtual Pet Simulator, as per Capstone project requirements, by the end of the Winter 2022 semester. The project will involve several different technologies including, but not limited to, Unity as an IDE, MySQL for database scripting, and Laravel for front-end web development. Project management technologies will include Git for source control, Bitbucket for version control, and Jira for project management and reporting.

The project is broken down into a series of 7 deliverables concluding on April 14, 2022. The deliverables are focused on the core aspects of the virtual pet, including lifespan, hunger, hygiene, happiness, sickness, and fussiness. Documents will be created outlining a hypothetical second phase involving a web-based user database for user battles. The project is designed to meet a series of high-level user objectives based on our position as graduating students at NSCC (Nova Scotia Community College).

We will be using Kanban as a project management methodology, as well as a series of SAAD components, including user flow documents and a dynamic project Gantt chart. We have also identified 9 key roles to be filled by members of the team at various stages of the project. Plans have also been made to meet QA standards.

Project Definition

Vision

To develop a Unity-based application for a virtual pet simulator by the end of the Winter 2022 semester, according to Capstone project requirements which will give users the opportunity to hatch and raise a virtual pet.

Objectives

1. To participate in and cooperate with a small team to manage a project and produce a final product
2. To apply and combine prior coursework knowledge (from both coding and non-coding related courses) to a larger project
3. To practice professional and effective communication between both team members and clients
4. To complete a project based on milestones, phases, and deliverables, according to a designated project management methodology
5. To produce a product worthy of a professional portfolio

Project Organization

Roles & Responsibilities

- **Project Manager**: Organizes the project and ensures all teams have what they need to get the job done. Helps facilitate communication between teams when needed.
- **Lead Game Designer**: Writes and updates the game design document describing gameplay in non-technical terms. Takes the disparate ideas from design team and brings them together into one cohesive vision.
- **Lead Developer**: Helps guide development team and provides resources, instruction, and demonstrations for working with Unity.
- **Lead UI/UX Developer**: Creates an appealing UI/UX to interact with game functionality.
- Lead Database Developer: Creates the database and integrates its functionality into the standalone windows build, and later the WebGL build.
- **Lead Web Developer**: Creates a website that displays the WebGL build and database data.
- **Design Team**: Works with the Lead Game Designer and Lead Visual Designer to create the look and feel of the product.
- **Development Team**: Works with the Lead Developer, Lead Database Developer, and Lead Web Developer to write the code that gives the product its functionality described in the game design document.
- **Quality Assurance Manager**: Maintains the QA Plan, Change Log, Library of SAAD documentation and Library of QA specific documentation.

Project Plan

Approach

1. A Kanban board to visually indicate tasks in the backlog, in progress, in review, and completed (see image “Project Board, Feb. 16”, below, for a current snapshot of this process in action). The entire team is responsible for the board as there is no “Scrum master” or any roles in Kanban.
2. Kanban’s ongoing flow, as opposed to Scrum’s fixed length sprints, affords us the flexibility to work around our diverse and sometimes unpredictable schedules.
3. Work-in-progress limits will assist us in identifying bottlenecks before they can become a larger problem.

