

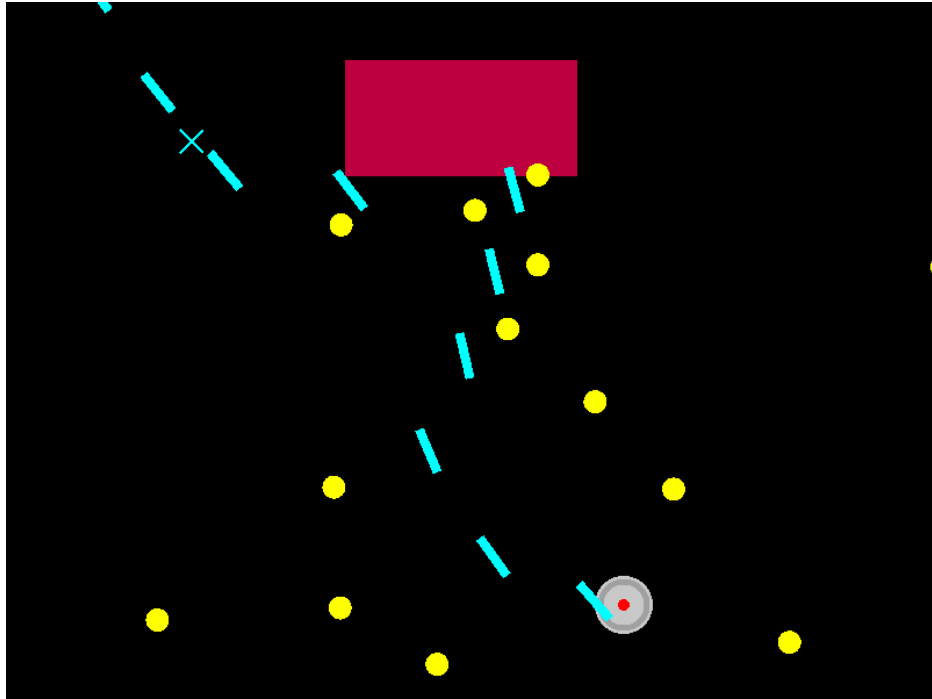
Low-Level Game Development

Prototype Games

High Level



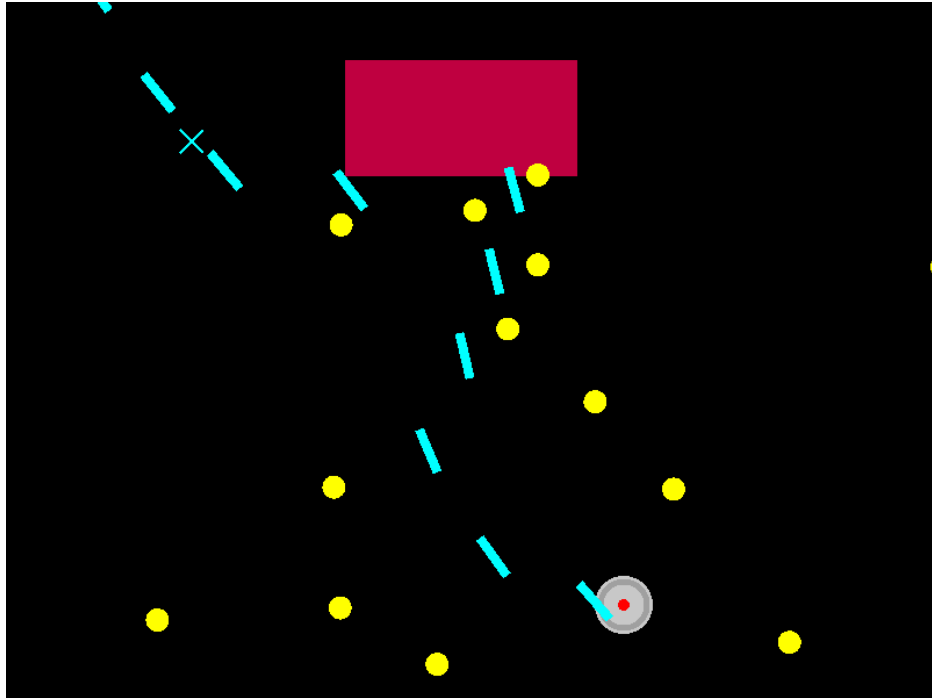
Low Level

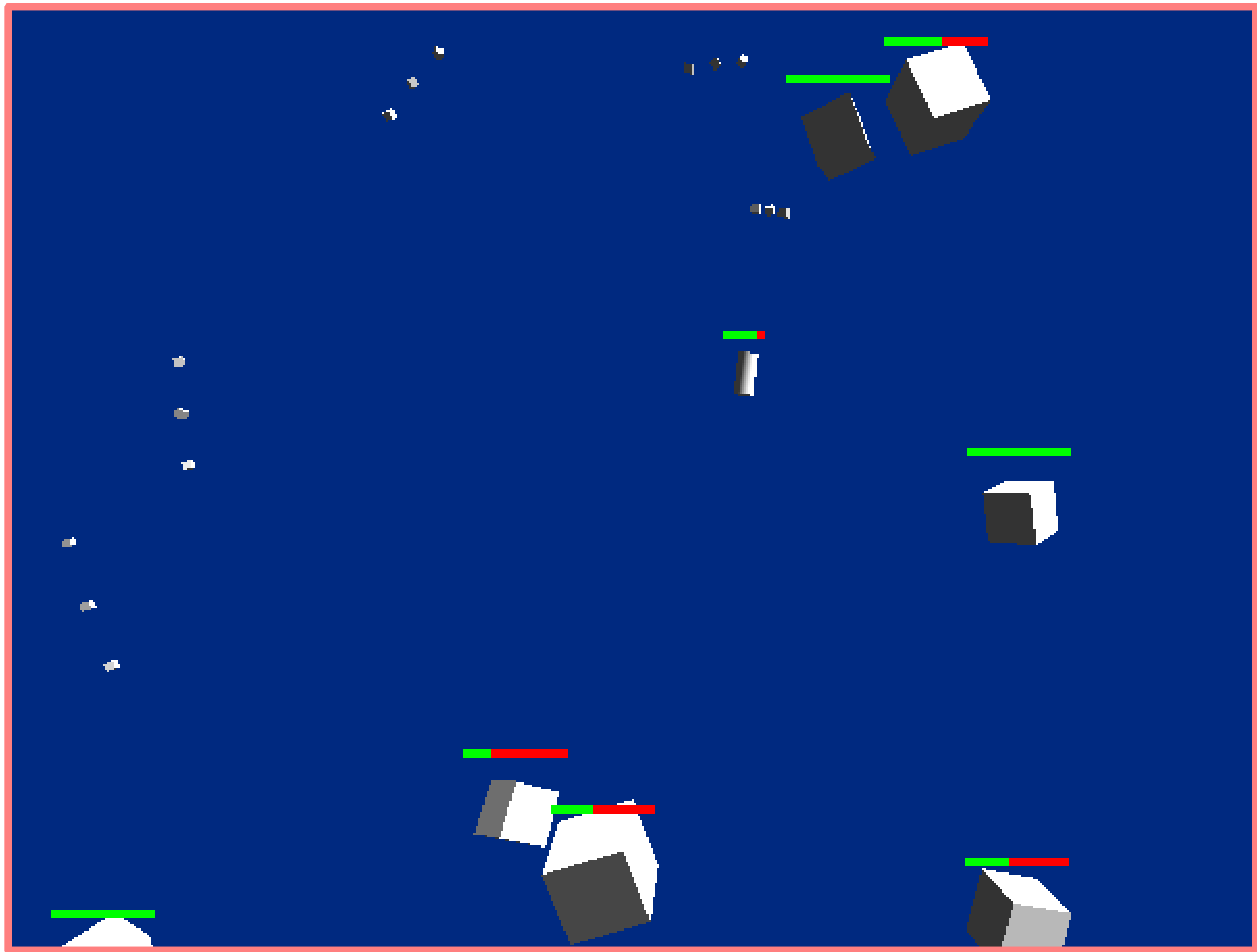


High Level



Low Level





Physics Engine



Game Objects

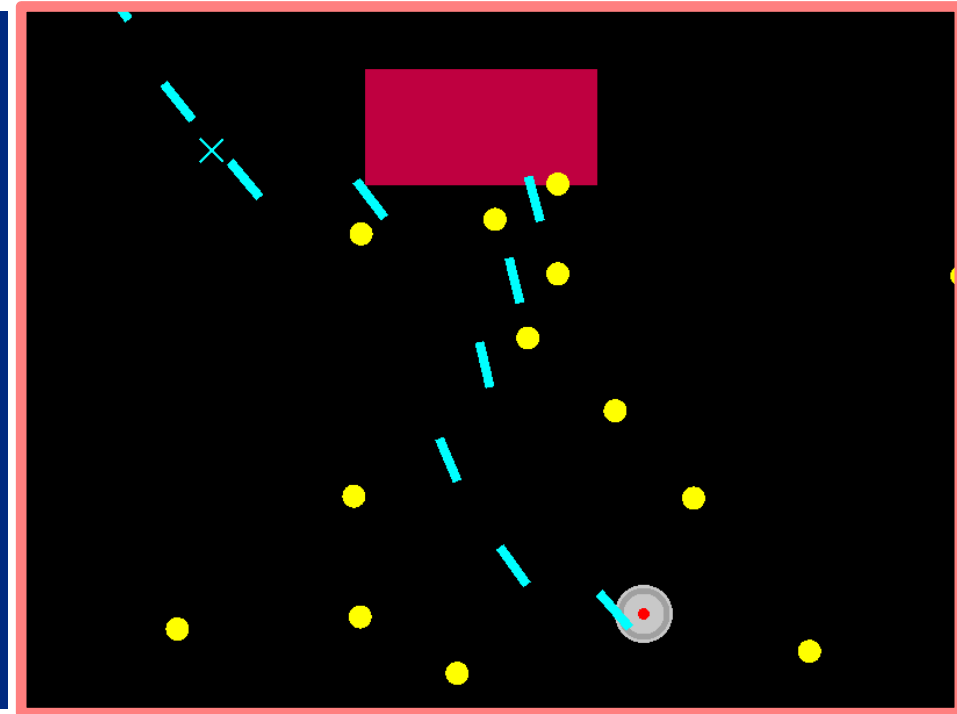
Physics Engine

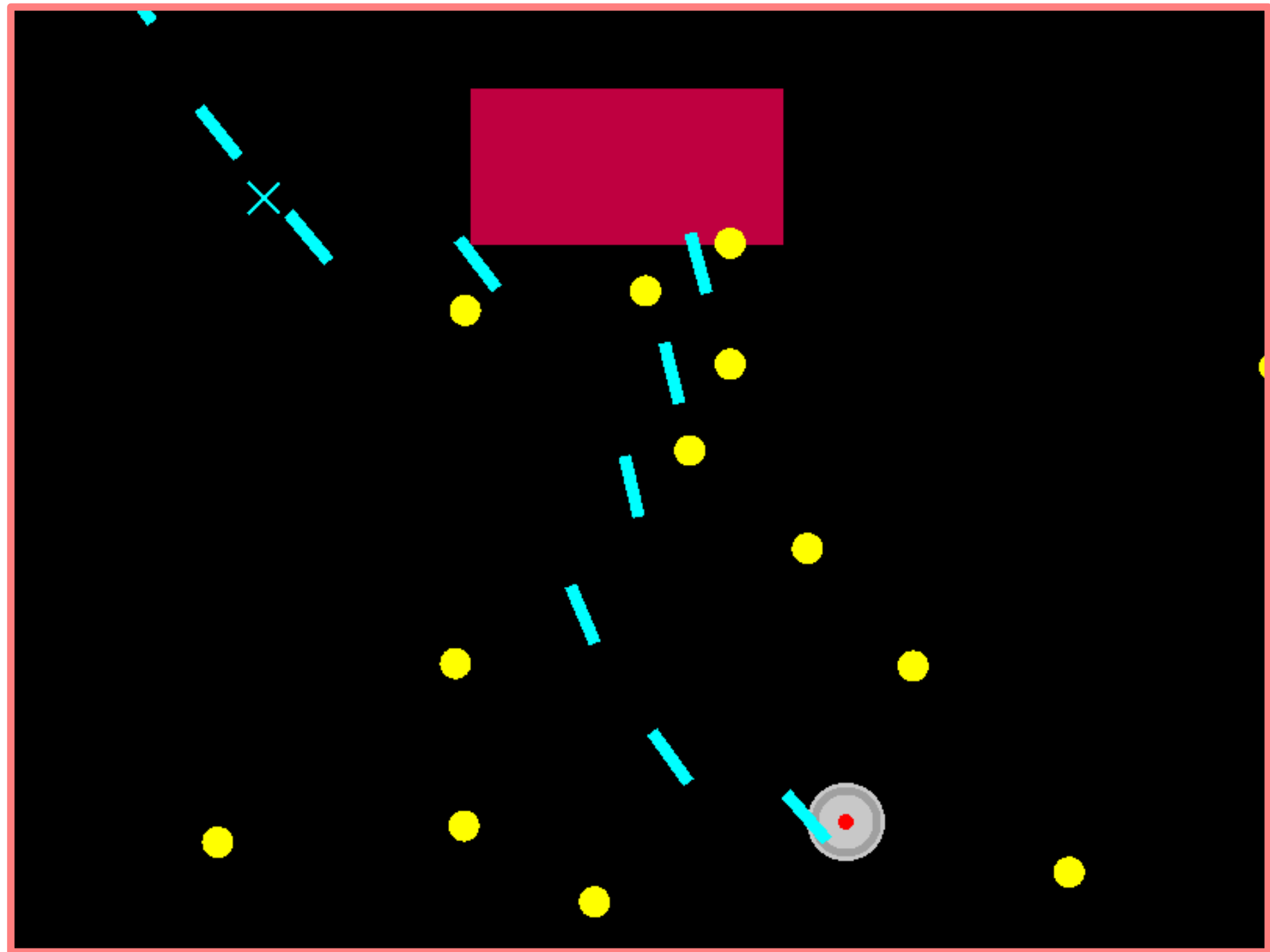


High Level

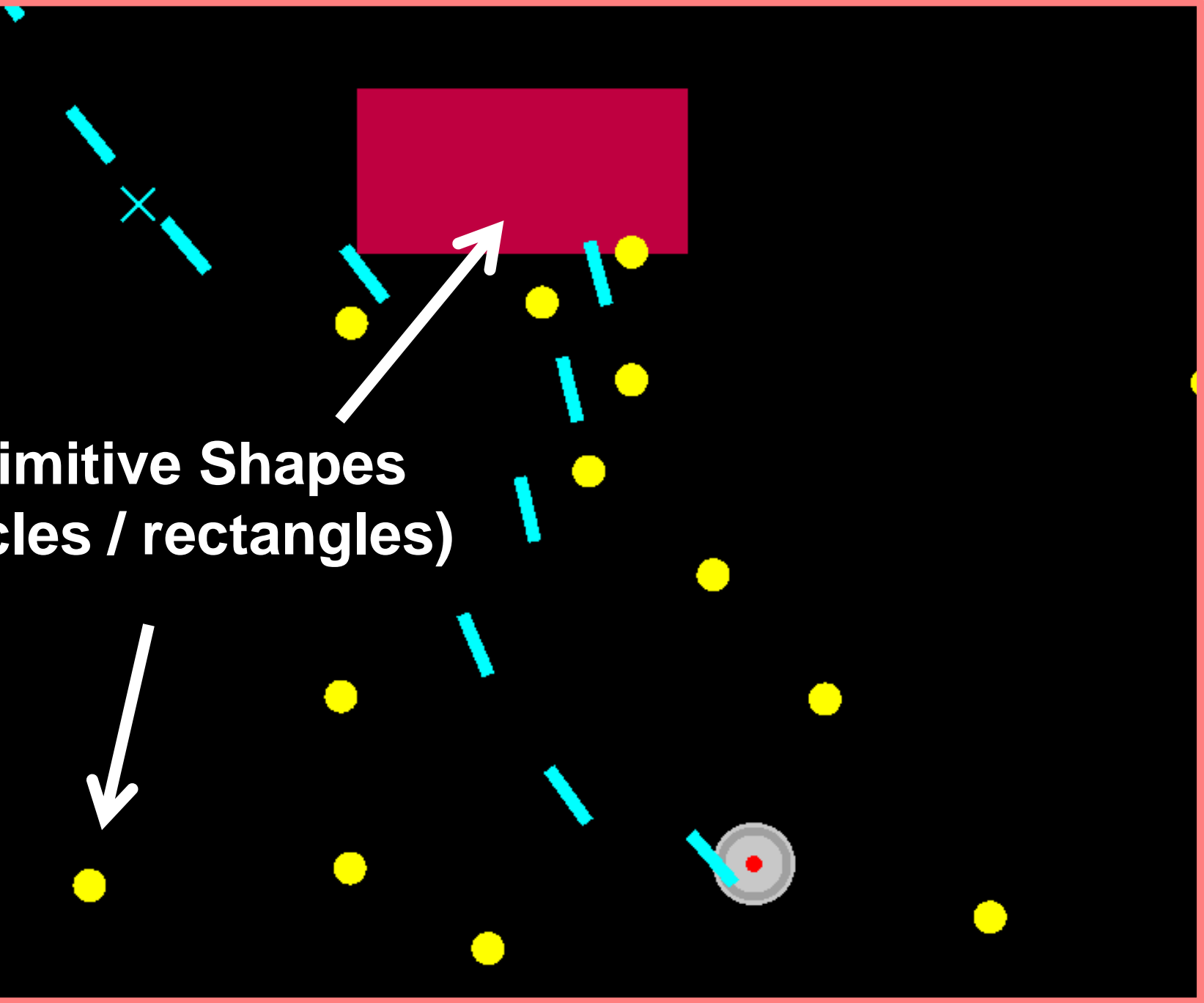


Low Level





**Primitive Shapes
(circles / rectangles)**





The diagram is set against a black background with a red border. It features several yellow circles of varying sizes scattered across the scene. A large red rectangle is positioned in the upper-middle section. Several cyan dashed lines, each with a small cyan 'x' at its midpoint, are drawn across the scene at various angles. Three white arrows point from text labels to specific elements: one points from the text 'Primitive Shapes (circles / rectangles)' to a yellow circle in the lower-left; another points from the text 'Manual collision detection' to a yellow circle near the red rectangle; and a third points from the same text to the red rectangle itself. In the lower-right area, there is a target symbol consisting of a red dot in the center, surrounded by two concentric gray circles, with a cyan dashed line passing through it.

