

6.170 Laboratory in Software Engineering

Fall 2005

Final Project Amendment: Gizmoball

Due: See [UPDATED SCHEDULE](#)

Note: There has been a change in schedule. **We have moved the final deadline to 9:00 AM, December 12th.** Please use this time to ensure that your design is thoroughly tested, and to make sure that your interface is as clean and usable as imaginable.

Contents:

- [Introduction](#)
- [Ghost Balls](#)
 - [Multiple Balls](#)
 - [Ghost Ball Definition](#)
- [Extender Gizmo](#)
 - [Extender Placement](#)
 - [Transforming Mode](#)
- [File Format](#)
- [Errata](#)

Introduction

For the amendment, two additional features are required: ghost ball support and an implementation of the extender gizmo.

One of the issues that you will likely encounter in a future software engineering project is that your client "roughly" knows what he wants, but he doesn't have many of the details resolved in his mind. To simulate such a situation, parts of this amendment have been left intentionally vague. It is up to you to decide what (if any) clarifications are necessary, and **part of your grade** will be based on how well you identify these ambiguities and border cases. You should include any changes to your Preliminary Specification documentation in your final submission.

Ghost Balls

Multiple Balls

The original specification does not explicitly require the application to support multiple balls, but a good design should handle this easily. The following basic requirements must be satisfied:

- **Building Mode:** the user must be able to add any number of balls to the playing area, and specify the initial velocity vector of each; the user should also be able to move or remove any ball from the playing area. An attempt to place a ball in such a way that it overlaps a previously placed gizmo or ball, or the boundary of the

- playing area should be rejected (i.e., it should have no effect). There is one exception: a stationary ball may be placed inside an absorber.
- **Running Mode:** ball-to-ball collisions should be handled in a realistic manner, as should ball-to-gizmo collisions. Colliding with any ball should trigger a gizmo in the standard gizmo set.

Note that the behavior of absorbers with respect to multiple balls is not specified. It is your responsibility to devise a reasonable solution and document it in your design.

Ghost Ball Definition

Each ball in the playing area can be marked as a ghost ball. A **ghost ball** is one that collides only with gizmos and not other balls; otherwise, the behavior of ghost balls and normal balls are identical. During running mode, ghost balls should be visually distinguishable from normal balls.

Extender Gizmo

An extending/contracting Gizmo with integral horizontal and vertical ranges of motion.
Trigger: generated whenever a ball hits it
Action: toggles between stationary and transforming modes (described below).
Coefficient of reflection: 1.0, see note below.

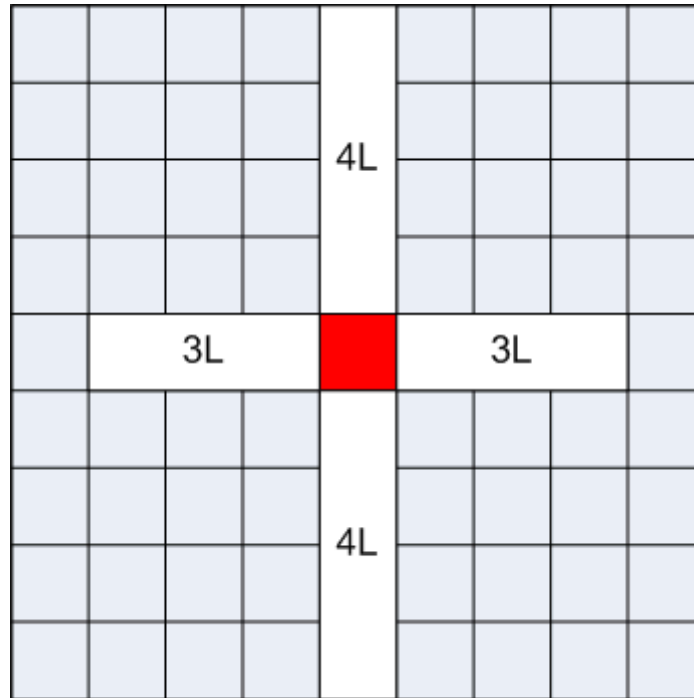
The extender gizmo is a bumper whose shape can change while in running mode. It initially occupies a space of one square, extends horizontally to a rectangle, contracts to its original shape, extends vertically, contracts, and so on.

An extender is always in one of two modes:

1. **stationary** mode: the shape of the extender stays fixed at the position it had when stationary mode was entered.
2. **transforming** mode: the shape of the extender changes indefinitely in the manner prescribed below, beginning at the position it had when transforming mode was entered.

