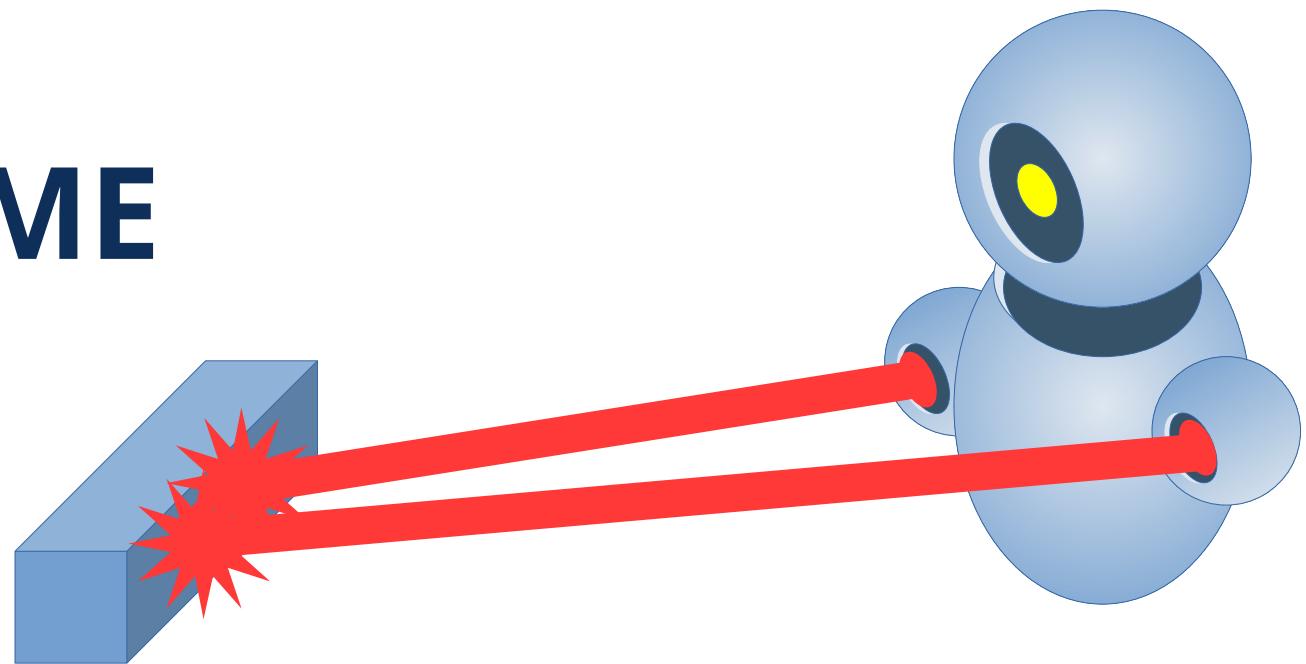
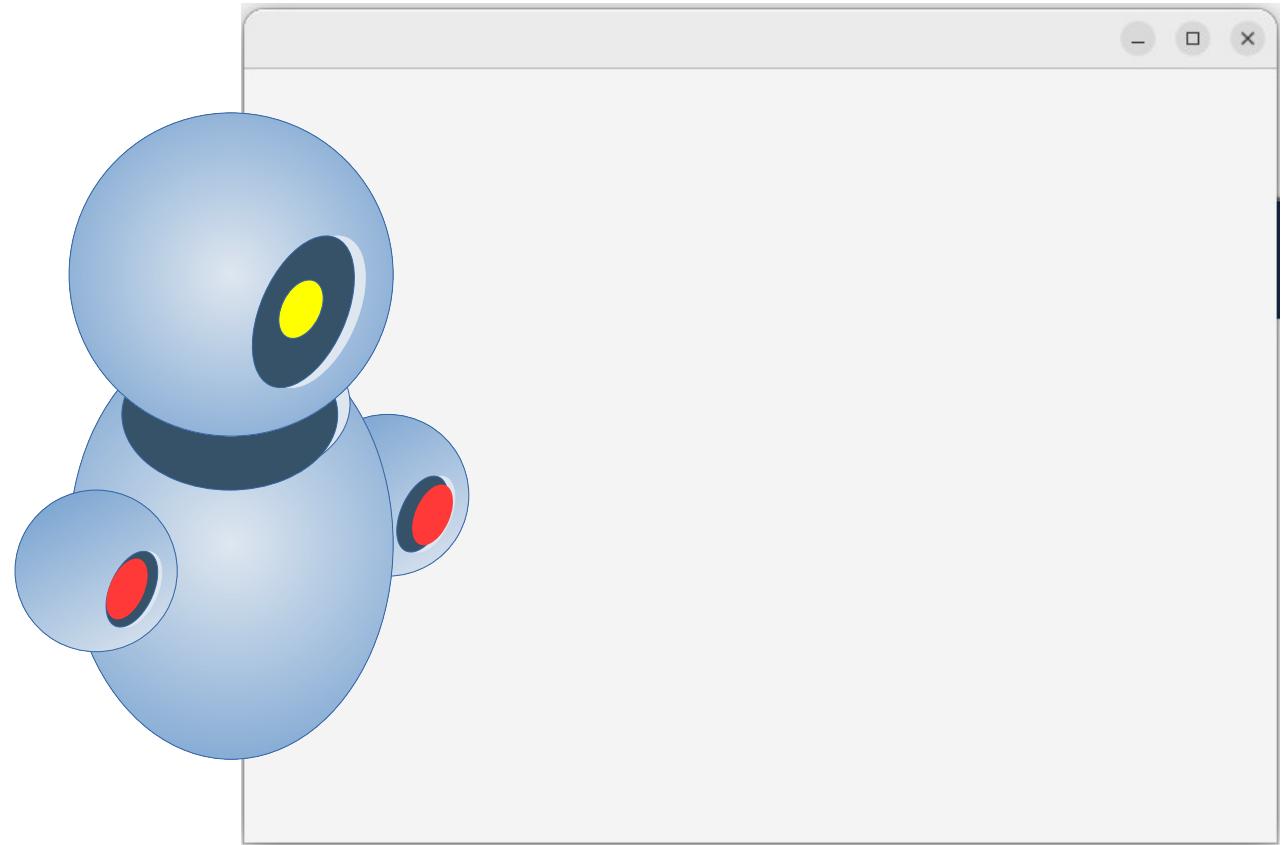