**Primitive Shapes
(circles / rectangles)**

**Manual collision
detection**

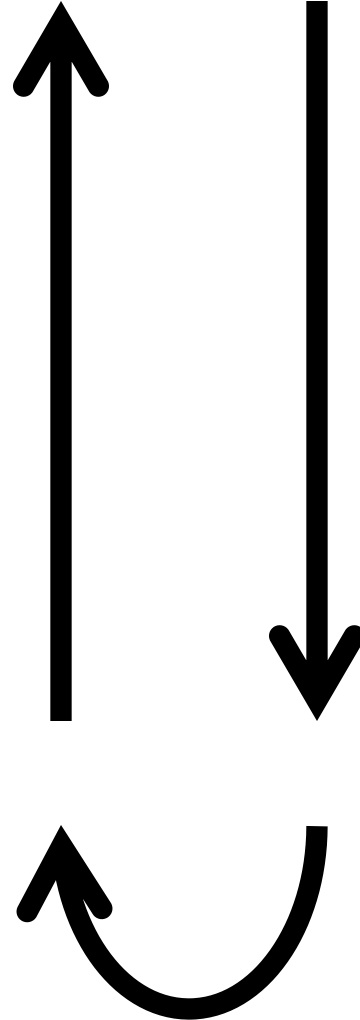
Display



DISPLAY

Infinite while loop

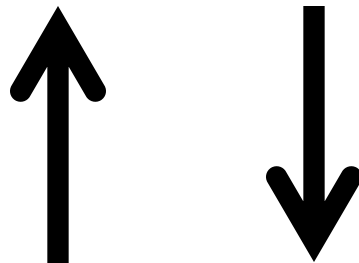
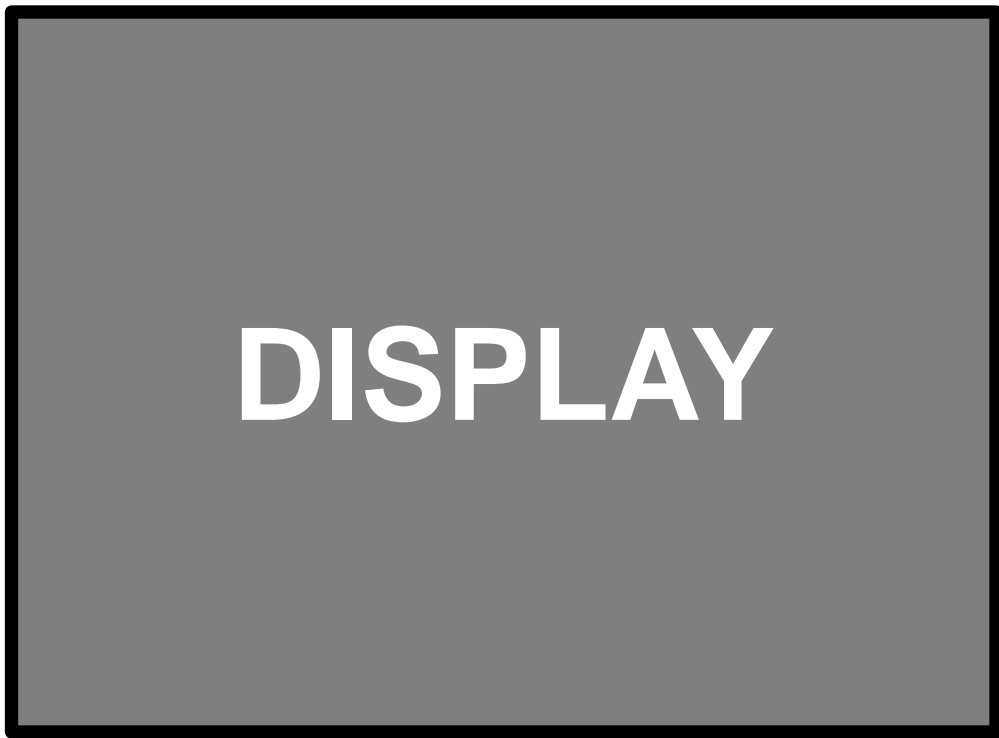
DISPLAY



Infinite while loop

Read events

DISPLAY

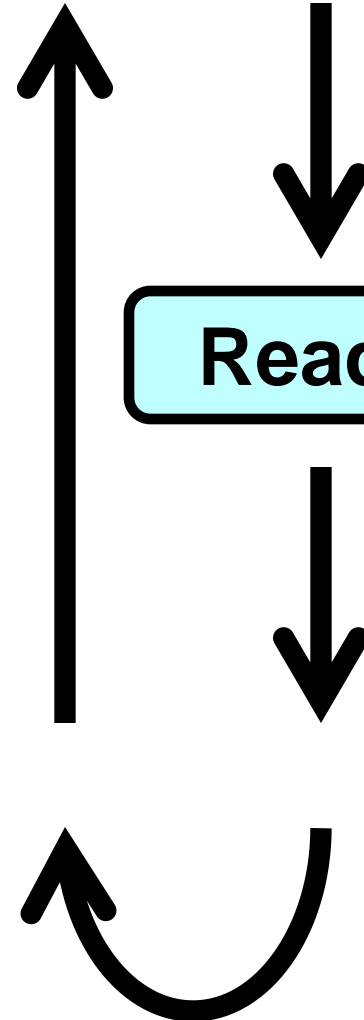
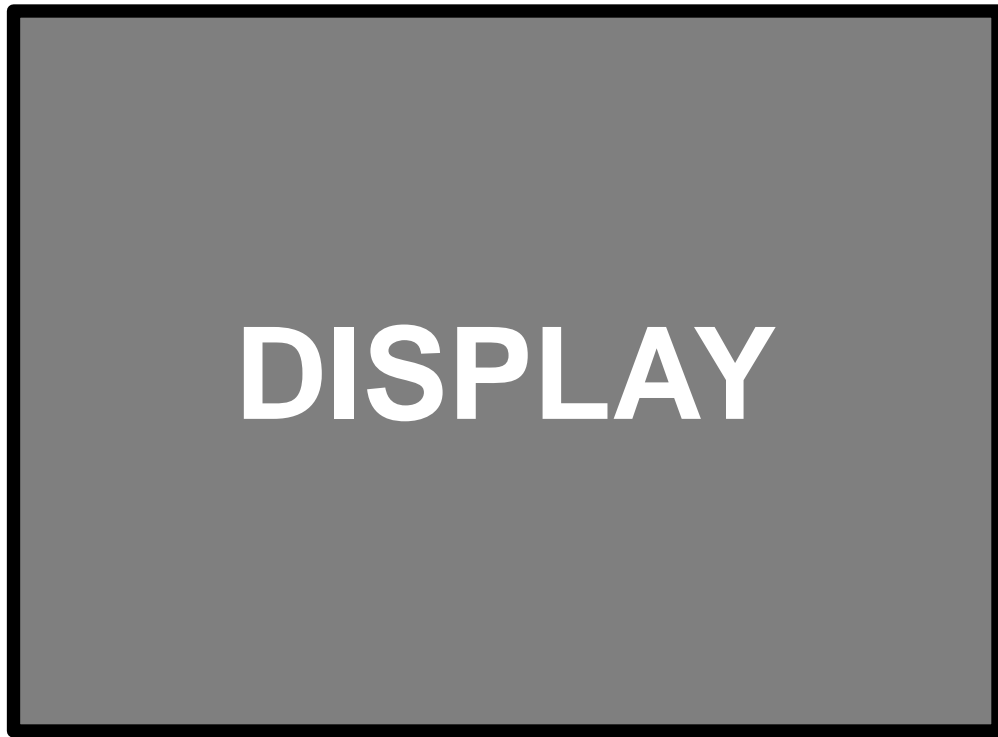


