Population Dynamics Simulation
(PDS)

Product Requirements Document

Prepared by:
Simay Cural, Na'ama Nevo,
James Settles, Walt Jones

Release Date
September 12, 2022

Executive Summary

PDS is a customizable population simulation engine focused on the educational market. PDS is intended to be an easy-to-use alternative to more complex population simulating software like HexSim and SPLATCHE2. A freemium monetization model is to officially be supported by PDS. PDS will be distributed as a standalone application without network connectivity. This document provides nontechnical information regarding the purpose and behavior of PDS.

Document Versioning

Date	Owner	Comment
9/6/2022	Simay	Created document outline, added feature discussion, user cases, example GUI
9/6/2022	Na’ama	Added features to the feature matrix, User Stories
9/6/2022	James	Added executive summary
9/6/2022	Walt	Added project description

Project Description

PDS is simulation software that out of the box will support multiple different types of population simulations, with 4 different modifiable parameters (size, shape, color, and aggressiveness). The end user will not need to modify any code to run the simulation, and all inputs will be managed in a graphical user interface running on the user's local machine.

PDS will not require other programs or network connectivity to be fully functionable and will, ideally, operate as an executable file. PDS will implement a full set of statistical features for educational purposes, such as multiple runs, saving and loading parameters, and ideally visualized results of simulations in the form of graphs, charts, or images. PDS will be multiplatform, and will support different hardware configurations. The simulation runs will ideally be computationally fast and as efficient as possible. PDS will also implement a random parameter generator so that the user doesn't have to specify parameters if they do not want to. PDS will support further development of parameters for populations.

Features

The feature matrix enumerates the features requested for the project and the discussion section provides details regarding the intent of the feature. The ids will be used for traceability. Features that all stakeholders have agreed can be removed should strike-through the feature id and have a comment added to discuss the feature being dropped.

Priority Codes:

H - High, a must have feature for the product to be viable and must be present for launch
M - Medium, a strongly desirable feature but product could launch without
L - Low, a feature that could be dropped if needed

Feature Matrix

ID	Pri	Feature Name	Owner	Comment	Case #
s.1	H	Multiplatform	Sales		
ux.1	H	GUI support	Design	No coding required	
s.2	H	Population of Creatures	Sales	With variance between each creature in population	
e.1	H	Creatures	Sales	With size, shape, color, aggressiveness	
s.3	H	No coding required	Sales	The user will only use the GUI interface	
s.4	H	Standalone Application	Sales		
e.2	H	Interactions	Eng		
e.3	H	2D World	Eng		
ux.2	H	Menu	Design		
e.4	M	Error Messages	Eng		
e.5	L	Statistic Tracking	Eng		
e.6	L	Zero Time	Eng		

Feature Discussion

S.1- Multiplatform

To support marketing and expand potential market penetration, PDS must be able to run on the most common computer platforms. MacOS, Windows, and Linux must be supported.

UX.1 - GUI Support

Most modern computer users who are unfamiliar with the command line expect a GUI that includes hints/prompts to help them know what actions are available.

S.2 - Population of Creatures

To run the simulator the user will have to establish differing groups of populations. Once the simulation is run either one population survives and the others go extinct or the populations live in symbiosis.

E.1 - Creatures

The user has the freedom to adjust certain features of creatures to manipulate the outcome of the simulation. The user also has agency over customizing what the creature will look like on the GUI.

S.3 - No Coding Required

The target users of PDS need not have to have any knowledge on any type of coding to run a simulation. What is expected of the user is to customize their simulation environment when prompted to do so by the GUI.

S.4 - Standalone Application

PDS is envisioned as a standalone application rather than network or web based solution in order to have an easier user experience.

E.2 - Interactions

Objects should trigger events when they collide in the GUI simulation. Different creature object collisions will have different interactions depending on the relationship between the two creatures.

E.3 - 2D World

All simulation spaces are on a 2D rectangular GUI space . Different types of objects/creatures on the 2D map like food, predator, or prey automatically interact with each other to trigger different outcomes. Over time as the population grows or diminishes the world updates itself.