# THE CHON GAME IN JAVA

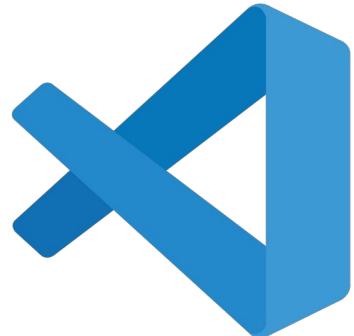


# MY FIRST JAVA FX APPLICATION

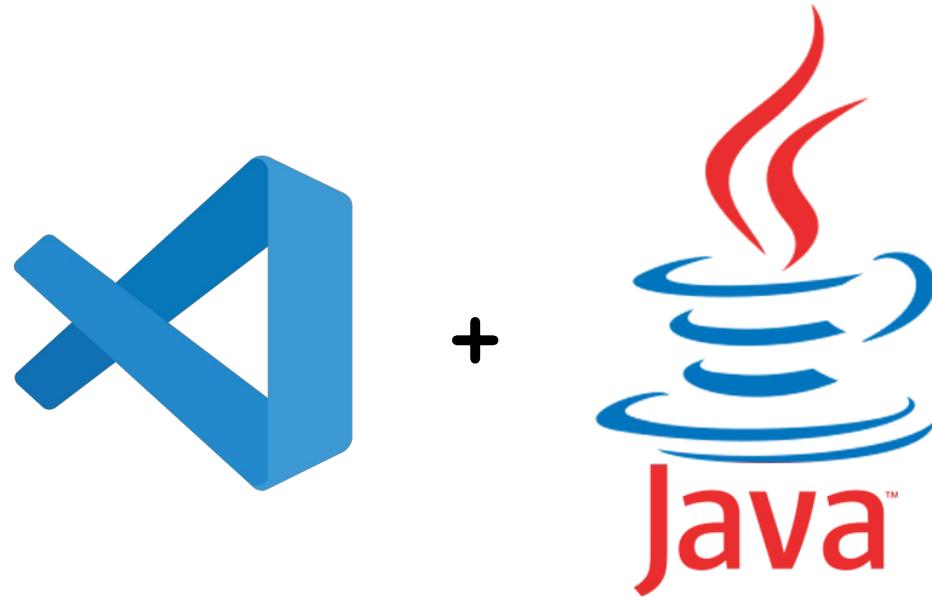


JavaFX is a comprehensive software platform for creating and programming client and cross-platform desktop applications.