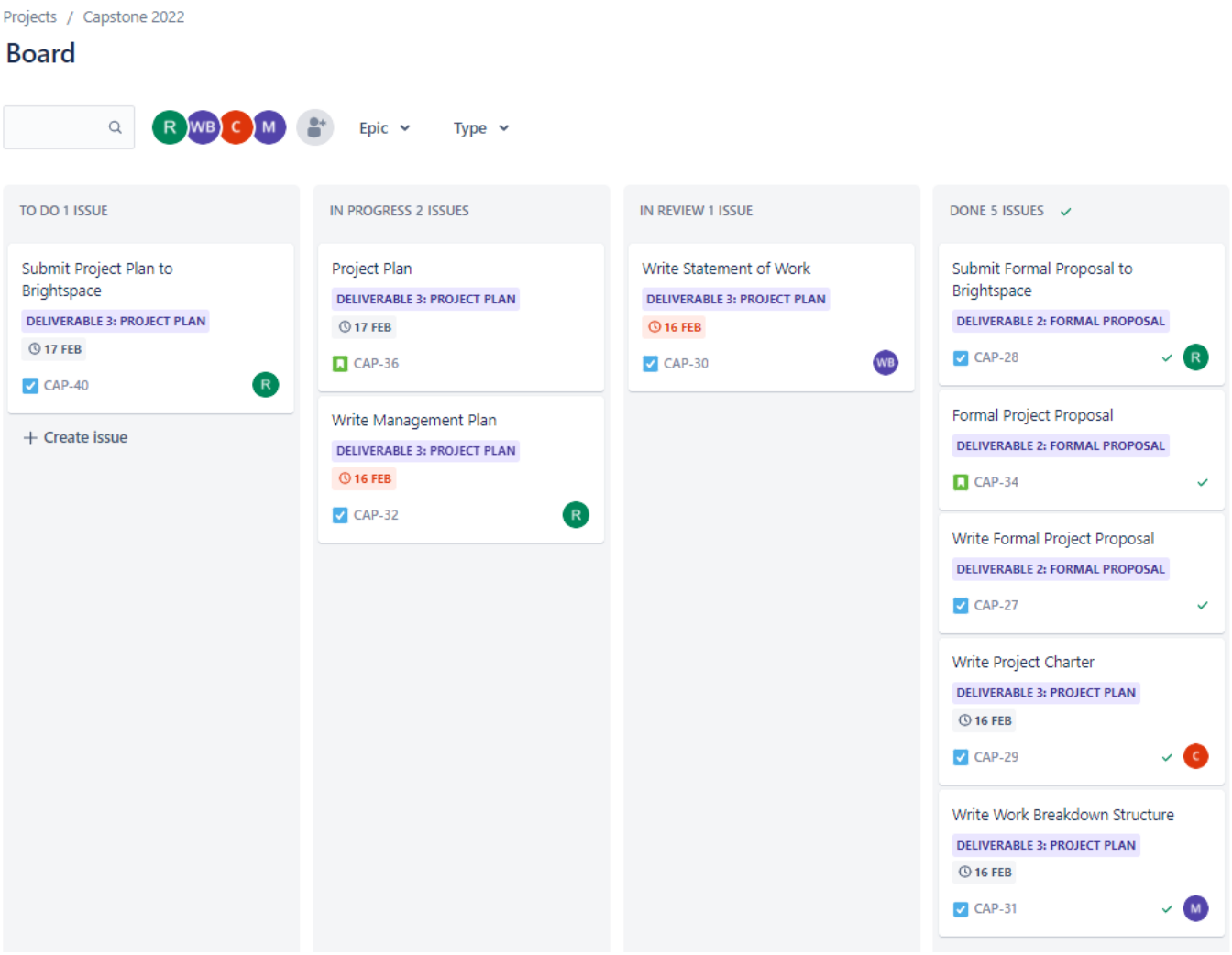


Image: Project Board, Feb. 16

Milestones

See Statement of Work: Milestones (page 11).

Quality Assurance

See Management: Quality Management (page 19).

Project Considerations

Risks and Mitigations

1. Risk 1: Intellectual property (IP) and trademark infringement
    a. Mitigation A: Avoid all trademarked IP (e.g., Toho's Kaiju properties, Tamagotchi-owned terminology).

    b. Mitigation B: Application is not to be published with any IP not owned by the team.
2. Risk 2: External work, life, and commitments overload
    a. Mitigation A: Team members are to try to be self-aware and communicate early to the rest of the team when life events, work overload, or burn-out might become an issue. The team will then re-delegate tasks appropriately and/or adjust project requirements as needed to meet deadlines.
3. Risk 3: Covid 19
    a. Mitigation A: Team members will adhere to all publicly required health and safety policies.
4. Risk 4: Technical difficulties due to unforeseen circumstances (e.g., storm-related power outages)
    a. Mitigation A: In the event of postponed work due to unforeseen events, the team will adjust project requirements as needed to meet deadlines.

Assumptions

1. The team will have five contributing team members throughout the project's duration.
2. Team members will be responsible for and complete all self-assigned/accepted work.
3. If any work is too much for a team member, they will communicate this early.
4. Team members will be communicative during agreed-upon meeting times (i.e., Capstone class hours and additional pre-set meetings if needed).

Constraints

1. Time constraints: the project must be finished by the end of the Winter 2022 semester
2. Intellectual constraints: all team members are second-year Programming students with limited programming experience
3. Physical constraints: all team members are working with their personal tech from their home offices
4. Mental constraints: all team members are also completing several other courses and have commitments and responsibilities beyond the completion of the project

Statement of Work

Note: See Project Roadmap and Gantt chart for a detailed scope breakdown.

Scope of Work

TO BE DONE	RESOURCES	OUTCOME	TIME	GENERAL STEPS
Development Milestone 1	• Git • Bitbucket • Jira • Discord • InVision • Visual Paradigm • Unity • MySQL • Laravel • Aseprite • Microsoft's Internet Information Services	• A basic virtual pet • Database and website are set up but empty	18/02/2022 - 03/03/2022	• Create a game design document to iron out detailed mechanics. • Create ERD (Entity Relationship Diagram) and user flows to plan design. • Use Git/Bitbucket for source control. • Create a virtual pet application in Unity, using art assets created in Aseprite. • Create database with MySQL. • Create website with Laravel.
Development Milestone 2	• Git • Bitbucket • Jira • Discord • InVision • Visual Paradigm • Unity • MySQL • Laravel • Aseprite • Microsoft's Internet Information Services	• Fully featured virtual pet • Database saves high scores from minigame, and the website displays them	04/03/2022 - 24/03/2022	• Create minigame for virtual pet in Unity. • Link minigame high score to database and database to website.