Infinite while loop

Read events

DISPLAY

CLOSE

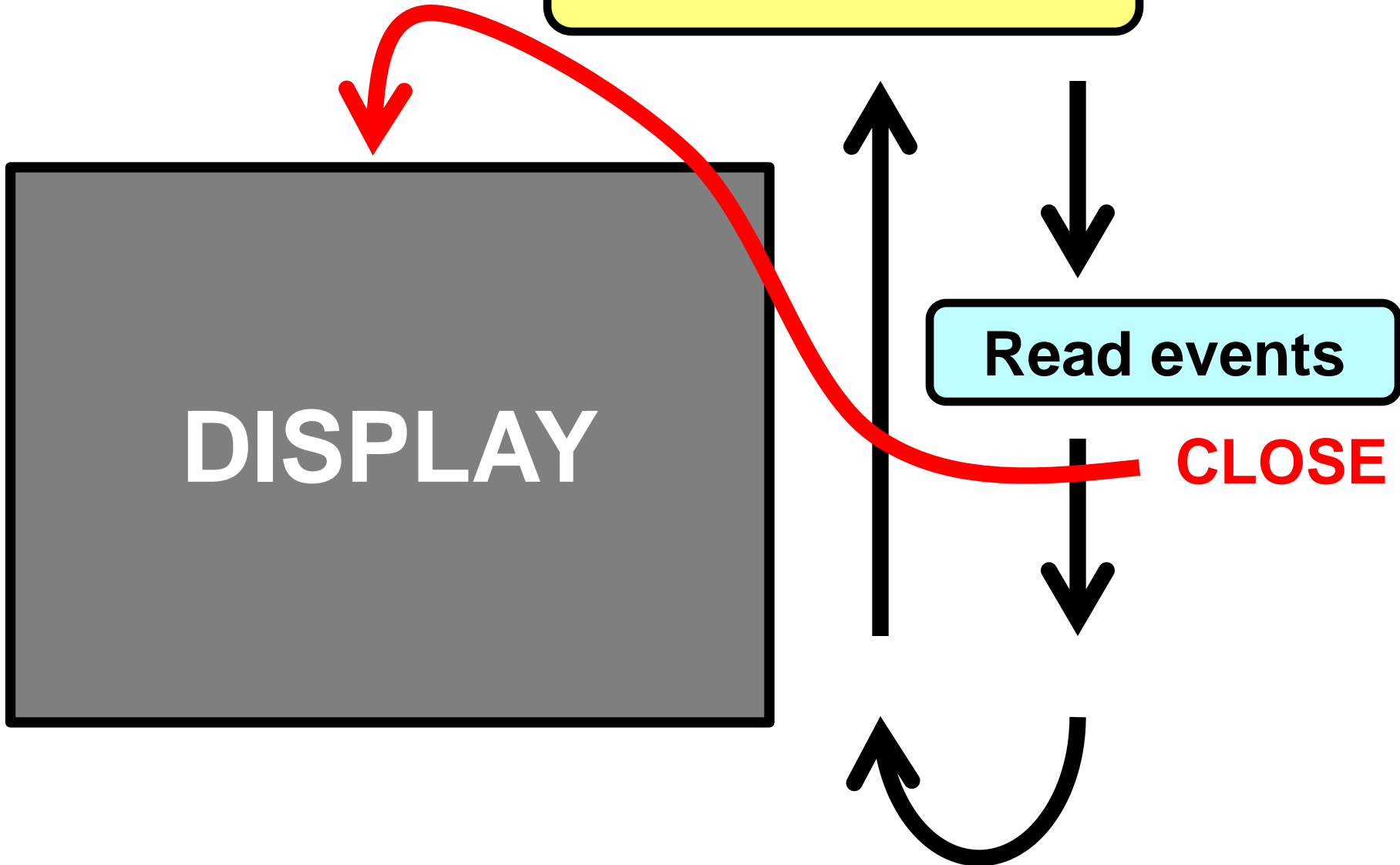


Infinite while loop

Read events

DISPLAY

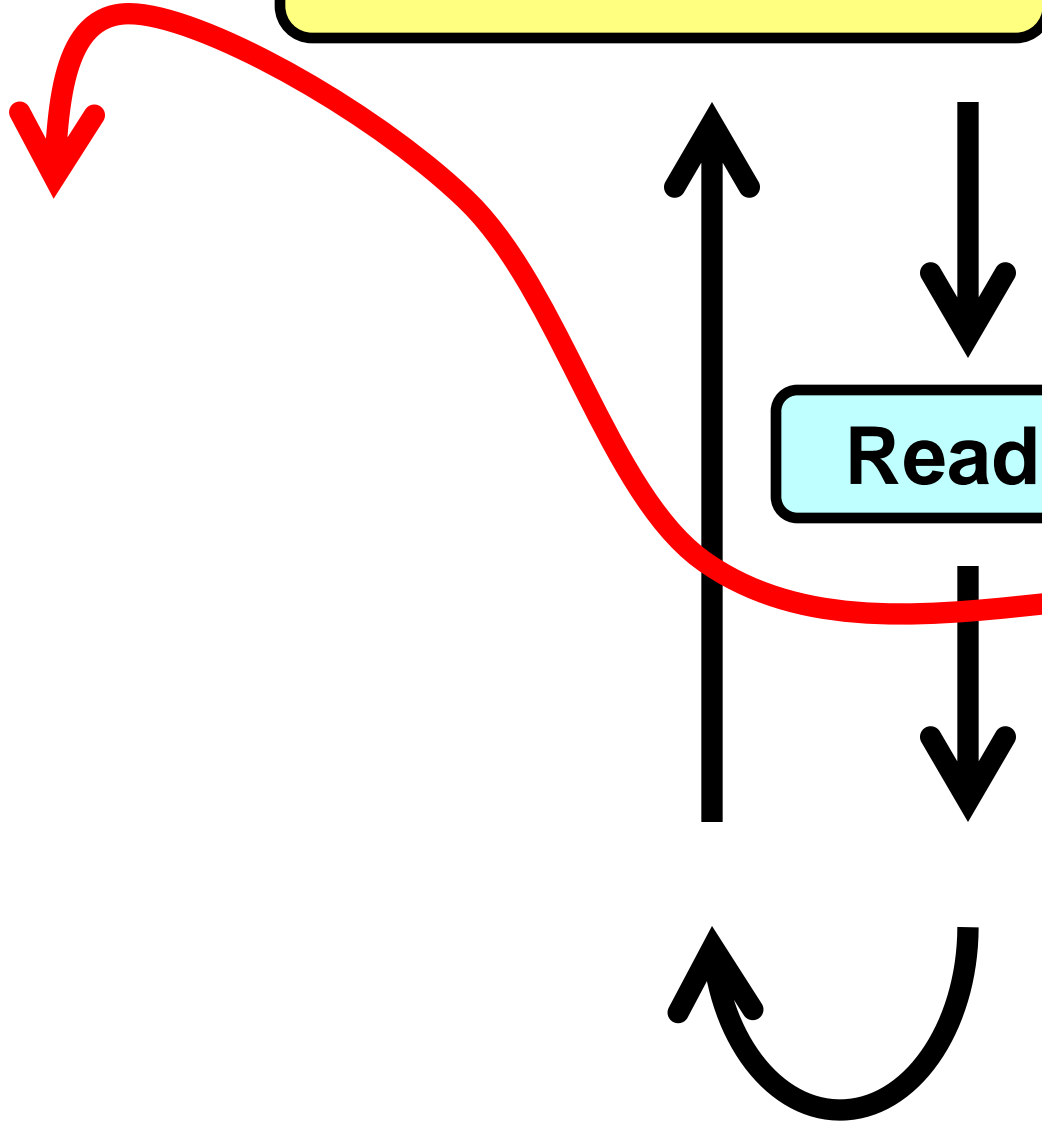
CLOSE



Infinite while loop

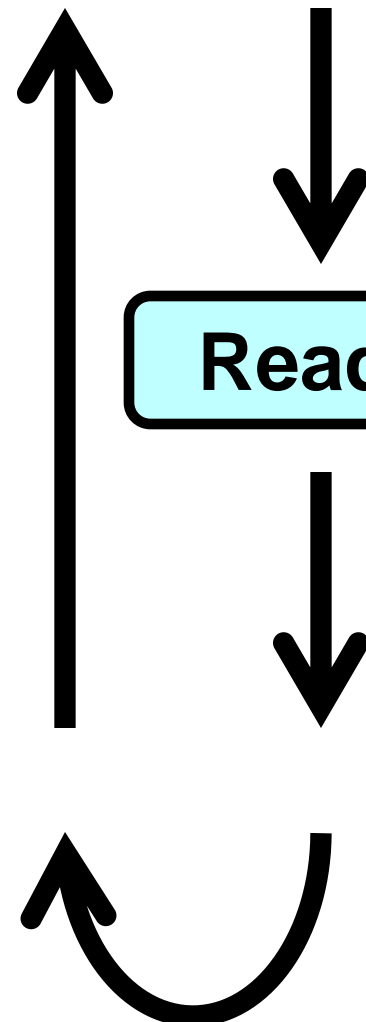
Read events

CLOSE



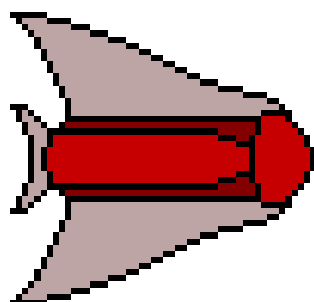
Infinite while loop

Read events

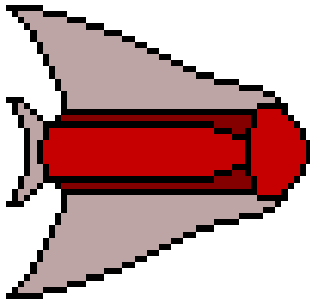


V0

Display



$x = 3, y = 5$



$x = 3, y = 5$



Update Frame

$x = 3, y = 5$



$x = 4, y = 5$

Update frame



$x = 5, y = 5$

Update frame



$x = 6, y = 5$

Update frame

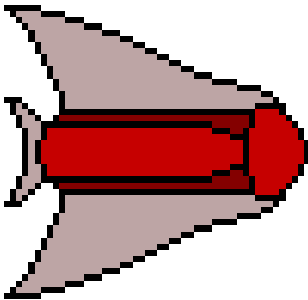


Draw Frame

$x = 3, y = 5$



$x = 3, y = 5$



Draw frame

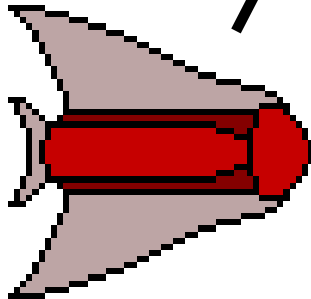
$x = 4, y = 5$

Update frame



$x = 4, y = 5$

Update frame



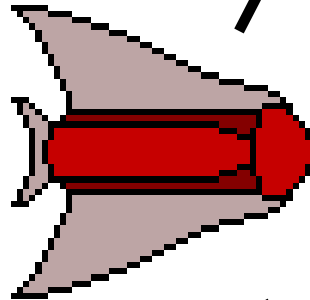
Draw frame

$x = 5, y = 5$

Update frame



$x = 5, y = 5$



Update frame

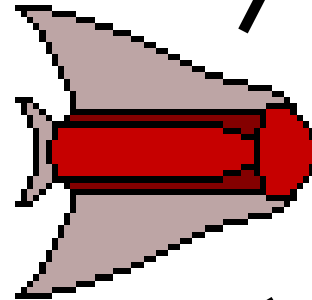
Draw frame

$x = 6, y = 5$

Update frame



$x = 6, y = 5$



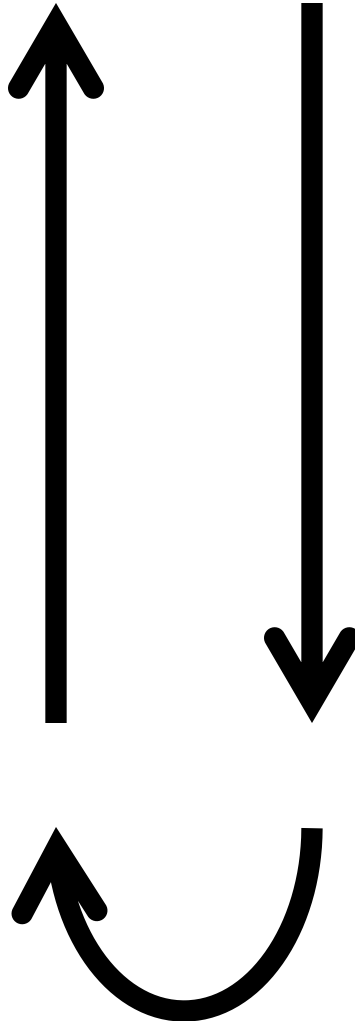
Update frame

Draw frame

Structure

Infinite while loop

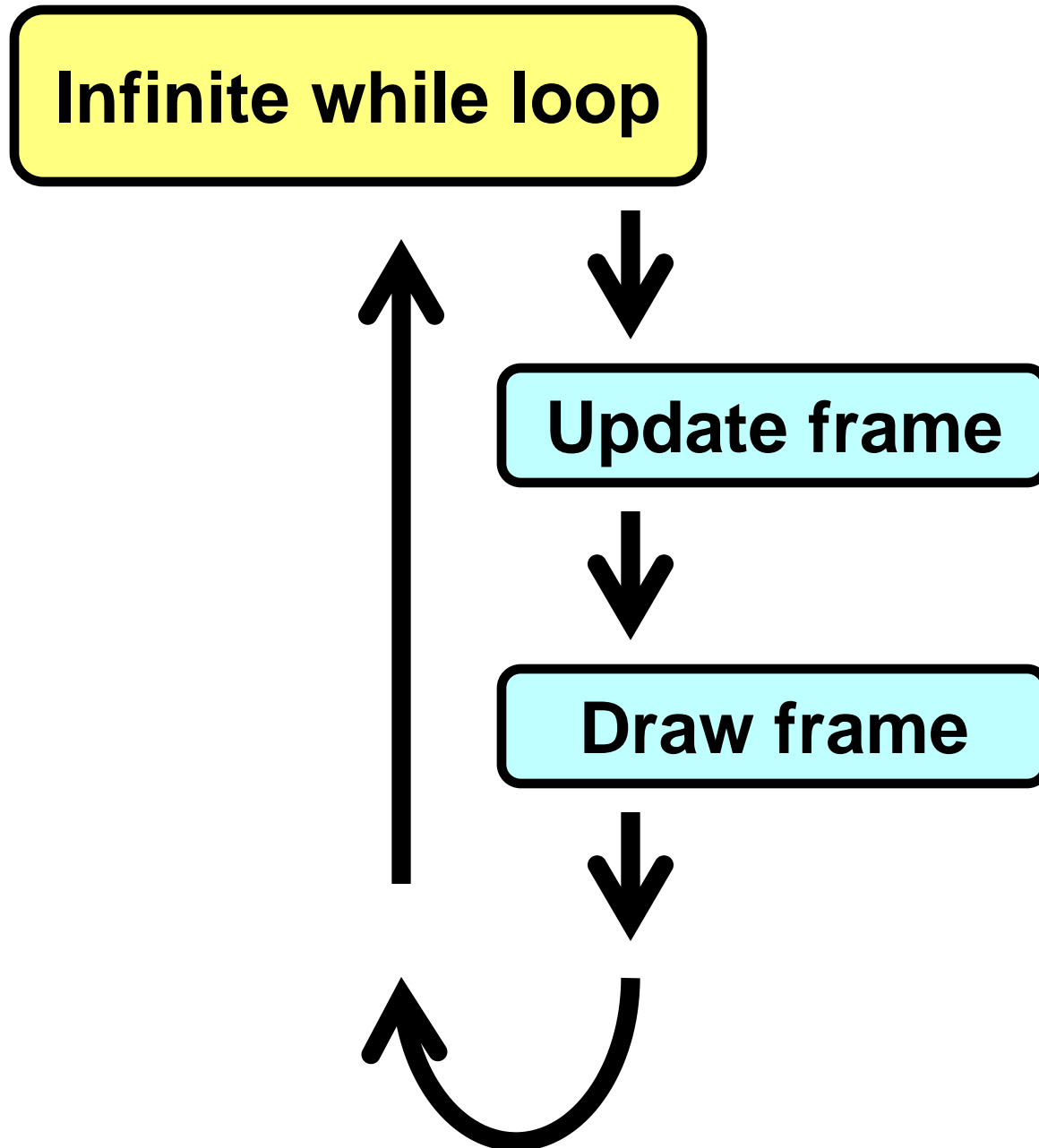
Infinite while loop



Infinite while loop

Update frame

Draw frame



Infinite while loop

```
graph TD; A[Infinite while loop] --> B[Read events]; B --> C[Update frame]; C --> D[Draw frame]; D --> A;
```

The diagram illustrates a continuous loop process. It begins with a yellow box labeled 'Infinite while loop'. An arrow points down to a light blue box 'Read events', followed by another arrow to 'Update frame', and a third to 'Draw frame'. A long vertical arrow points from the 'Draw frame' box back up to the 'Infinite while loop' box, and a curved arrow at the bottom completes the loop back to the start.

Read events

Update frame

Draw frame

V1

Update and Draw frames

Infinite while loop

```
graph TD; A[Infinite while loop] --> B[Read events]; B --> C[Update frame]; C --> D[Draw frame]; D --> A;
```

The diagram illustrates a continuous loop process. It begins with a yellow box labeled 'Infinite while loop'. An arrow points down to a light blue box 'Read events', followed by another arrow to 'Update frame', and a third to 'Draw frame'. A long vertical arrow on the left points upwards from the bottom back to the 'Infinite while loop' box, and a curved arrow at the bottom indicates the loop's continuation.

