



Physicles

User Manual

Version 1.0 (07-2018)

Index

Index.....	2
Preface.....	3
Before we begin	4
Common gestures in all Physicles	4
Conventions used in this manual	4
1. Physicles Bouncy	5
1.0 Description of Bouncy	5
1.0 Panels: Elements of the screen.....	5
1.1 Box Panel	6
1.1.A <i>AutoRun / Play Sync ON</i> button	6
1.1.B <i>Select Polygon</i> segment control	6
1.1.C Current Polygon	6
1.1.D ‘Box Side’ MIDI notes.....	7
1.2 Playground Panel	7
1.2.A Playground selector.....	7
1.2.B <i>Configuration</i> button	7
1.2.C Playground.....	8
1.2.D <i>Change Velocity</i> buttons	8
1.2.E “MIDI mode” selector	8
1.3 Ball Panel	9
1.3.A Ball selector.....	9
1.3.B Current ball.....	9
1.3.C <i>Ball note</i> picker	10

Preface

NOTE: Physicles will only work with iOS 11+ and with AU Hosts that support MIDI Audio Units – such as AUM, AudioBus 3, apeMatrix, Beatmaker, Cubasis 2, Sequencism, and more in the near future.

Shameless plug: If you want to try Physicles for free, download sequencism – my free music sketchbook tool 😊

Physicles is a container of MIDI Audio Unit plugins, which generate MIDI messages based on the physical interactions between entities using an underlying physics engine (Chipmunk). Physicles plugins (at present, only one plugin exists: Bouncy) do not generate any sound: they simply generate MIDI messages. All compatible AU Hosts have various mechanisms to connect the MIDI output of these plugins to other audio units, synthesizers, or external hardware. Once connected, the receiver instrument will play the generated physics-based sequences of notes.

The plugins included in Physicles are:

1. Physicle **Bouncy**: In this plugin, MIDI messages are created by balls bouncing inside a polygon. Notes are generated whenever a ball collides with a side.

MIDI Audio Unit plugins were possible thanks to pioneers such as Bram Bos (Rozeta), Jonatan Liljedahl (Kimatika – AUM), Mathieu Garcia (INTUA – Beatmaker), and Fred Anton Corvest (FAC). This app also was made possible by the help of multiple libraries (like the underlying physics engine, chipmunk, which rocks!), developers (like Gene De Lisa) and beta testers. Hopefully this small experiment that pursued the replication of some sounds of the past grew into something that will be useful for your own music.

Best,
Rodrigo Roman Castro

Apple and iPad are trademarks of Apple Inc., registered in the U.S. and other countries. All product names and company names are the trademarks or registered trademarks of their respective owners.

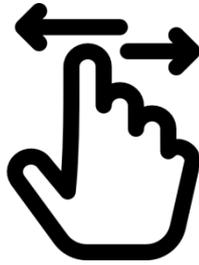
Before we begin

Common gestures in all Physicles



TAP

Creates an element, stops the movement of an element.



SWIPE

Changes the direction, speed or rotation of an element.



DOUBLE TAP

Deletes and element, or resets its position.



LONG PRESS

Performs various special operations.

Conventions used in this manual

NOTE: This icon signifies a general note

TIP: This icon signifies a tip or suggestion.

WARNING: This icon indicates a warning or caution

Cursive text indicates an element / option of the application. Underlying text highlights an operation that the user can perform.