Location

All project work will be done remotely at team members' individual work sites, with communication and resource sharing on digital platforms (see Reporting and Communications, page #12), for the full duration of the project.

Schedule

Note: See Project Roadmap for complete Gantt chart. See Project Board for task details, including the team member responsible and steps to completion.

Phases	Start	Finish
Phase 1: Initial Proposal (pre-production)	12/01/2022	24/01/2022
Phase 2: Formal Proposal (pre-production)	25/01/2022	03/02/2022
Phase 3: Project Planning (pre-production)	04/02/2022	17/02/2022
Phase 4: Development Milestone 1	18/02/2022	03/03/2022
Phase 5: Development Milestone 2	04/03/2022	24/03/2022
Phase 6: Presentation (post-production)	25/03/2022	07/04/2022
Phase 7: Review and Reflection (post-production)	08/04/2022	14/04/2022

Deliverables

Deliverable	Objective	Due date
Initial Proposal	Define the internal and external sides of the project.	20/01/2022
Formal Proposal	Outline how the project will be done.	03/02/2022
Project Plan	Detail how the project will be done.	17/02/2022
Milestone 1 Report	Report updates on timeline and work.	03/03/2022
Milestone 2 Report	Report updates on timeline and work.	24/03/2022
Final Presentation	Present completed product and project's process.	07/04/2022

Review and Reflections	Review project successes, lessons learned, and opportunities for future improvement, with a focus on how the team worked together.	14/04/2022

Milestones

Milestones	Start date	Finish date
Phase 1 completion	January 12, 2022	January 24, 2022
Phase 2 completion	January 25, 2022	February 3, 2022
Phase 3 completion	February 4, 202	February 17, 2022
Phase 4 completion	February 18, 2022	March 3, 2022
Phase 5 completion	March 4, 2022	March 24, 2022
Phase 6 completion	March 25, 2022	April 7, 2022
Phase 7 completion	April 8, 2022	April 14, 2022

Tasks

Task #	Activity	Due	Responsible member(s)
1	Set up a team communication channel for ongoing discussions and to collect documentation links.	Jan. 20	Mark Moulton
2	Project Manager to explore and set up project management tool(s) to organize and visualize project process and status for its full duration.	Jan. 20	Rose Scoville
3	Project Manager to explore and set up team's version control for development phase, link this to project management tool if possible.	Jan. 20	Rose Scoville
4	Collect brainstorming notes and team members' input on own strengths/interests and weaknesses/disinterests.	Jan. 20	Cassandra Curtis
5	Establish each team member's role.	Jan. 20	All team members
6	Project Manager to add deliverables, objectives, and tasks to project board and roadmap.	Jan. 20	Rose Scoville
7	Define the minimum viable product to be developed, including feature list and requirements.	Jan. 20	All team members
8	Dev Team to explore technology options and define project's stack.	Jan. 20	All dev team members
9	Project Manager to submit Deliverable 1: Project Goal Statement and Objectives doc to Client.	Jan. 20	Rose Scoville

