Blob Simulator User Guide

Tony An and Tazik Shahjahan

Last updated 18 Jun. 2018

1. Getting Started

Preface

Welcome to Blob Simulator, an interactive simulation program developed by Tony An and Tazik Shahjahan. This user's guide will detail how to install and use Blob Simulator, as well as some common issues experienced by users.

Installation process:

The Blob Simulator program files are archived in a 7zip folder. 32-bit and 64-bit Windows installers are included, so the user need only install the appropriate program. Then, to run the program, the user need only unzip the archive with a simple right-click command. IntelliJ IDEA or Eclipse is recommended to run Blob Simulator, as the creators cannot guarantee support for older IDEs such as Dr. Java or Ready to Program. Once an IDE has been selected, the user need only click on 'run', and the simulation will begin.

2. How to Use Blob Simulator

Overview

Blob Simulator simulates the behaviour of various coloured circles, obstacles, and environments for the viewing pleasure of the user.

Blobs are the main focus of the simulation. Blobs are circles with 3 main unique properties: radius, speed, and decay rate. Blobs are free to move within the window. When blobs collide, the blob with the larger radius consumes the smaller one and increases in size (radius). However, all blobs will decay in size over time, although at different rates. The user is able to add food, different blobs, obstacles, set environmental parameters, and even play as a blob.

Adding Blobs (Blobs ComboBox)

To add a blob, simply select a type from the Blobs ComboBox (drop-down menu) and then click anywhere inside the window.

There are 8 types of blobs, which can be distinguished by colour.

Blob	Description
Food (Light green)	Stationary blobs that are automatically generated by the program, so that there will always be at least 45 food blobs onscreen. Blobs consume these to increase their radius.
Blob (Green)	The default blob, with no remarkable features. Has average initial radius, speed, and decay rate.

Quick Blob (Red)	A faster variant of the default blob. Starts with smaller initial radius, but has higher speed and slower decay rate.
Slow Blob (Orange)	The opposite of the Fast Blob. Starts with larger initial radius, but has lower speed and faster decay rate.
Sonic the HedgeBlob (Blue)	A blob imitating the iconic SEGA video game character. Has moderate initial radius, extremely high speed, and very fast decay rate.
Blub (Magenta)	The opposite of Sonic. Very large initial radius, extremely slow speed, and very slow decay rate.
Chaser Blob (Grey)	Randomly chooses a target blob and then accelerates towards the current position of the target. Chasers do not target food. They start with relatively large radius and relatively fast decay rate.
User-Controlled Blob (Yellow interior + Black border)	Moderately sized blobs with average decay rate that follow the user's mouse movements.

Adding Other Objects

The user is able to add non-blob squares which affect the blobs. These are also accessible under the Blobs ComboBox, and added to the window in the same way. There are 3 types, which are distinguished by colour.

Object	Description
Wall (White interior + Grey border)	Serves as an impenetrable barrier. Blobs bounce off of walls in the same way they bounce of the boundaries of the window.
Bomb (Black)	When a blob collides with a bomb, it destroys the blob as well as itself.
Magnet (Red)	Causes nearby blobs (including food) to be attracted to the magnetized blob. Once

	magnetized, a yellow aura will appear around the blob. Note that food particles can be magnetized, even though they will not consume each other.

Autogenerate

Toggles a generator that automatically generates blobs of a user-specified type in random positions on the board until the number of blobs exceeds a maximum number. To switch types, the user must first click the Autogenerate button again to deselect it, select a blob from the Blob combobox, and click on Autogenerate again to activate it.

Gravity ComboBox

Allows the user to activate a constant acceleration which affects every blob on the board in one of the four cardinal directions, simulating gravity in that direction. Gravity also affects food blobs. Gravity can also be deactivated at any point, although any blobs that had nonzero velocity at the moment it was deactivated will retain that velocity.

Timescale ComboBox

Allows the user to slow down or speed up the passage of time. The user can set time to be as slow as 5 times slower (0.2x) to as fast as 5 times faster (5x) than the default timescale.

Reset Button

The reset button clears the board and resets the simulation. Any blobs or obstacles added by the user will be removed, and Autogenerate is toggled off. Settings such as Gravity and Timescale are not affected.

3. What Not to Expect

The following features were deemed to be outside the scope of the project and were either not developed to a great extent or just not included in the final product.

- Blob Simulator also does not always handle collisions in the most realistic manner, especially those between blobs and user-created walls
    - When a blob collides with a wall, the signs of the x and y components of its velocity are simply flipped
    - This causes glancing collisions with walls to sometimes look unrealistic (or at least counter to what one might expect).
- Blob simulator does not have very tight or easily maneuverable user controls, but they work
- Blob simulator does not support shapes that do not have a solid fill
    - For example, blobs who show a pattern or picture instead of a solid colour
    - However, blobs and squares can have borders! Truly revolutionary.
- Blob Simulator does not support a local or online multiplayer mode
- Blob Simulator does not allow the user to save or load states

- Blob Simulator does not have an official soundtrack, nor does it have music

4. List of Known Bugs/Issues

The following issues are known to occur while using Blob Simulator.

- After a little time, all subsequent instances of blobs consuming each other throw a large number of exceptions
    - This does not actually affect the performance of the program, but does make the programmers feel inadequate because they could not figure out how to fix it
- Wall collisions are sometimes unrealistic
    - This is due to how collisions are handled
- Occasionally blobs will get stuck in walls, slowly pass through walls, or just fly straight through walls
    - This is more likely to occur to faster blobs
- Spawning a wall on top of a blob causes the blob to get stuck

5. Conclusion

Thank you for installing Blob Simulator. The creators of Blob Simulator hope it provides at least 30 seconds of novelty and fun in your day.