
Population Dynamics Simulation (PDS)

Test Plan

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

Release Date
September 12, 2022

Executive Summary	3
Document Versioning	4
Project Description	5
Features	6
Feature Matrix	6
Test Steps	7
1 - Windows Startup	7
Process	7
Success	7
2 - Apple Startup	7
Process	7
Success	7
3 - Linux Startup	7
Process	7
Success	8
4 - Simulation Generator	8
Process	8
Success	8
5 - GUI Design	8
Process	9
Success	9
6 - Error Messages	9
Process	9
Success	9

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
9/9/2022	Na'ama	Started Document
9/12/2022	Na'ama	Updated test steps

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.

User Acceptance Testing Matrix

The test matrix enumerates tests to be conducted that verify the delivered system meets the requirements from the BRD. Following the matrix, the testing steps for each test are provided. Tests should be able to be completed without understanding of the internal technologies being used.

Feature Matrix

ID	Test Name	Comment	BRD ID
1	Windows Startup		s.1, s.4
2	Apple Startup		s.1, s.4,
3	Linux Startup		s.1, s.4
4	Simulation Generator		ux.1, s.2, e.1, e.2, e.3
5	GUI Design		ux.1, s.3, ux.2
6	Error Messages		ux.1, ux.2, e.4

Test Steps

1 - Windows Startup

Process:

1. Run the Simulation Program on a Windows computer.
2. Verify that the settings window is displayed.
3. Verify that the population and food buttons open new windows that allow the user to input new parameters.
4. Verify that the Run Simulation button successfully starts the simulation in a new window based on the user input.

Success:

- The program opens the Settings window with no errors.
- Clicking the population buttons and food button opens new windows that allow the user to type in parameters or select options from drop down menus.
- The simulation GUI opens when the Run Simulation button is pressed.
- The creature and food populations appear on the GUI window and perform the correct interactions.

2 - Apple Startup

Process:

1. Run the Simulation Program on a Windows computer.
2. Verify that the settings window is displayed.
3. Verify that the population and food buttons open new windows that allow the user to input new parameters.
4. Verify that the Run Simulation button successfully starts the simulation in a new window based on the user input.

Success:

- The program opens the Settings window with no errors.
- Clicking the population buttons and food button opens new windows that allow the user to type in parameters or select options from drop down menus.
- The simulation GUI opens when the Run Simulation button is pressed.
- The creature and food populations appear on the GUI window and perform the correct interactions.

3 - Linux Startup

Process:

1. Run the Simulation Program on a Windows computer.

2. Verify that the settings window is displayed.
3. Verify that the population and food buttons open new windows that allow the user to input new parameters.
4. Verify that the Run Simulation button successfully starts the simulation in a new window based on the user input.

Success:

- The program opens the Settings window with no errors.
- Clicking the population buttons and food button opens new windows that allow the user to type in parameters or select options from drop down menus.
- The simulation GUI opens when the Run Simulation button is pressed.
- The creature and food populations appear on the GUI window and perform the correct interactions.

4 - Simulation Generator

Process:

1. In the settings GUI, enter two populations that have unique characteristics
 - a. One population will be the prey of the other population
 - b. The prey population will eat the food
2. Enter in a food type with a fast regeneration rate.
3. Press the Run Simulation button
4. Check that when two of the same creature collide, they reproduce.
5. Check that when two different creatures collide, the predator creature eats the prey and the prey creature disappears.
6. Check that the food regenerates.
7. Check that the prey creature eats the food when they collide.
8. Check that creatures die (disappear) if they have not eaten in a while.

Success:

- The simulation window opens with the correct creature population based on the input.
- The food regenerates in random locations on the simulation GUI at a constant rate.
- The food disappears when the prey creature collides with it.
- The prey creature disappears when it collides with a predator creature.
- When two of the same creature collide, they sometimes produce another creature.
- Creatures die after some time of not colliding with their food population.

5 - GUI Design

Process:

1. Run the program to open the settings GUI.
2. Click on each population and food button.
3. Verify that the windows that pop up have drop down menus for most features and are not editable.
4. Verify that the user can enter the name of the population in a textfield.
5. Verify that the starting population size can be entered in a textfield as an integer.

Success:

- Settings GUI opens and contains the right buttons.
- The population input window contains drop down menus that are not editable and some JTextFields.
- The save button does not change the window components.
- The text fields are editable.

6 - Error Messages

Process:

1. Enter a non integer into the initial population size JTextField and press the save button.
2. Run the simulation without defining one of the populations.

Success:

- An error appears in the command line when the save button is pressed and there is a non-integer string in the field for initial population size.
- The simulation does not run when the populations are not defined.
- The drop down options prevent errors from occurring in those fields because the user cannot type in something that is not a valid input.