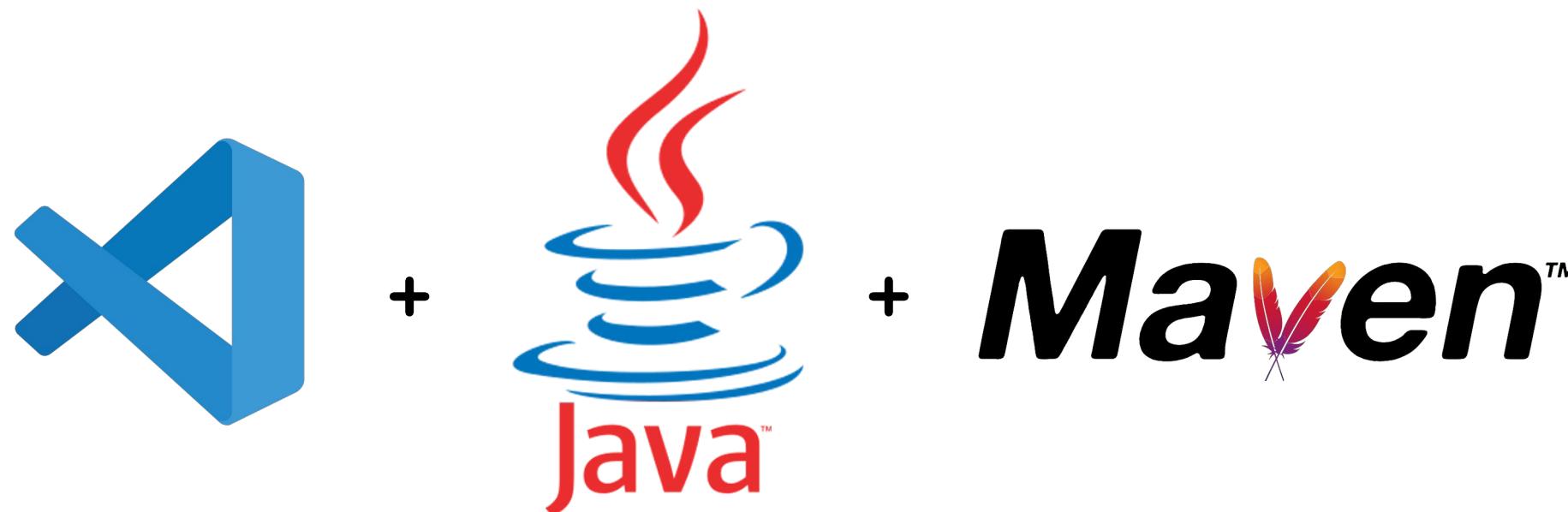
# Project Technologies



# Project Technologies

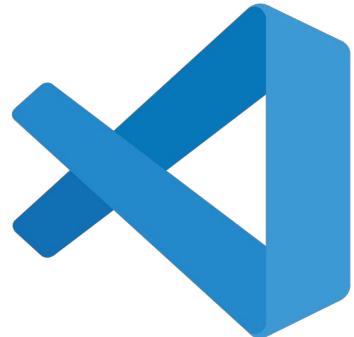


# Project Technologies

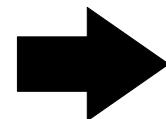
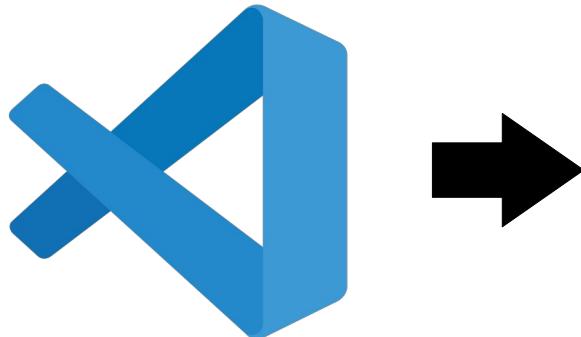


# Project Technologies





# VSCode



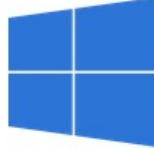
https://code.visualstudio.com/download

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Version 1.87 is now available! Read about the new features and fixes from February.