10	Team to write formal Project Proposal.	Feb. 3	All team members
11	Project Manager to submit Deliverable 2: Formal Proposal to Client.	Feb. 3	Rose Scoville
12	Define Milestone 1 requirements to be developed.	Feb. 17	All team members
13	Define Milestone 2 requirements to be developed.	Feb. 17	All team members
14	Define product user flow(s) to account for all minimum viable product requirements.	Feb. 17	Rose Scoville
15	Create minimum viable product wireframes for application pages.	Feb. 17	Cassandra Curtis
16	Create minimum viable product page mock-ups and prototype to show application flow.	Feb. 17	Cassandra Curtis
17	Team to write Project Plan.	Feb. 17	All team members
18	Project Manager to submit Deliverable 3: Project Plan.	Feb. 17	Rose Scoville
19	Dev team to research and familiarize with stack (Unity, WebGL, etc.).	Feb. 17	All dev team members
20	Code basic virtual pet v0.01 with placeholder art. All stats should be tracked, but no mini game at this point.	Feb. 24	All dev team members
21	Use v0.01 to design and hook up UI.	Feb. 28	Cassandra Curtis
22	Create a MySQL database.	Feb. 28	Wiktoria Biernat
23	Set up environment for website development.	Feb. 28	Josh Jones
24	Install and create deployment environment for website.	Feb. 28	Josh Jones
25	Create test database for the site during development (phpMyAdmin).	Feb. 28	Josh Jones
26	PM to submit Deliverable 4: Milestone 1 Report.	Mar. 3	Rose Scoville
27	Code simple minigame to meet requirements defined in Project Plan.	Mar. 14	Wiktoria Biernat, Rose Scoville
28	Hook up minigame score to database leaderboard.	Mar. 20	Wiktoria Biernat, Rose Scoville, Josh Jones
29	Setup basic functionality for communication between site and database.	Mar. 23	Josh Jones, Wiktoria Biernat
30	Establish connection to game through website.	Mar. 23	Josh Jones, Mark Moulton, Wiktoria Biernat
31	Deploy WebGL version to website for testing.	Mar. 23	Josh Jones
32	PM to submit Deliverable 5: Milestone 2 Report.	Mar. 24	Rose Scoville
33	Create Project Presentation.	Apr. 5	Cassandra Curtis, Rose Scoville
34	Team to present project.	Apr. 7	All team members
35	Project Manager to submit Deliverable 6: Project Presentation.	Apr. 7	Rose Scoville
36	Have a project post-mortem discussion.	Apr. 12	All team members
37	Team to complete Project Reflections document.	Apr. 14	All team members

38	Submit Deliverable 7: Project Reflections document.	Apr. 14	All team members
39	Team to complete and submit Peer Review Documents.	Apr. 14	All team members

Reporting and Communications

Project Updates and Reports

- Weekly project updates (#weekly-updates, Discord)
    - Description: The project manager will post weekly updates outlining the project's timeline progress, project board status, and tasks for the week ahead. These will be posted to the #weekly-updates channel on the team's Discord server.
    - Occurrence: Mondays
- Project Board (Jira)
    - Description: Overview of all tasks belonging to the current project phase. The Board displays tasks statuses within 'to do,' 'in progress,' 'in review,' and 'done' columns. Task details, including due date, assignee(s), descriptions, reference links, and associated development branches, can be accessed by clicking on the task within the board.
    - Occurrence: Created in the project's preproduction phase and updated throughout the project's duration.
- Project Backlog (Jira)
    - Description: List of all tasks belonging to future phases. Tasks will be moved from the Backlog to the Board when their respective phases begin.
    - Occurrence: Created in the project's preproduction phase and updated throughout the project's duration.
- Project Roadmap (Jira)
    - Description: Overview of project's timeline, phases, and current progress within overall timeline
    - Occurrence: Created in the project's preproduction phase and updated throughout the project's duration.

Team Communications

- Team Capstone Server (Discord): general teamwide discussions
    - #design: product design discussion
    - #dev: product development discussion
    - #documents: ongoing discussions around documentation
    - #resources: project resources shared
    - #team-norms-and-expectations: discussions on self-management and mutual expectations
- Task changes and updates (Jira):
    - Documented within task tickets on Project Board and Backlog
- Code commits and pull requests reviews and discussions (Bitbucket, Jira):
    - Documented within Commits and Pull Request comment threads.
- Bug tracking and reporting (Bitbucket)

- Documented on Project Board and Backlog as 'bug' category tickets.

Meetings

- Meeting minutes (Discord): documented by all members on a new channel per meeting.
- Meeting archives (Discord): past meeting minutes are accessible within the Meetings Archive.

Documentation Library and Archive

A library of documents will be maintained by the Project Manager and Quality Assurance Manager. Documents can be accessed in their respective categories as follows:

- Final documentation and project deliverables (Discord)
- SAAD documentation (Discord)
- QA documentation (Discord)
- Programmers Reference Manual (Discord)
- Change Log (Discord)

Standards and Testing

STANDARD/TEST	TEAM MEMBER	DUE DATE
Standards to be followed will be detailed in the Programmers Reference Manual.	All	Ongoing
Tests will be ongoing, new features are to be tested prior to pull requests being accepted.	Unity: Mark Moulton Database: Wiktoria Biernat Web: Josh Jones	Ongoing

Definition of Success

As this project is intended to be a cooperative educational experience, success will be measured by the following:

1. We effectively manage the project within a team
2. We complete all Capstone project deliverables with a satisfactory grade
3. We identify constructive learning experiences

The above will define our success more than whether we meet all product deliverables. This means that the project may still be deemed a success if we do not complete our intended vision of the working virtual pet, so long as we effectively self-managed, and an effort was put in by the team to solve whatever technical and interpersonal issues arose as we worked toward our product vision.