UX.2 - Menu

When the simulation is first run the GUI should prompt the user to customize their creature. While doing so different menus should pop up for different creatures on the user's desktop. This GUI behavior should continue until the user clicks run to start the simulation.

E.4 - Error Messages

When PDS encounters issues that make the game unable to run/load/proceed, an error message of suitable format for a user with no ability to make any changes to the game should be displayed and the game should stop.

E.5 - Statistic Tracking

Once a simulation is run, there should be an output file of the creature features, the population growth, and game time. After having multiple test runs of the same simulation there can be a statistical tracker to analyze simulation data.

E.6 - Zero Time

Most creature interactions are expected to use the same game time. Meaning, once the simulator is run: 1) the population growth/regression and 2) food usage/regeneration should all be in zero time without buffer. In later versions of the simulator a slider can be implemented to adjust creature features, these changes should be made in real time.

User Stories

Todd, a high school biology student
Todd is 17 years old. He is learning about ecosystems and predator prey models, and uses PDS to observe how different populations fluctuate in size due to their changing surroundings.

Kate, from the State wildlife agency
Kate is 36 years old. She teaches at community learning days put on by the State wildlife agency. She uses PDS to educate about wildlife population fluctuations

Nasir, a middle school biology teacher
Nasir uses PDS to add informative graphics to his lectures. He feels his students stay more engaged when they have graphics to follow along with.

Use Case 1

Name	Start Simulation
ID	UC01
Description	User starts a new simulation
Actor	User
Organizational Benefits	
Use Frequency	Very Frequent
Triggers	GUI window for adjusting creature features
Preconditions	JAR file must be implemented correctly
Main Success Scenario	GUI pops up on users desktop ● Prompts user to create a creature
Extensions (error scenarios)	CannotStartSimulation
Alternative Courses	
Post Conditions	Creature creation done, run game

Use Case 2

Name	Enter creature features
ID	UC02
Description	GUI window for user to add creature features
Actor	User
Organizational Benefits	Users can customize the populations to their liking
Use Frequency	Very Frequent
Triggers	New GUI that takes in features
Preconditions	Having the initial GUI window
Main Success Scenario	Feature GUI pops up ● User should be able to customize features like size, color, name, shape, aggressiveness, health, movement etc.
Extensions (error scenarios)	InvalidInputError
Alternative Courses	
Post Conditions	Back to main GUI to run simulation

Use Case 3

Name	Run Simulation
ID	UC03
Description	Once user is done initializing populations the simulation can be run
Actor	User
Organizational Benefits	Getting the simulation to run
Use Frequency	Very Frequent
Triggers	Simulation GUI that runs interactions based on the user input
Preconditions	Having created populations/creatures
Main Success Scenario	User clicks ‘Run Simulation’ Simulation GUI window pops up

	Populations interact based on user input
Extensions (error scenarios)	UnableToRunSimulation
Alternative Courses	
Post Conditions	Population interaction

User Case 4

Name	Creature Interaction
ID	UC04
Description	Creatures interacting on the 2D map
Actor	Populations
Organizational Benefits	User can generate simulation
Use Frequency	Very Frequent
Triggers	An output within populations
Preconditions	
Main Success Scenario	Populations start interacting on the 2D map based on give attributes ● Creatures from the same species can reproduce ● Creatures from different species can attack each other ● Creatures can eat food that randomly generates on map to regain health
Extensions (error scenarios)	
Alternative Courses	
Post Conditions	Populations ending up in symbiosis or parasitic relationship

User Case 5

Name	Statistics Tracker

ID	UC05
Description	Once the populations have reached a certain plateau or a population goes extinct PDD outputs a file about certain statistic from the simulation
Actor	PDD
Organizational Benefits	User can keep track of simulation details and compare scenarios
Use Frequency	Somewhat Frequent
Triggers	The simulation engine should write a save file
Preconditions	Having successfully run the simulation
Main Success Scenario	When a state of symbiosis or extinction is reached PDD should tell the user which state is reached ● Then PDD should create a save file consisting of features of the populations, simulation time, etc.
Extensions (error scenarios)	UnableToSaveStatistics
Alternative Courses	
Post Conditions	User can quit game or try out a new scenario