Read events

Update frame

Draw frame



Draw



Draw



Draw



Draw

**Press K
Key**



Draw

**Press K
Key**

**Press Right-
Arrow Key**



Read 2 Events

**Press K
Key**

**Press Right-
Arrow Key**

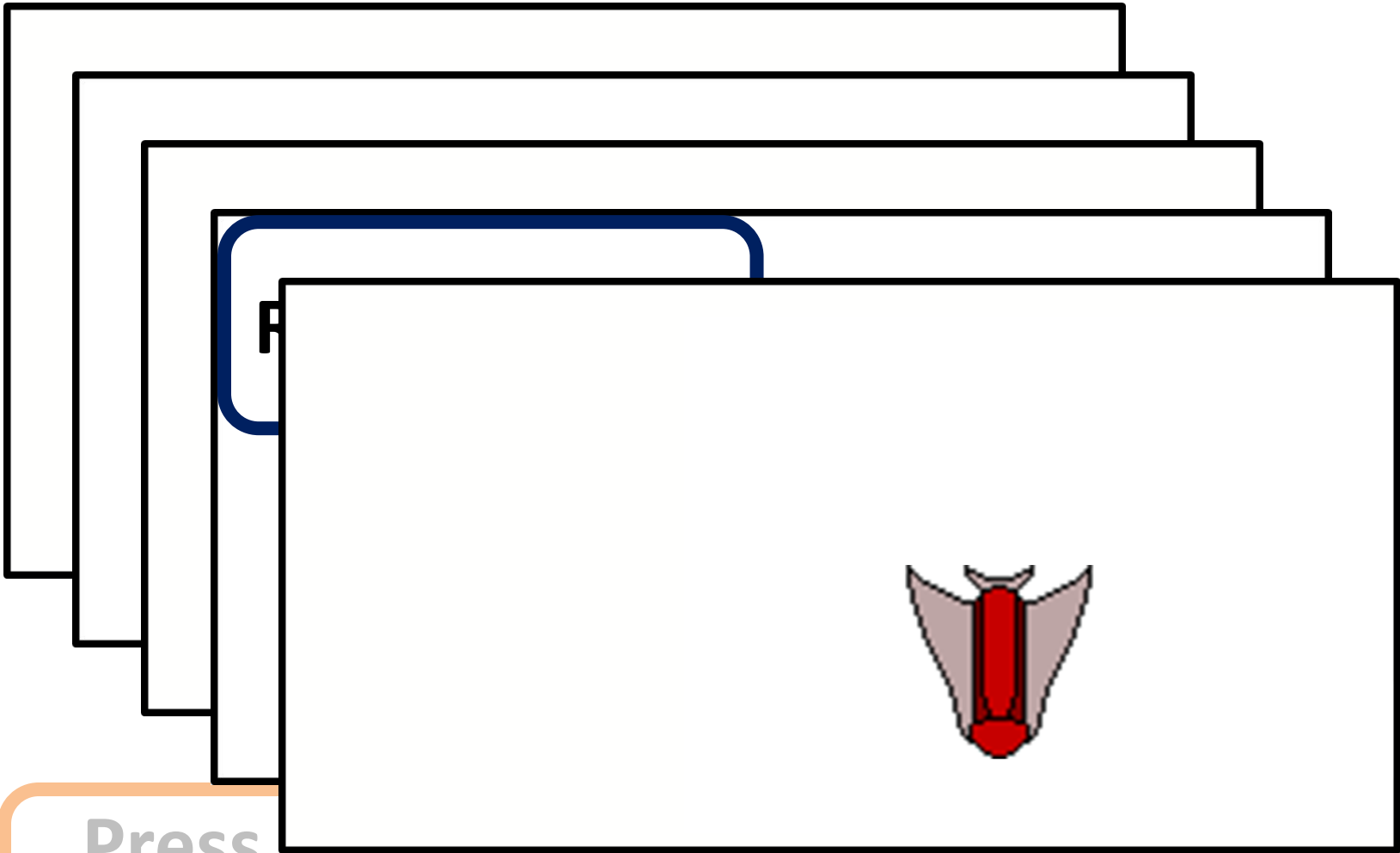
Read 2 Events



Draw

Press K
Key

Press Right-
Arrow Key

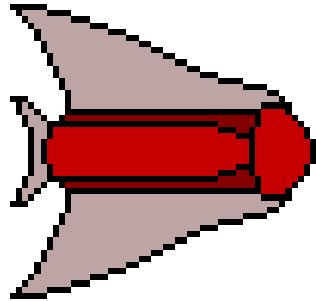


Press R
Key

Press Right-
Arrow Key

V2

Basic Keyboard Input

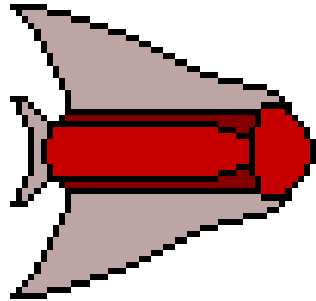


int x

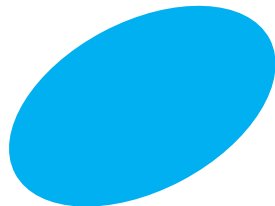
int y

int vx

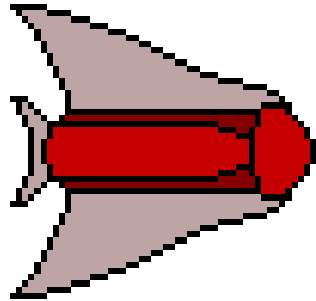
int vy



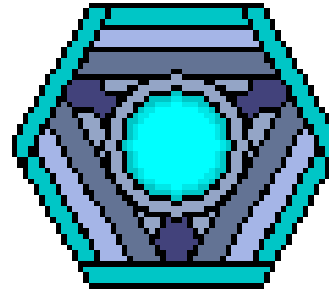
int x
int y
int vx
int vy



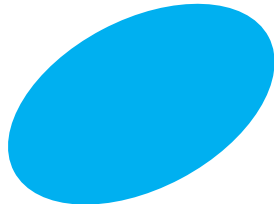
int x
int y
int vx
int vy



int x
int y
int vx
int vy



int x
int y
int vx
int vy



int x
int y
int vx
int vy

V3

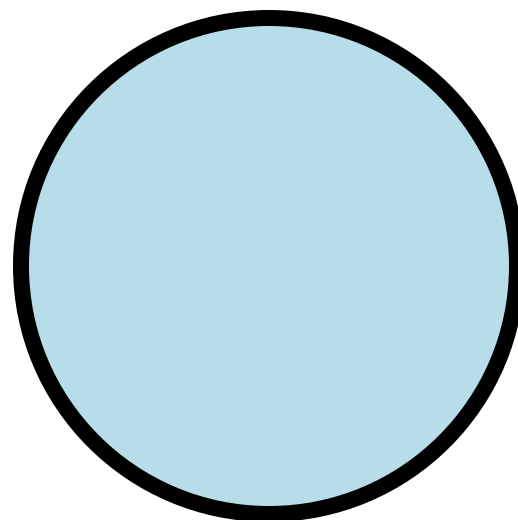
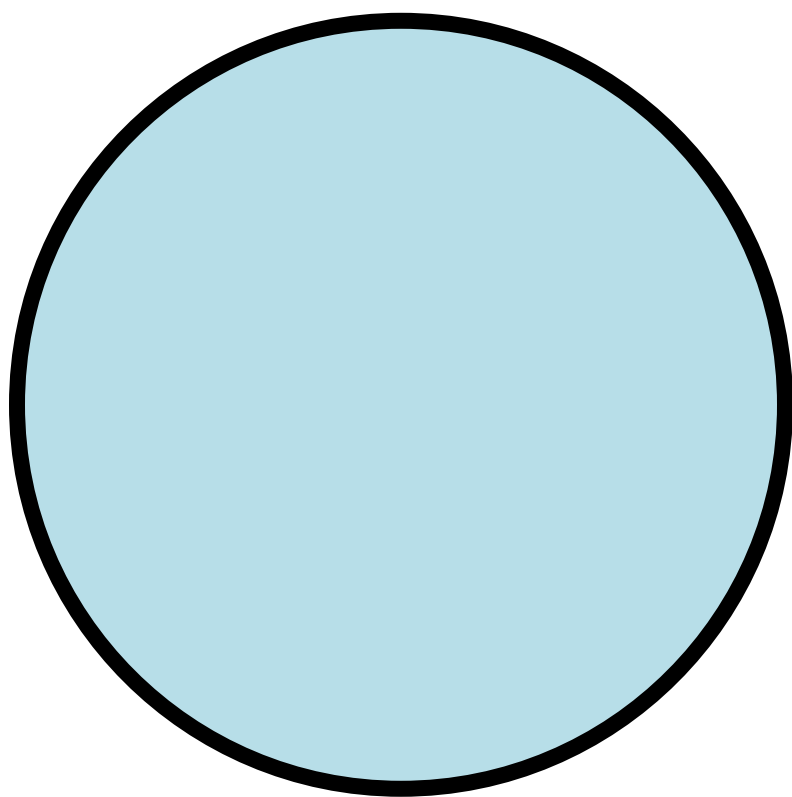
Player Class

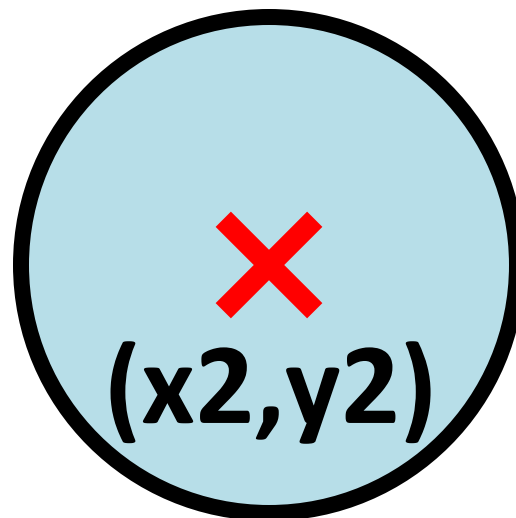
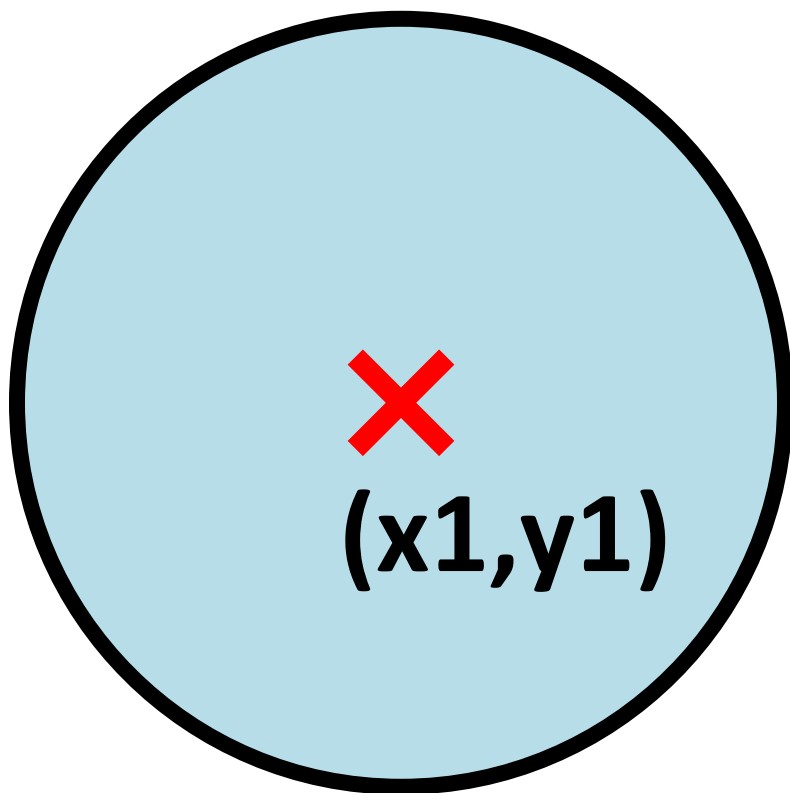
V4