Requirements

Product Related Technologies

1. Unity (IDE)
2. C# (back-end/application language)
3. WebGL (application-web migration)
4. MySQL (database)
5. PHP (front-end/web language)
6. Laravel (front-end/web language)
7. CSS (front-end styling)
8. Aseprite (asset creation)
9. Microsoft's Internet Information Services (HTTP hosting)

Project Management Related Technologies

1. Git (source control)
2. Bitbucket (version control)
3. Jira (project management and reporting)
4. Discord (team communication)
5. Microsoft Teams (client communication)
6. InVision (wireframing and mock-ups)
7. Visual Paradigm (ERD)

Hardware

Each team member will require a computer that can handle the required IDEs and software to complete the project tasks.

Knowledge and Training

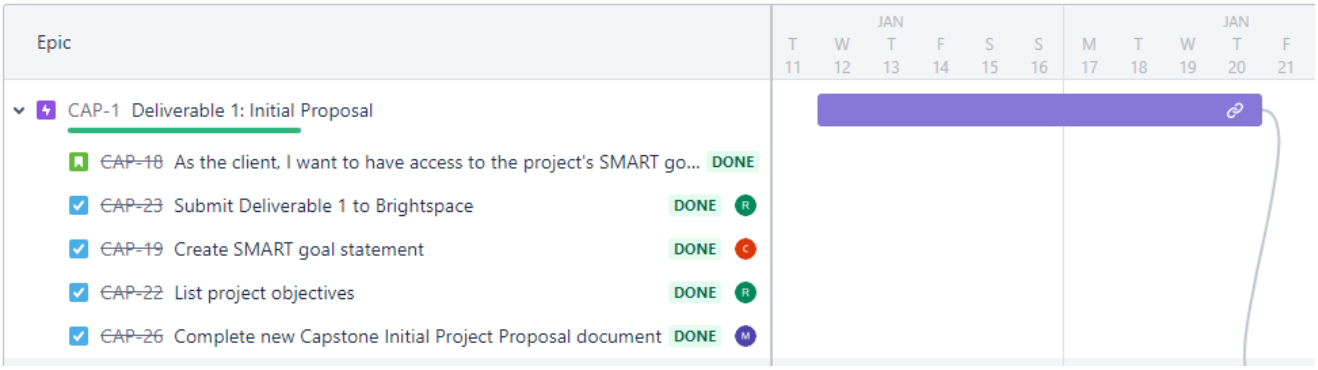
Team members will require adequate knowledge of the product and project related technologies used for their designated tasks, or the knowledge of where to access resources to supplement learning (self-study).

Closure

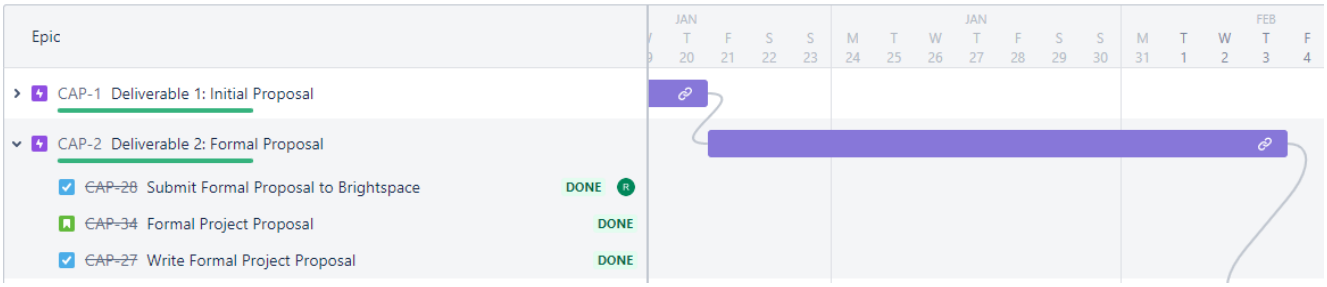
Document	Date Due	Deliverable To	Responsible Team Member	Method of Delivery
Initial Proposal	20/01/2022	David Russell	Project Manager	Brightspace submission
Formal Proposal	03/02/2022	David Russell	Project Manager	Brightspace submission
Project Plan	17/02/2022	David Russell	Project Manager	Brightspace submission
Milestone 1 Report	03/03/2022	David Russell	Project Manager	Brightspace submission
Milestone 2 Report	24/03/2022	David Russell	Project Manager	Brightspace submission
Final Presentation	07/04/2022	David Russell	All team members	In-class presentation
Review and Reflections	14/04/2022	David Russell	All team members	Brightspace submission