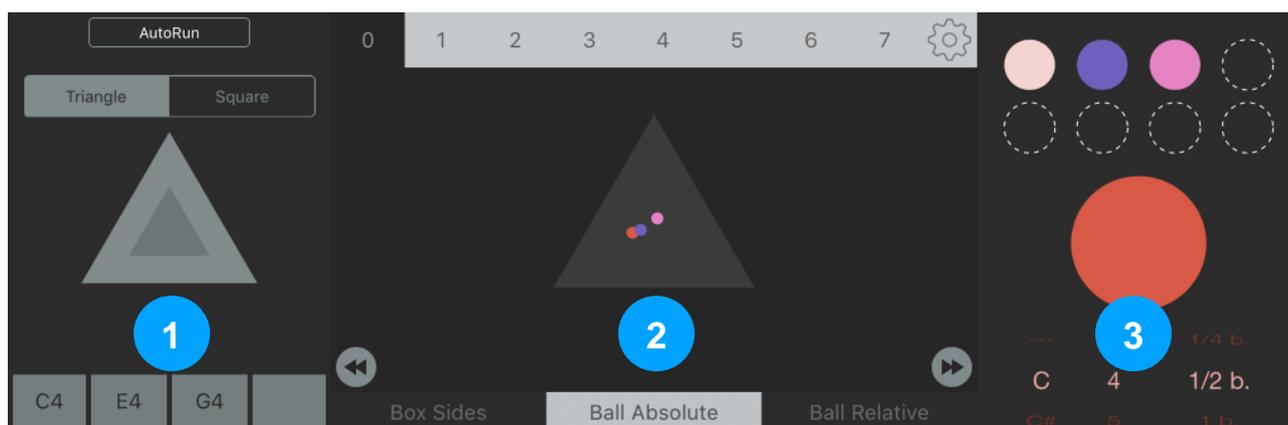
1. Physicle Bouncy

1.0 Description of Bouncy

Bouncy provides eight *playgrounds* where multiple *balls* bounce inside a *box* (polygon). MIDI messages are generated whenever a ball collides with the side of the box. The note played by the collision depends on the current *MIDI mode* of the playground: “Box Sides” (notes are assigned to the side of the box), “Ball Absolute” (every ball can have an associated note), and “Ball Relative” (notes only send MIDI message whenever the host piano is being played).

NOTE: Although I will use the word “box” to refer to the polygon where the balls are bouncing, the polygon can also be a triangle.

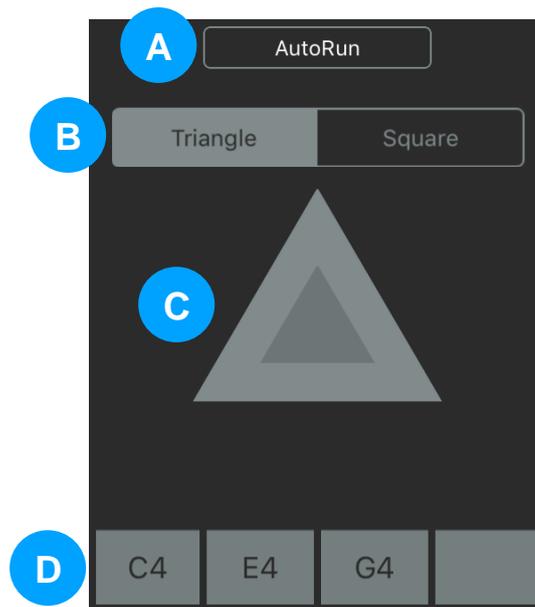
1.0 Panels: Elements of the screen



The main panels in Bouncy are as follows:

1. **Box panel.** This panel shows the current selected box. It also shows what notes are played in the “Box Sides” MIDI mode. Besides, it also allows users to select when the physics simulation is active.
2. **Playground panel.** This panel allows users to interact with the current playground, where the balls bounce inside the box. It also allows users to choose a playground, choose a MIDI mode, and show the configuration menu, amongst other options.
3. **Ball panel.** This panel allows users to manage the balls of the current playground: select them, change their direction and speed, change the notes they produce, delete them, and other operations.

1.1 Box Panel



1.1.A *AutoRun / Play Sync ON* button

This button indicates whether Bouncy is running the underlying physics simulation at all times, or only when the host is playing, respectively. TAPPING the button will switch between these two modes.

WARNING: Due to the behavior of certain hosts, in some cases Bouncy will not run the underlying simulation while in the AutoRun mode.

1.1.B *Select Polygon* segment control

This control allows users to select the polygon that encloses the balls for the selected playground.

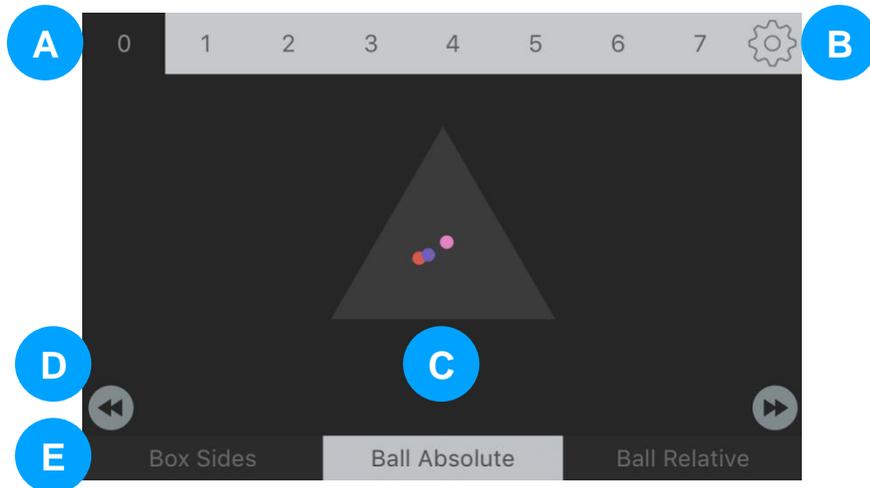
1.1.C *Current Polygon*

This control shows the current selected polygon and its state. SWIPING will rotate the polygon left or right. TAPPING the polygon will stop its rotation, and DOUBLE TAPPING the polygon will restore it to its initial state.