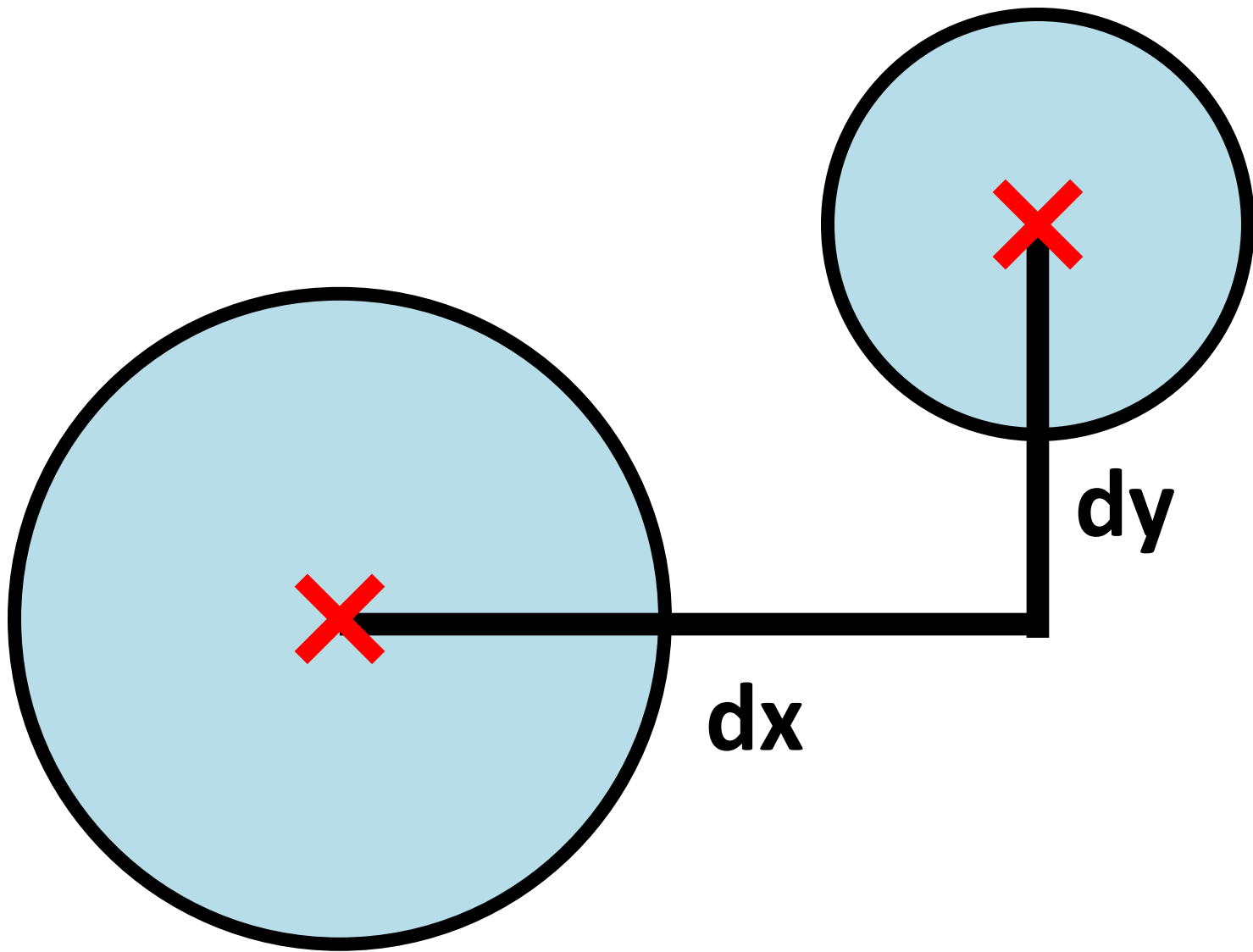
Bullets

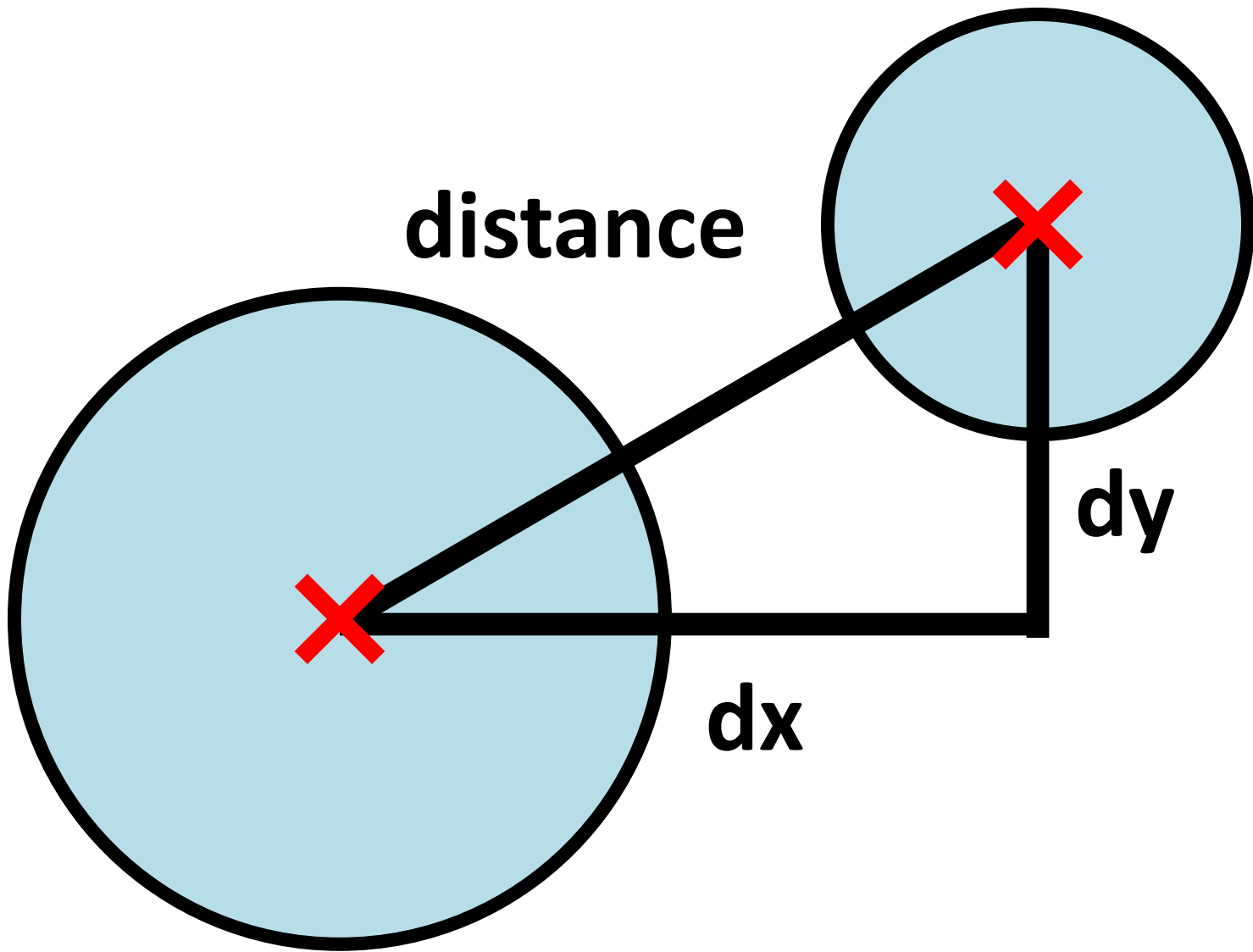
V5

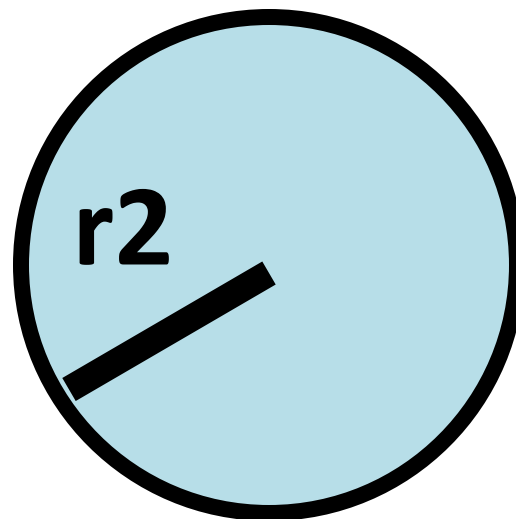
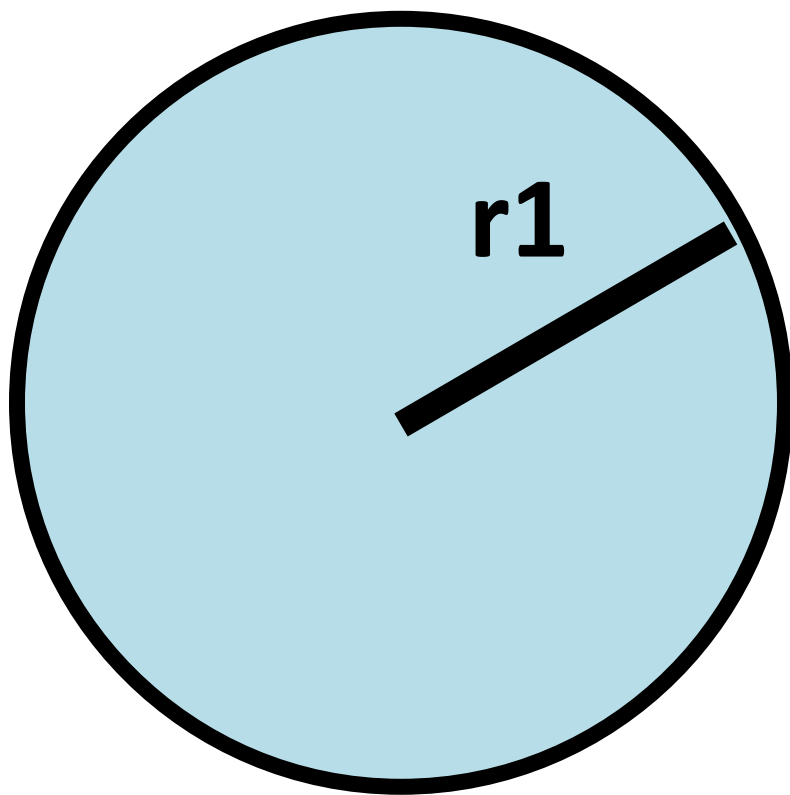
Enemies

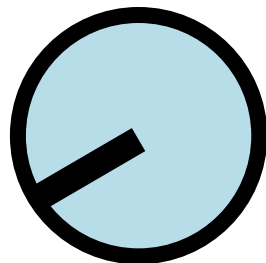
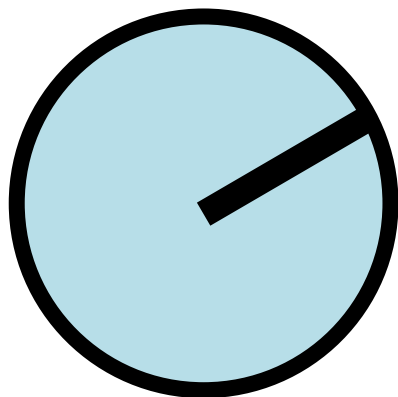


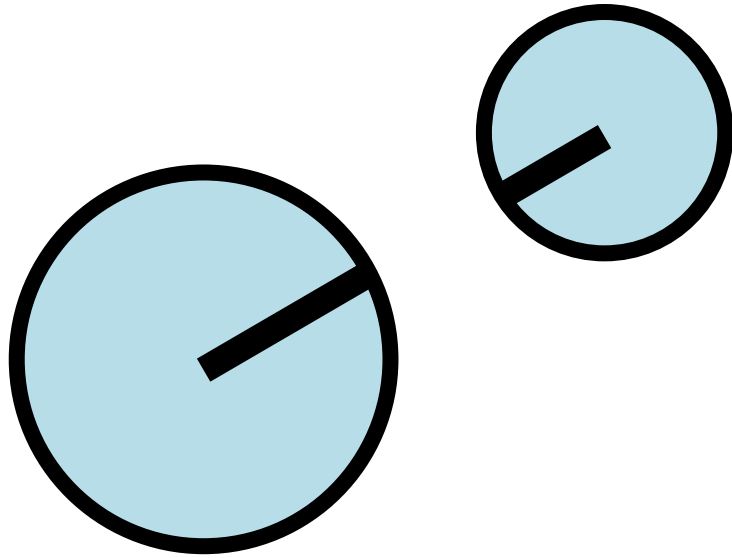




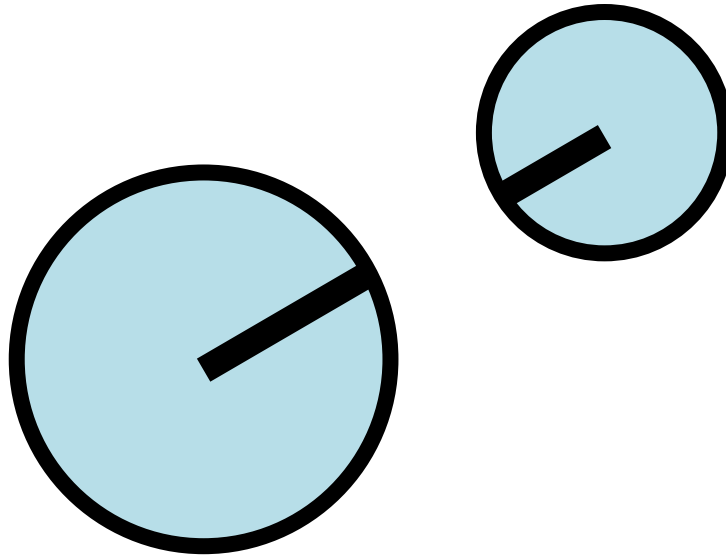






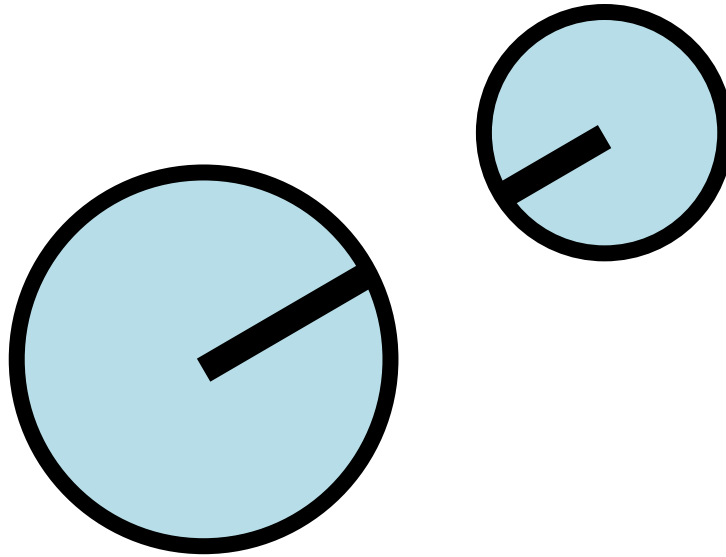


$$dx = x1 - x2$$



$$dx = x1 - x2$$