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

 Windows  
Windows 10, 11

 .deb  
Debian, Ubuntu

 .rpm  
Red Hat, Fedora, SUSE

 Mac  
macOS 10.15+

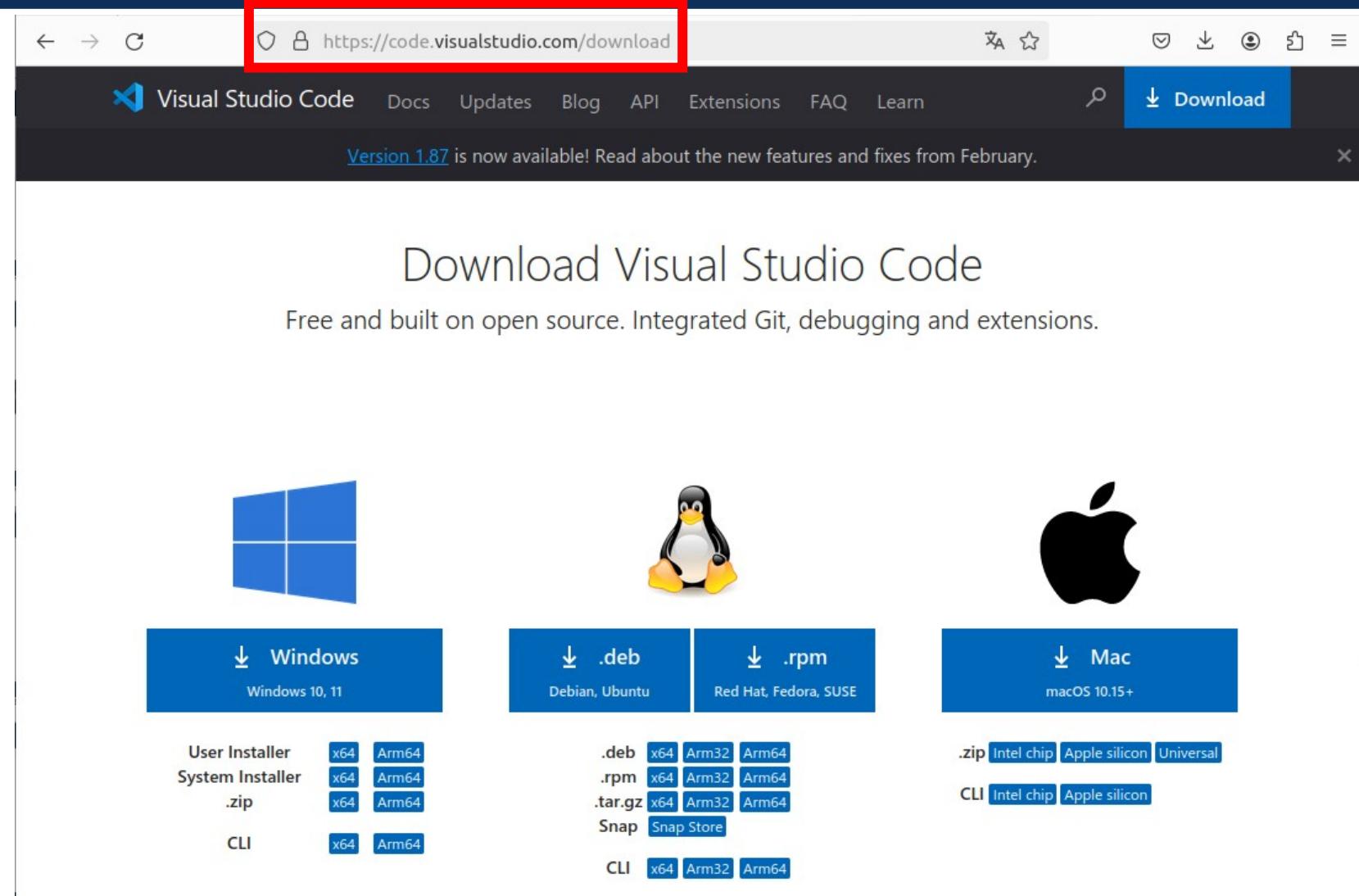
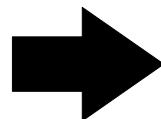
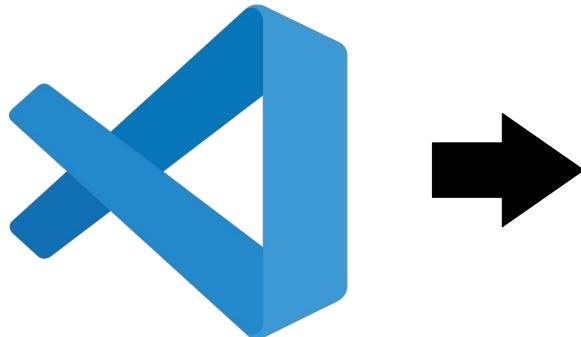
User Installer  
System Installer  
.zip  
CLI

.deb  
.rpm  
.tar.gz  
Snap  
CLI

x64  
Arm64  
x64  
Arm64  
x64  
Arm64  
x64  
Arm32  
Arm64  
x64  
Arm32  
Arm64  
Snap Store  
x64  
Arm32  
Arm64

.zip  
Intel chip  
Apple silicon  
Universal  
CLI  
Intel chip  
Apple silicon

# VSCode



The screenshot shows the official Visual Studio Code download page at <https://code.visualstudio.com/download>. The page features a dark header with the Visual Studio Code logo and navigation links for Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A prominent blue 'Download' button is located in the top right. A banner at the top of the main content area announces 'Version 1.87 is now available! Read about the new features and fixes from February.' Below the banner, the title 'Download Visual Studio Code' is displayed, followed by the subtitle 'Free and built on open source. Integrated Git, debugging and extensions.' Three operating system icons are shown: Windows (blue square), Linux (Tux the Penguin), and macOS (apple). Below each icon are download buttons for Windows, .deb, .rpm, and Mac respectively. Each download section also lists alternative installer formats and supported architectures (x64, Arm32, Arm64).