Extender Placement

While placing an extender on the board, the user should be able to specify its **span of motion** in both the horizontal and vertical directions. Assuming the upper-left corner of the extender's initial (1Lx1L) shape was placed at location (X,Y), its horizontal span can be an integer between 0L and $\min\{X, 19L-X\}$ (inclusive), and its vertical span can be an integer between 0L and $\min\{Y, 19L-Y\}$ (inclusive). A horizontal span of 3L means that the extender can extend from its initial 1L width by 3L both to the left and to the right. Vertical spans are defined in a similar fashion. The following example shows an extender with a horizontal span of 3L and a vertical span of 4L:



An extender with x and y spans of $3L$ and $4L$ (in its initial position)

Placement of other objects: when determining potential overlap between an extender and another *gizmo* during building mode, the cross-shaped space corresponding to the extender's entire range of motion may be used. However, when determining overlap between a ball and an extender during building mode, only the actual $1L \times 1L$ center square, in which the extender initially lies, should be used; that is, the user can place balls in the white area in the figure above.

Transforming Mode

Let XS and YS be an extender's horizontal and vertical spans. Then the extender cycles between the following states in transforming mode (an W -by- H rectangle has width W and height H):

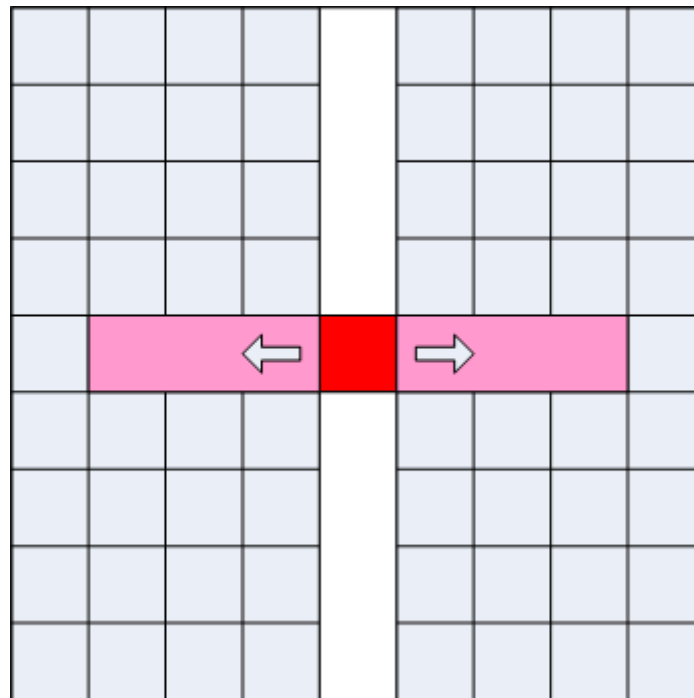
1. a $1L$ -by- $1L$ square in the center of the horizontal and vertical ranges of motion (initial state)
2. an $(2XS+1)$ -by- $1L$ rectangle spanning the entire horizontal range of motion
3. a $1L$ -by- $1L$ square in the center of the horizontal and vertical ranges of motion (same as initial state)
4. a $1L$ -by- $(2YS+1)$ rectangle spanning the entire vertical range of motion

The extender should pause between state transitions for a short but reasonable amount of time. The extender transitions from one state to the next by extending (or contracting) at the same linear velocity on each end of the rectangle. Transitions should be **smooth** (both in its animation and in its collision detection handling). That is, the transition from one state to the next should happen smoothly over the course of multiple frames, and during any particular frame, the underlying physical representation of the extender should be equivalent to the rectangle that is displayed on the screen. The speed at which each transition occurs is at your own discretion, provided that it is between $0.5L/s$ and $4L/s$ on each end (i.e. the extender increases/decreases in width/height at twice that rate since both ends move simultaneously).