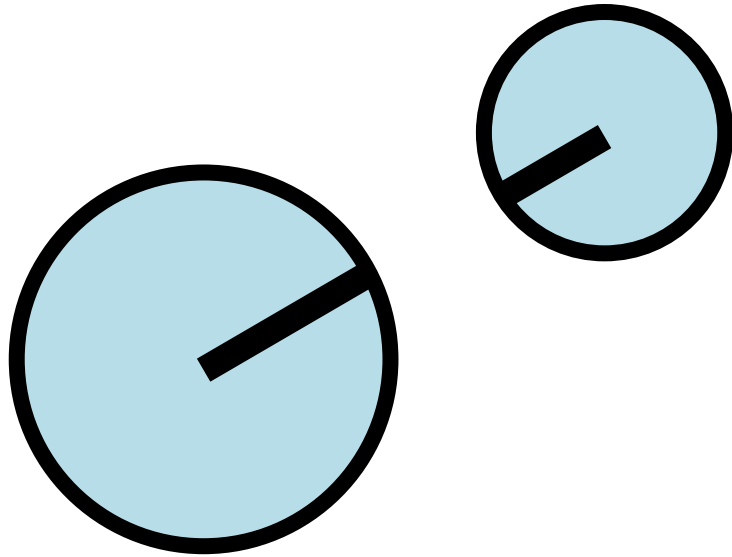
$$dy = y1 - y2$$



$$dx = x1 - x2$$

$$dy = y1 - y2$$

$$r = r1 + r2$$



$$dx = x1 - x2$$

$$dy = y1 - y2$$

$$r = r1 + r2$$

$$dx * dx + dy * dy \leq r * r$$

V6

Collision Detection

V7

Restart Game

End

ACKNOWLEDGEMENT

This presentation benefitted from

PowerPointLabs

a PowerPoint plugin for creating
better presentations with less effort.

PowerPointLabs
is available for free at
<http://PowerPointLabs.info>