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Windows

.deb

.rpm

Mac

User Installer  
System Installer  
.zip  
CLI

.deb  
.rpm  
.tar.gz  
Snap  
CLI

x64  
Arm32  
Arm64

x64  
Arm32  
Arm64

x64  
Arm32  
Arm64

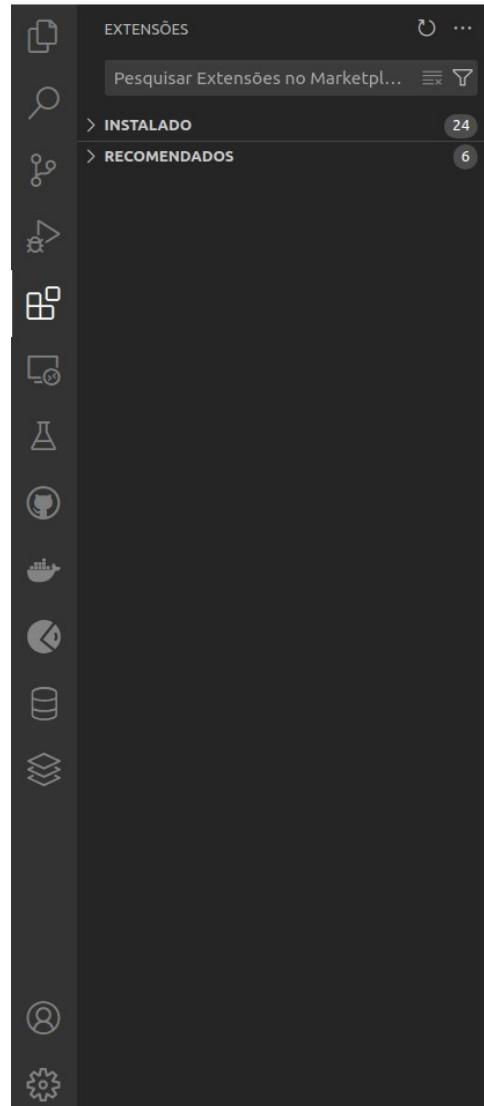
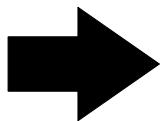
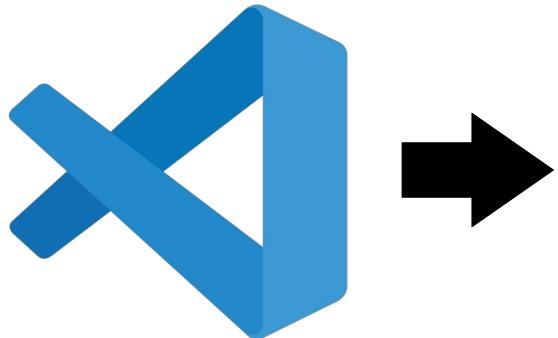
Snap Store

x64  
Arm32  
Arm64

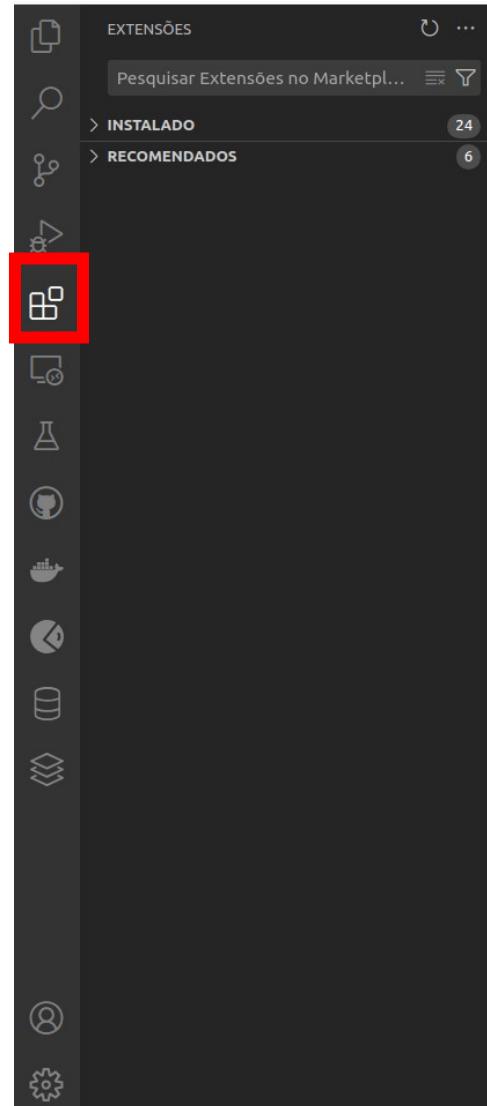
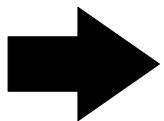
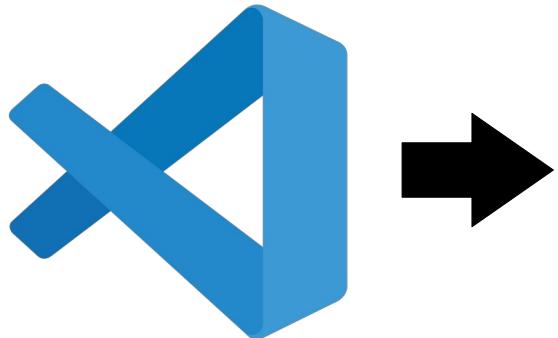
.zip  
Intel chip  
Apple silicon  
Universal