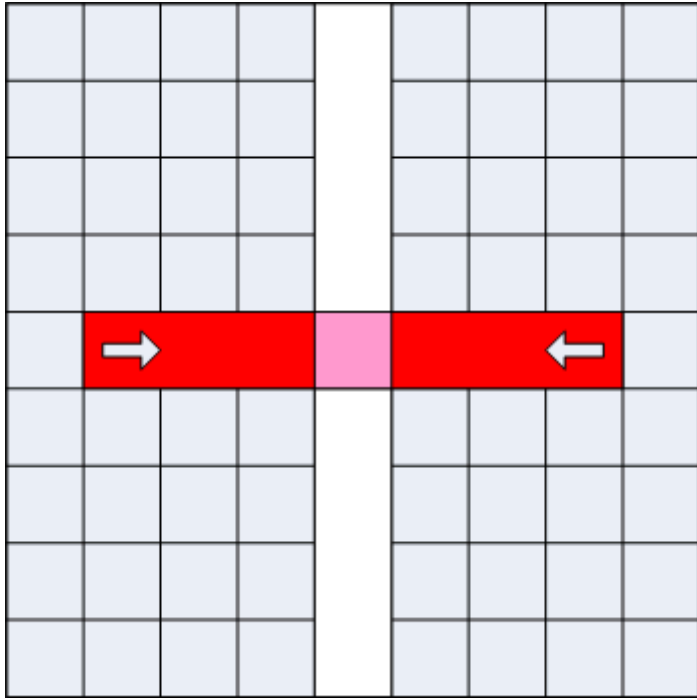
Physics: Note that the physics of collisions between a ball and a transforming extender are intentionally underspecified. The minimum requirement for such a collision is that the ball must bounce off the extender with a coefficient of reflection of 1.0. That is, you are **not** required to model the physics of energy transfer from a transforming extender to a ball and can essentially treat it as stationary (but its underlying shape must be the one that is displayed at the time of collision!).

However, if you would like to implement something more realistic, you should consider and decide how elastic this collision might be. Also, a relatively simple (but physically inaccurate) way to get something that looks reasonably realistic is to use the idea of relative reference frames. That is, to treat a collision between a moving ball and a moving extender, you can adjust the ball's vector velocity by the extender's and subsequently treat the extender as stationary. Note that you will have opposite vector velocities on the extender depending on which half the collision occurs at.

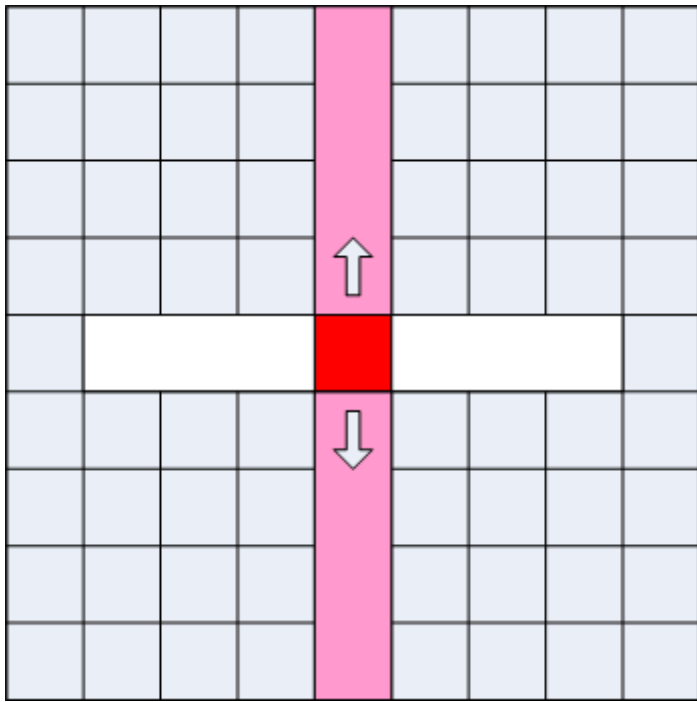
Summary: the following figures summarize the transitions between states:



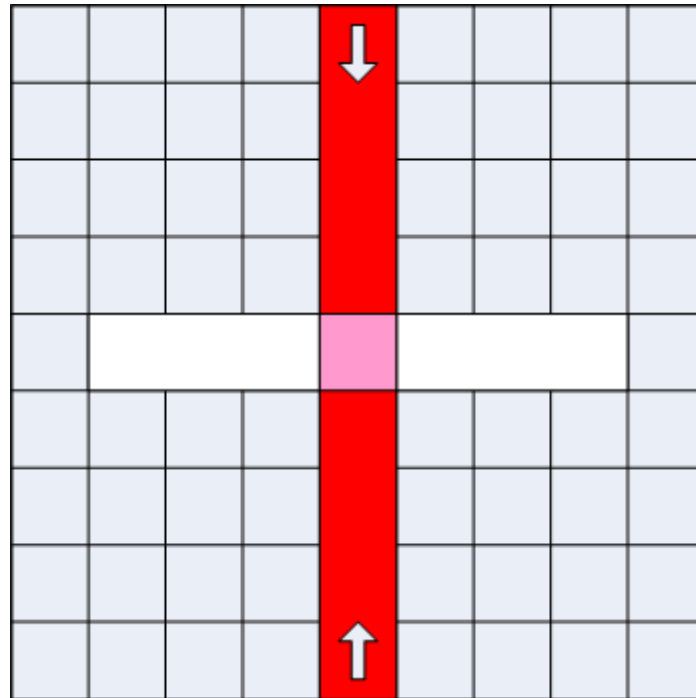
State 1 to 2



State 2 to 3



State 3 to 4



State 4 to 1

File Format

The amendment of multiple balls requires no change to the file format, except that there can be multiple "ball" commands each corresponding to a new, different ball. In order to accommodate ghost balls and Extender gizmos, we made the following changes to the XML schema.

1. A ball now has an optional new attribute that declares whether it is a ghost ball or not.
2. A new element, Extender, has been added. The Extender declaration includes the Extender's initial position and its span in the x and y direction.

The new XML schema file which includes specifications for the new features (ghost balls and Extender gizmos) can be found in the projects section. Connections between Extender Gizmos and other Gizmos are made using the same Connection tag as before. Here is a sample file using ghost balls and the new Extender gizmo element:

```
<board>
<ball name="Ball1" x="1.8" y="4.5" xVelocity="-3.4" yVelocity="-2.3"
ghost="false" />
<ball name="Ball2" x="7" y="4.5" xVelocity="1.4" yVelocity="2.3"
ghost="true" />
<ball name="Ball2" x="7" y="9.5" xVelocity="1.4" yVelocity="2.3" />
<gizmos>
  <extender name="W1" x="11" y="4" xspan="3" yspan="4" />
  <extender name="W2" x="7" y="9" xspan="1" yspan="1" />
</gizmos>
<connections>
  <connect sourceGizmo="W1" targetGizmo="W2" />
</connections>
</board>
```

Here is the syntax and semantics of the updated schema descriptions.

```
<ball name="STRING" x="FLOAT" y="FLOAT" vx="FLOAT" vy="FLOAT"
g="BOOLEAN" />
```

Creates a ball with center (x, y) , initial velocity (v_x, v_y) -- no change from before. If the attribute g is not present, then create a regular ball. If the attribute g is present, then creates the ball as a ghost ball if g is `true`. Within the file, the `name` must be unique, and may be used later to refer to this specific ball.

```
<extender name="STRING" x="INTEGER" y="INTEGER" xs="INTEGER"
ys="INTEGER" />
```

Creates an Extender gizmo with its initial position's upper left-hand corner at (x, y) , and a spans x_s squares in the positive and negative x direction, and y_s squares in the positive and negative y direction.

In the above sample file, 3 balls are created: 1 ghost (Ball2) and 2 regular (Ball1, Ball3). In addition, two Extender gizmos are created. Extender gizmo W1 describes the spanning of the example in the screen shots above. Note that the connection specifies that each time a ball hits Extender W1, W2 will switch between stationary and transforming mode.

Errata

1. An incorrect statement was made in the above about extender physics. At the time of the writing, the author did not fully appreciate the fact that the edge of the extender and the rest of the playing field are indeed two frames in a constant relative motion (more specifically, that the velocity of the extender is unchanged by a collision). But in fact, a Gallilean transformation as suggested in the above would indeed be physically accurate.

2. There is an inconsistency in the syntax of the extender and ghost ball as defined in the previous section. The provided schema file is correct so if you have been using the provided schema to validate the xml boards, this error should not affect you.

In the previous section,

```
<ball name="STRING" x="FLOAT" y="FLOAT" vx="FLOAT" vy="FLOAT"
g="BOOLEAN" />
```

should be

```
<ball name="STRING" x="FLOAT" y="FLOAT" xVelocity="FLOAT"
yVelocity="FLOAT" ghost="BOOLEAN" />
```

and

```
<extender name="STRING" x="INTEGER" y="INTEGER" xs="INTEGER"
ys="INTEGER" />
```

should be

```
<extender name="STRING" x="INTEGER" y="INTEGER" xspan="INTEGER"
yspan="INTEGER" />
```