Work Breakdown Structure

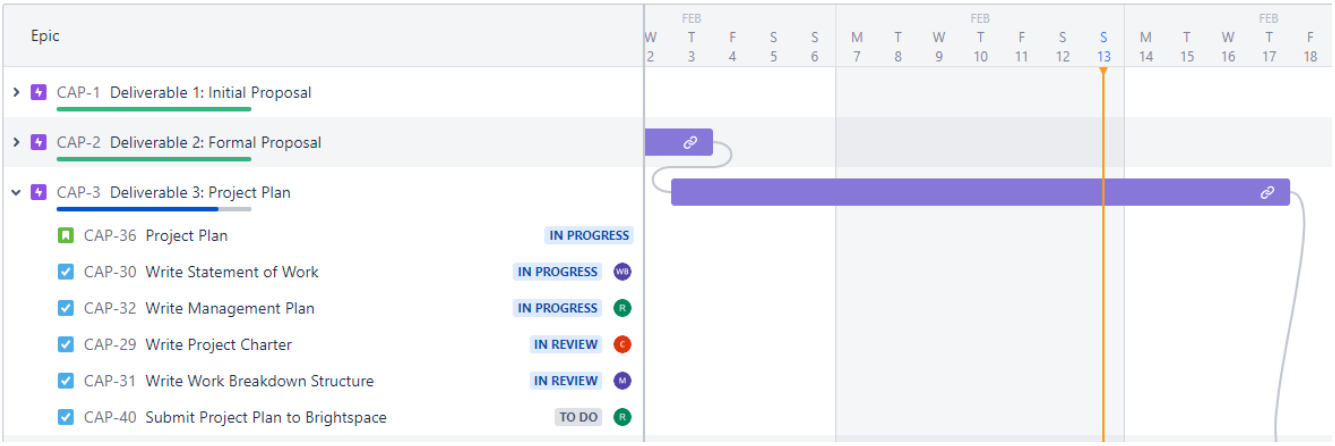
Phase 1



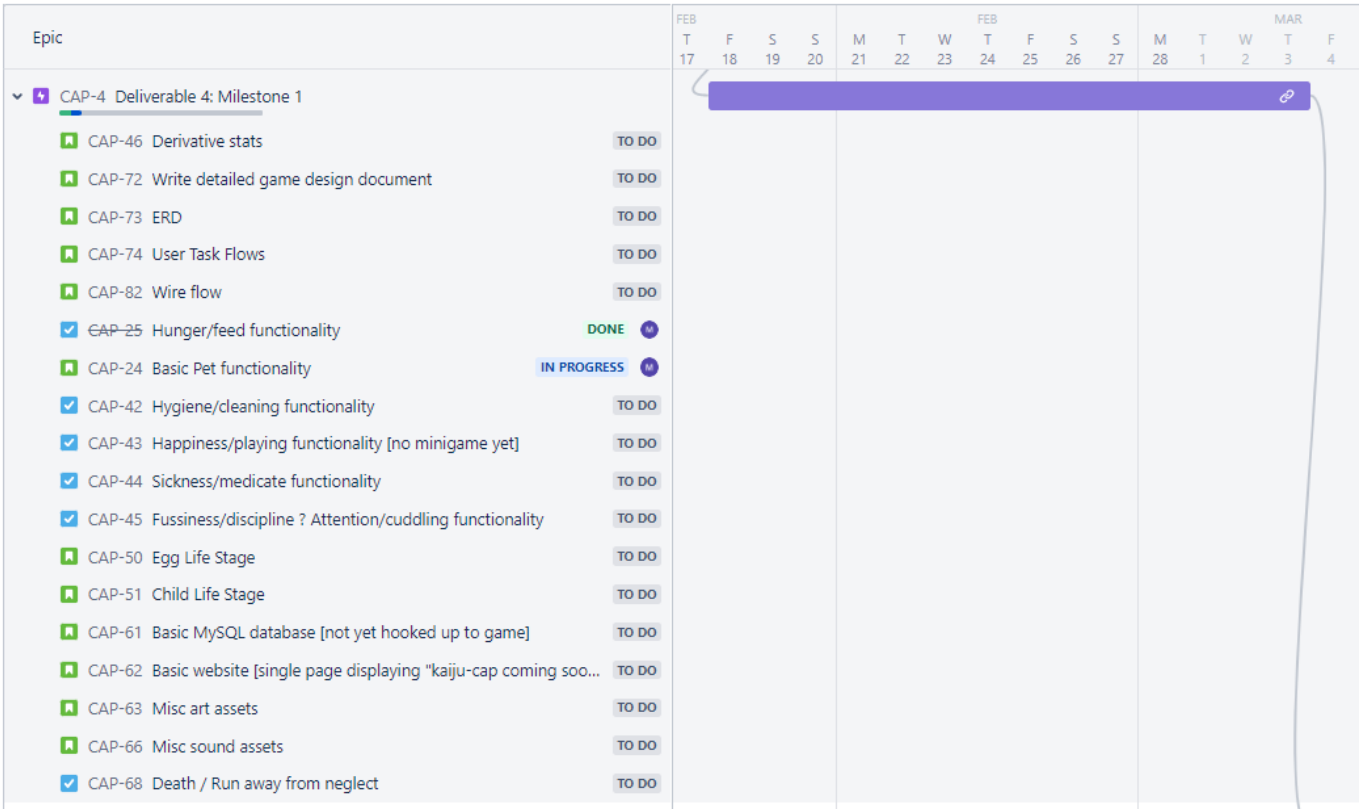
Phase 2



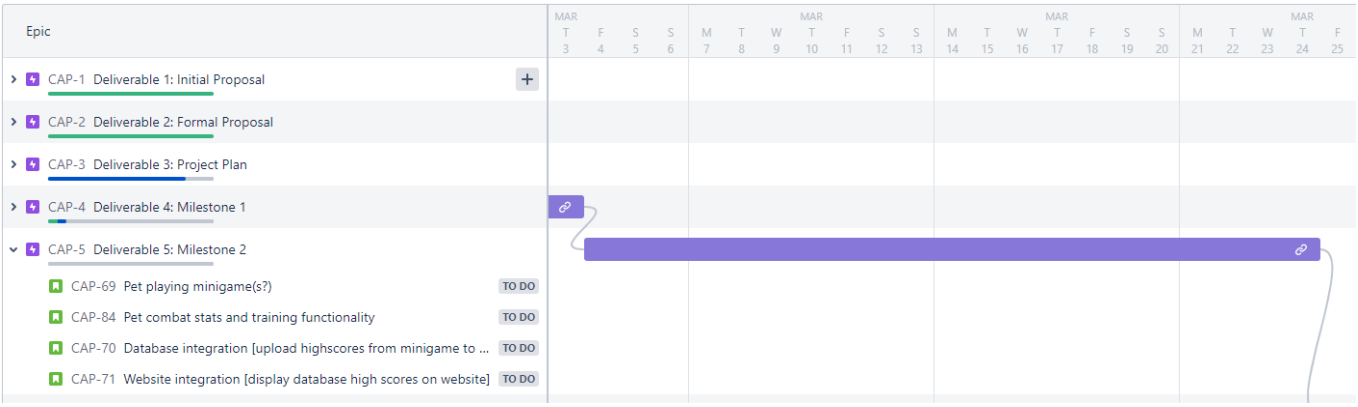
Phase 3



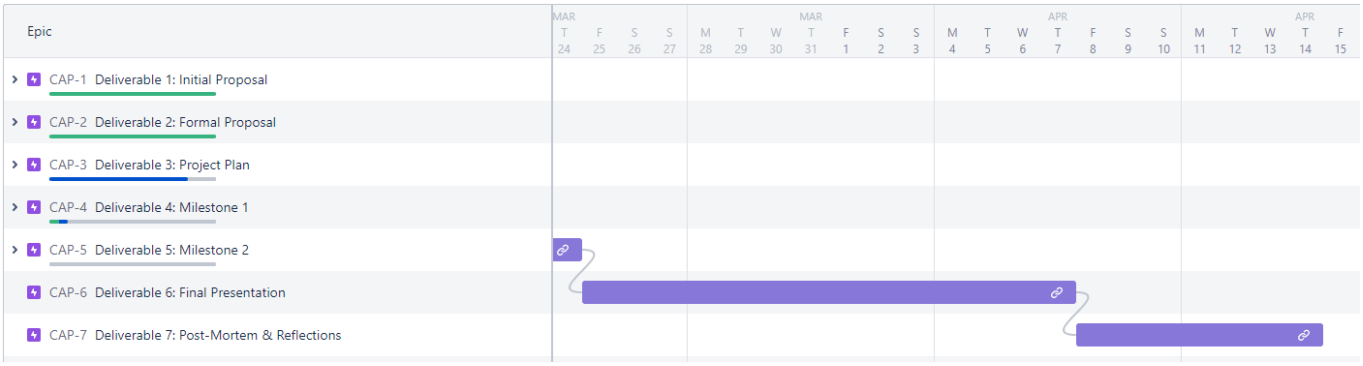
Phase 4



Phase 5



Phase 6 / Phase 7



Management

Scope Management

Should Do

1. Virtual Pet care stats and associated activities
    a. Lifespan / Growth
    b. Hunger / Feeding
    c. Hygiene / Cleaning
    d. Happiness / Playing
    e. Sickness / Medicine
    f. Fussiness / Discipline
2. Pet derivative stats
    a. Age
    b. Weight
    c. Raising Mistakes (tracked but hidden from player)
3. Minigame for playing activity
4. Egg and Child life stages
5. One type of pet
6. Pet assets and animations
7. Background art assets
8. Audio assets