CLI  
Intel chip  
Apple silicon

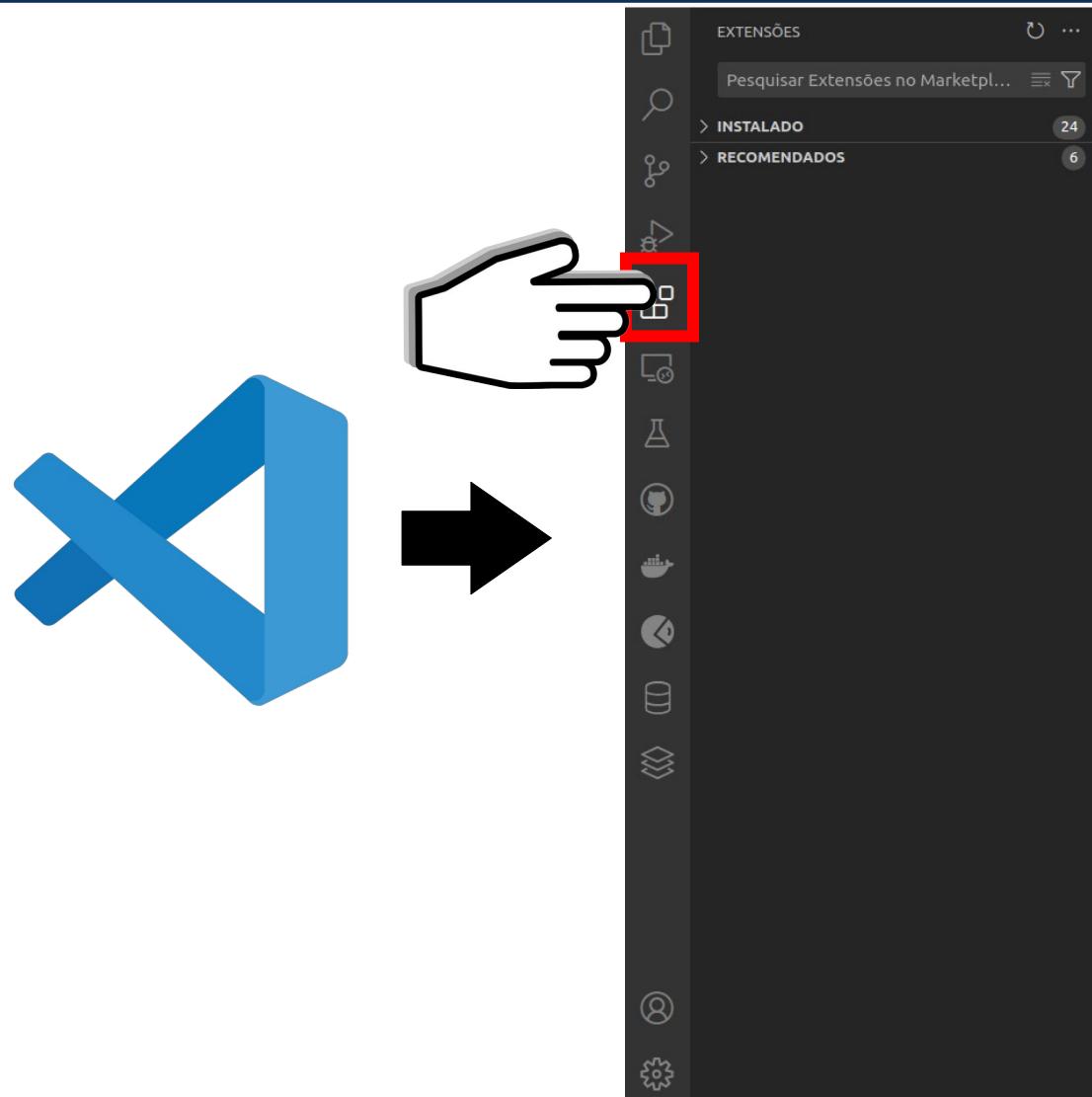
# VSCode



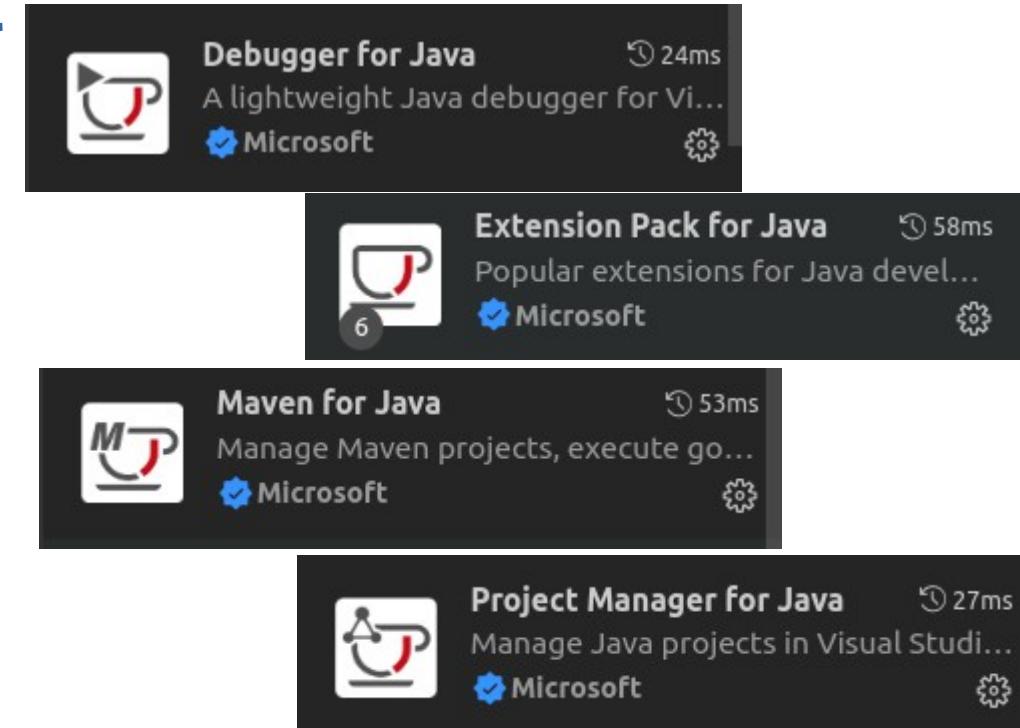
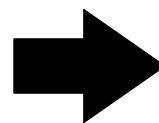