User Case 6

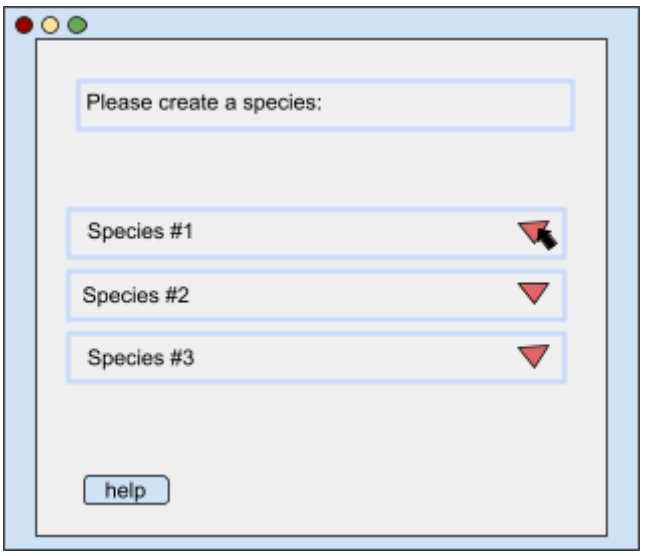
Name	Help
ID	UC06
Description	Help for understanding the simulation rules
Actor	User
Organizational Benefits	Helps the user navigate the simulation, better UX
Use Frequency	Somewhat Frequent
Triggers	A help page with the instructions to the simulation should pop up
Preconditions	Running GUI
Main Success Scenario	User clicks ‘Help’ User reads through instructions User resumes simulation

Extensions (error scenarios)	
Alternative Courses	
Post Conditions	User resumes simulation

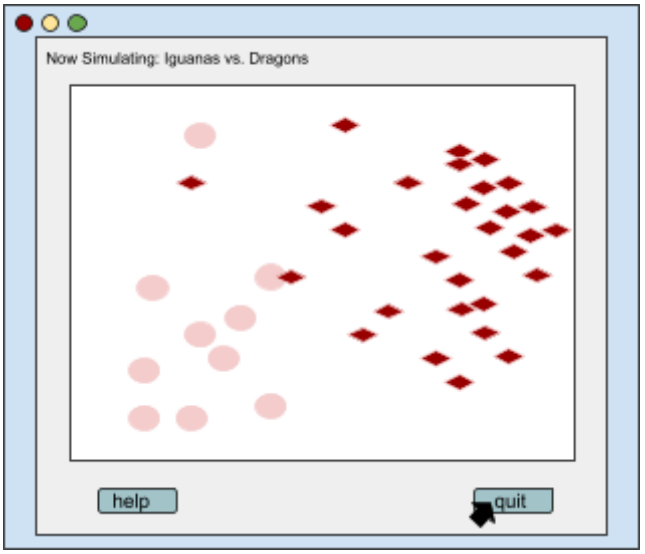
User Case 7

Name	Quit
ID	UC07
Description	User can exit simulation any time
Actor	User
Organizational Benefits	Allows user to end game
Use Frequency	Very Frequent
Triggers	
Preconditions	Running GUI
Main Success Scenario	User clicks ‘Quit’ PDD asks the user if current scenario should be saved in tracker ● If yes, PDD saves the scenario then exits game ● If no, PDD exits game
Extensions (error scenarios)	
Alternative Courses	
Post Conditions	

Example GUI


Please create a species:
Species #1
Species #2
Species #3
help


Now editing: Species #1
Enter Name: Iguana
Movement Speed: ..25.../ 50
Select Shape
Select Color
Reproduction Aggression: .../1000
Food Aggression: .../1000
help
go back
save


Now Simulating: Iguanas vs. Dragons
help
quit