Could Do

1. MySQL database for storing user data and high scores
2. Website for displaying high scores
3. Game deployed onto website for browser play
4. Asynchronous multiplayer allowing combat between two player's pets
5. Pet combat stats and ability to improve them
    a. Strength
    b. Speed
    c. Health
    d. Defense
6. Adult life stage
7. Multiple choices of pet

Quality Management

QA Organization, Tasks, and Responsibilities

- Designated role: Quality Assurance Manager (QAM)
- The QAM will maintain:
    - QA Plan
    - Change Log
    - Library of SAAD / design documentation
    - Library of QA specific documentation

QA Specific Documentation Required

1. Quality Assurance Plan
2. Test Plan
3. Programmer's Reference Manual
4. Change Log (document all changes to the final product that affect milestones already past)

Tools, Techniques, and Methodologies

1. Tools
    a. Developers will use Unity as an IDE and Git/Bitbucket for source/version control. Developers will use version 2020.3.26f1 of Unity. MySQL will be used as the database.
2. Techniques
    a. All files created by the developers should have saved copies whenever changes were made. These files should be committed to the repository accessible by all project members.
2. Methodologies
    a. During testing, a separate branch of the repository will be used to maintain the integrity of the main branch.
    b. A Kanban board (See Project Board) will be used to track known bugs, bugs being worked on, and bugs that have been fixed.

Media Control, Records Collection, Maintenance, and Retention

Steps

1. The development team will use source control via Bitbucket to version and backup all project code. This will be done via the following steps:
    a. Cloning: developer clones the main development branch from the project's repository into the local environment.
    b. Branching: developer creates a new feature-specific or bug-specific branch directly from the Jira issue on the project board, or from the issue as displayed on Jira issue list in Bitbucket. In the local environment, the developer will switch to the issue branch as the working branch. This is the branch used for commits.

   c. Pushing and creating Pull Requests: after the feature is developed or the bug is resolved, the developer then pushes the issue branch to the project repository. After pushing, the developer creates a pull request for the Quality Assurance (QA) Lead to review prior to merging this issue's branch into main.
2. The use of pull requests (PR) will allow other team members to comment on code and ensure code quality and compatibility prior to merging into the main branch. Pull request commenting will also maintain an ongoing record of issue discussions.
3. Developers should keep their local copies of the project code up to date by synchronizing with the project's source code regularly. This will create multiple backups of all code that will allow us to restore the project should anything happen to the Bitbucket repository.

Links

- Project repositories
- Main branch
- Active branches
- Active pull requests
- Commit history
- All project tasks (development branches can be made from each task/issue on this page)

Risk Assessment

Risks and Mitigations

1. Risk 1: Intellectual property (IP) and trademark infringement
   a. Mitigation A: Avoid all trademarked IP (e.g., Toho's Kaiju properties, Tamagotchi-owned terminology).
   b. Mitigation B: Application is not to be published with any IP not owned by the team.

2. Risk 2: External work, life, and commitments overload
   a. Mitigation A: Team members are to try to be self-aware and communicate early to the rest of the team when life events, work overload, or burn-out might become an issue. The team will then re-delegate tasks appropriately and/or adjust project requirements as needed to meet deadlines.

3. Risk 3: Covid 19
   a. Mitigation A: Team members will adhere to all publicly required health and safety policies.

4. Risk 4: Technical difficulties due to unforeseen circumstances (e.g., storm-related power outages)
   a. Mitigation A: In the event of postponed work due to unforeseen events, the team will adjust project requirements as needed to meet deadlines.

Resource Management

- Each team member is responsible for managing and maintaining their own resources and technology needed for project participation.
- Shared digital resources will be managed by the following designated roles:
    - Game code repository: managed by Lead Developer/Quality Assurance Manager
    - Web code repository: managed by Lead Web Developer
    - Communication channels: managed by all team members
    - Project documentation: managed by Project Manager

Stakeholder Management

N/A

Schedule Management

- **Project Schedule**: Project Manager and department Leads are responsible for managing and maintaining project's schedule as documented on the Project Roadmap (Jira).
- **Meeting Schedul**e: All team members will collectively establish meeting times. Each team member is then responsible for managing expectations and updating the rest of the team if rescheduling is needed. If needed, all team members will decide on a new meeting time. All meeting scheduling is documented in Discord. Meeting schedule invitations are sent out by Project Manager via Microsoft Teams.

Change Management

- Change Log (Discord) will be managed by the QAM. All changes will be logged here.
- The product will include an updates and changes list in the accompanying Readme file.

End of document