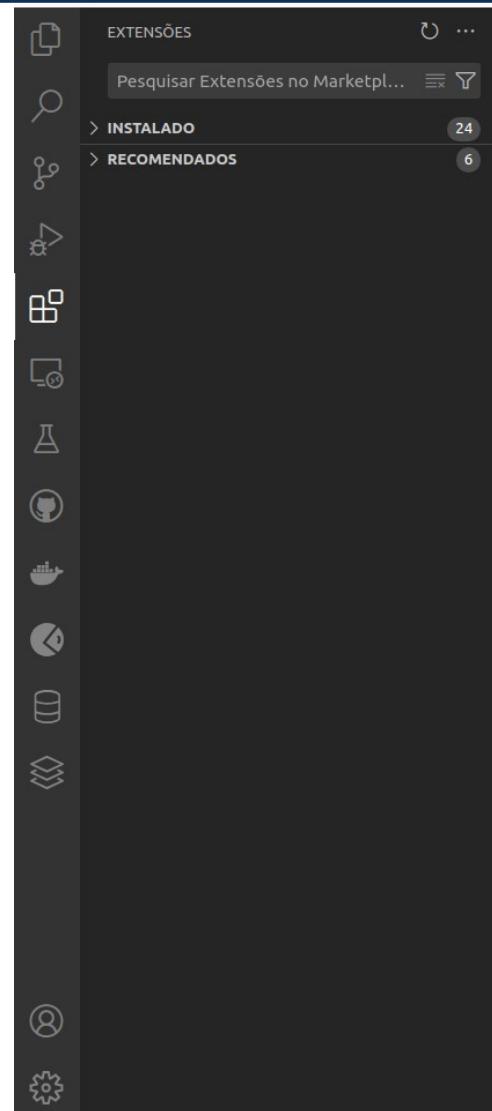
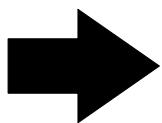
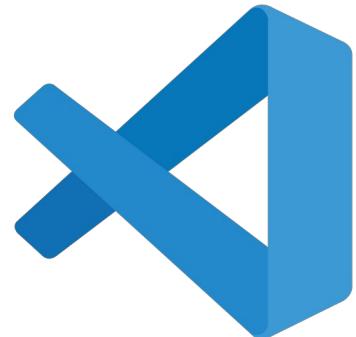
# VSCode



