**Revised Specification**

**Building Mode**

In building mode the user can

* Add any of the available types of gizmos to the playing area.
* Move a gizmo from one place to another on the playing area.
* Attempting to place a gizmo in such a way that it overlaps a previously placed gizmo or the boundary of the playing area should be rejected and have no effect, the board should return to the state it was in before the rejected action was taken.
* Apply a 90 degree clockwise rotation to any gizmo. This actions should have no effect on gizmos with rotational symmetry (they should still be rotated).
* Delete a gizmo from the playing area.
* Connect a particular gizmo's trigger to a particular gizmo's action.
  + The standard gizmos produce a trigger when hit by the ball, and exhibit at most one action (for example, moving a flipper, shooting the ball out of an absorber, or changing the colour of a bumper). The trigger that a gizmo produces can be connected to the actions of many gizmos. Likewise, a gizmo's action can be activated by many triggers. The [required triggers and actions](https://personal.cis.strath.ac.uk/murray.wood/Gizmoball/Gizmoball_spec.htm#standard-gizmos) for the basic gizmos are described below.
  + Note that triggers do not "chain". That is, when A is connected to B and B is connected to C, a ball hitting A should only cause the action of B to be triggered.
* Connect a key-press trigger to the action of a gizmo.
  + Each keyboard key generates a unique trigger when pressed. As with gizmo-generated triggers, key-press triggers can also be connected to the actions of many gizmos.
* Add a ball to the playing area.
  + The user should be able to specify a position and velocity.
  + An attempt to place the ball in such a way that it overlaps a previously placed gizmo or the boundary of the playing area should be rejected (i.e., it should have no effect). There is one exception in the [standard gizmo set](https://personal.cis.strath.ac.uk/murray.wood/Gizmoball/Gizmoball_spec.htm#standard-gizmos): a stationary ball may be placed inside an absorber.
* Save to a file named by the user, the file will be in the standard format given. You may, if you wish, define an extension to the standard format that handles special features of your implementation. If you do so, the user must have the choice of saving in the standard format or in your special format.
  + The saved file must include information about all the gizmos currently in the playing area, all of the connections between triggers and actions, and the current position and velocity of the ball.
* Load from a file named by the user. You must be able to load a game saved in the standard format.
* Switch to running mode.
* Quit the application.
* connect the action of flippers and bumpers to triggers, such as a keyboard key being hit or one of the bumpers being bumped, and
* save and load the user's game configuration to and from a file.
* The gizmo palette on the side provides the user with a variety of operations (square, circle, triangle, flipper) for placing gizmos in the playing area.
* The "modifications" toolbar on the bottom provides the user with a variety of operations (move, delete, rotate) for editing the gizmos in the playing area.
* The modifications toolbar also provides a connect button that connects the trigger of one gizmo to the action of another. After this button is pressed the user can connect gizmos together. For example, the user might press the connect button, then click on one of the circular bumpers, and then click on one of the flippers. As a result, every time the bumper's trigger is activated (which occurs when a ball hits the bumper), the flipper's action (to rotate around its pivot) will occur. Alternatively, the user might press the connect button, then press a key, then answer a question about whether the up or down keypress is of interest, then click on one of the flippers. As a result, every time the user depresses (or, respectively, releases) that key, the flipper will move. Several triggers may activate the same gizmo.

**Running Mode**

In running mode, the user can:

* Press keys, thereby generating triggers that may be connected to the actions of gizmos.
* Switch to building mode at any time.
  + If the user requests to switch to building mode while a flipper is in motion, it is acceptable to delay switching until the flipper has reached the end of its trajectory.
  + Similar short delays in order to finish transitional states of gizmos you create are also acceptable.
* Quit the application.

In running mode, Gizmoball should:

* Provide visually smooth animation of the motion of the ball.
  + The ball by default must have a diameter of approximately 0.5L.
  + Ball velocities must range at least from 0.01 L/sec to 200 L/sec and can cover a larger range if you wish. 0 L/sec (stationary) must also be supported.
  + An acceptable frame rate should be used to generate a smooth animation. We have found that 20 frames per second tends to work well across a reasonably wide range of platforms.
* Provide intuitively reasonable interactions between the ball and the gizmos in the playing area. That is, the ball should bounce in the direction and with the resulting velocity that you would expect it to bounce in a physical pinball game.
* Continually modify the velocity of the ball to account for the effects of gravity.
  + You should support the standard gravity value of 25 L/sec2, which resembles a pinball game with a slightly tilted playing surface.
* Continually modify the velocity of the ball to account for the effects of friction.
  + You should model friction by scaling the velocity of the ball using the frictional constants *mu* and *mu*2. For sufficiently small *delta\_t*'s you can model friction as *Vnew* = *Vold* \* (1 - *mu* \* *delta\_t* - *mu*2 \* |*Vold*| \* *delta\_t*).
  + The default value of *mu* should be 0.025 per second.
  + The default value of *mu2* should be 0.025 per L.

General

Ball properties

Ball has a maximum velocity of 200L/sec and a minimum velocity of -200L/sec

The vector of the ball can only have a maximum magnitude (velocity) of 200 L/sec, any higher values will be rounded down, there will not be a limit on the direction the ball can move.

The ball can have a velocity of 0 L/sec, though there may be scenarios where friction or gravity is set to 0, the ball may get stuck.

Keypresses

Gizmos

Friction and Gravity

* The values for friction may be changed
* The value for gravity may be changed, the minimum gravity will be 0 L/sec2, direction of gravity may not change.
* The default gravity setting is 25 L/sec2
* The menu at the top of the window allows the user to **save** or **load** game configurations and to **run** or **stop** the game. In the animation, a game is in progress: the ball is the small blue circle which started in the lower right-hand corner. The ball bounces off the red, green, and blue bumpers, and is hit by the yellow flippers.