1.1.D 'Box Side' MIDI notes

This control shows the notes that are played in the 'Box Side' MIDI mode. While in this mode, users can **PLAY THE PIANO OF THE HOST** to change these notes. The notes will be changed in sequence.

1.2 Playground Panel



1.2.A Playground selector

This control allows the user to change the current playground, immediately loading the box and balls of that particular playground.

NOTE: In smaller iPhone screens, the position of the Playground selector will change in order to provide more screen space to the Playground.

1.2.B Configuration button

This control opens the configuration menu, where users can toggle the following options:

- **MIDI changes playgrounds.** If active, the user can change the playground using the piano of the host. The notes used to change the playground are C1 (24), D1 (26), E1 (28), F1 (29), G1 (31), A1 (33), B1 (35), and C2 (36).
- **Show ball linear velocity.** If active, the linear velocity of the ball will be shown whenever it is changed by the user.

1.2.C Playground

Shows the current playground.

TAPPING the playground will create a new ball. Note that balls will not be created if there are already eight balls within the box.

1.2.D *Change Velocity* buttons

These buttons allow users to change the linear velocity of all balls by TAPPING them. The left button will slow down all balls, while the right button will speed up all notes.

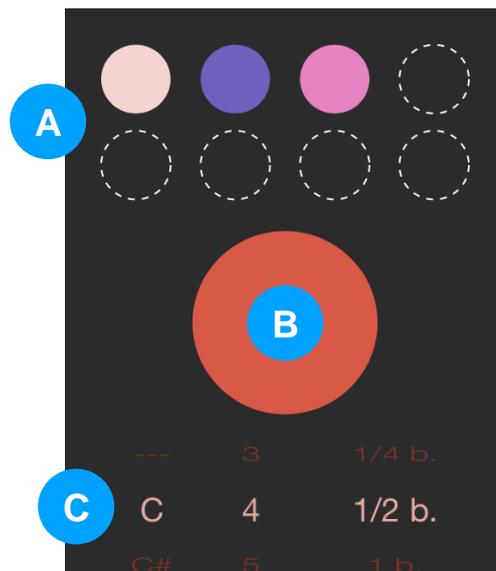
NOTE: The velocity of ‘frozen’ balls will not be changed.

1.2.E “MIDI mode” selector

This control allows users to change the current MIDI mode. Available modes are:

- ‘Box Side’ mode. Upon collision with the box, balls will generate MIDI notes depending on the notes associated with the sides of the box. As aforementioned, in this mode, the notes of the sides of the box can be changed using the host piano.
- ‘Ball Absolute’ mode. Balls play their associated note when colliding with the box. Note that a ball can have an empty note (---). In that case, the ball will play the note associated to the side of the box, just like in the ‘box side’ mode.
- ‘Ball Relative’ mode. Balls only play their associated note when users play the host piano. The notes played are the notes played by the user in the host piano plus the offset indicated in the Ball Panel. This effectively allows users to “play” Bouncy 😊.

1.3 Ball Panel



1.3.A Ball selector

This control allows users to select the current ball by TAPPING them.

1.3.B Current ball

This control shows the current selected ball and its state. SWIPING will launch the ball towards the direction of the swipe. The linear velocity of the ball will also depend of the speed of the swipe. TAPPING the ball will stop it. DOUBLE TAPPING the ball will delete it.



An important concept here is the concept of a 'frozen' ball. LONG PRESSING the ball will toggle the 'freeze' state, which will change the state of the ball (as seen in the image above). When frozen, the linear velocity of the ball will never change.

TIP: "Freezing" a ball is useful when a box is rotating, as (due to the underlying physics) a ball bouncing against a rotating box will increase its speed over time.

NOTE: The velocity of 'frozen' balls will not be changed when swiping.

1.3.C *Ball note picker*

This control allows users to change the MIDI note played by the ball when colliding with the sides of the box. Users can choose the note, octave, and note length.

NOTE: Not all options will be available in certain MIDI modes.

NOTE: As aforementioned, a ball that has a “---” note in the ‘Ball absolute’ mode will behave as in the ‘Box sides’ mode.