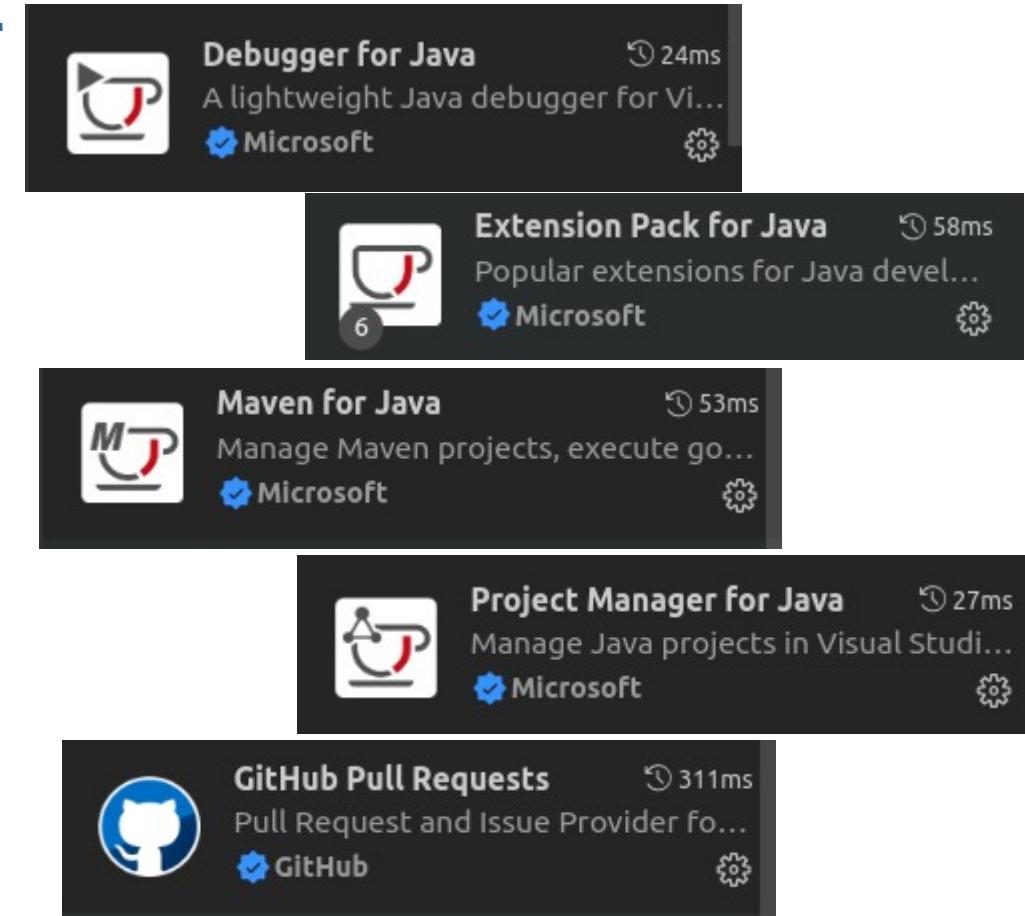
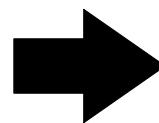
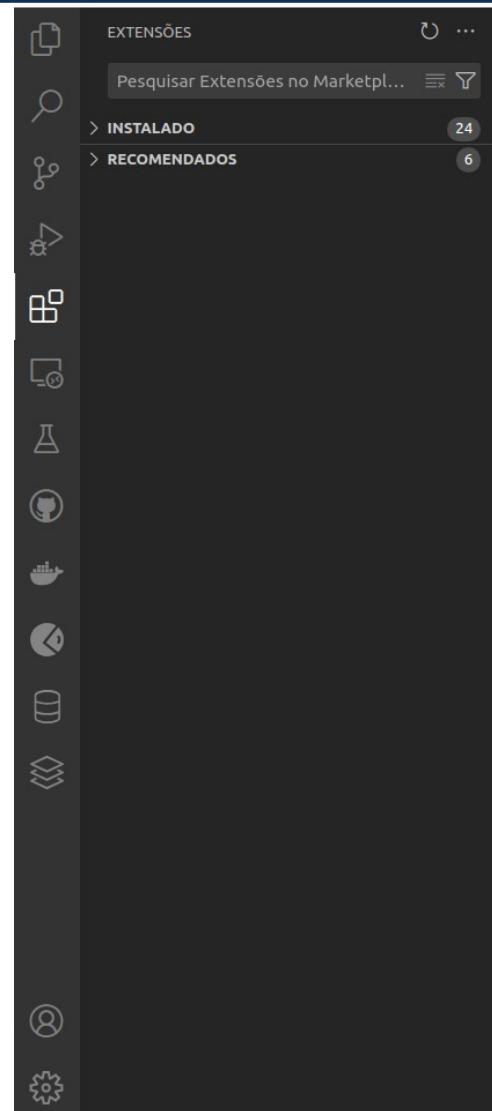
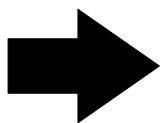
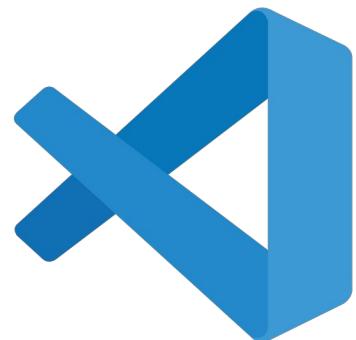
# VSCode



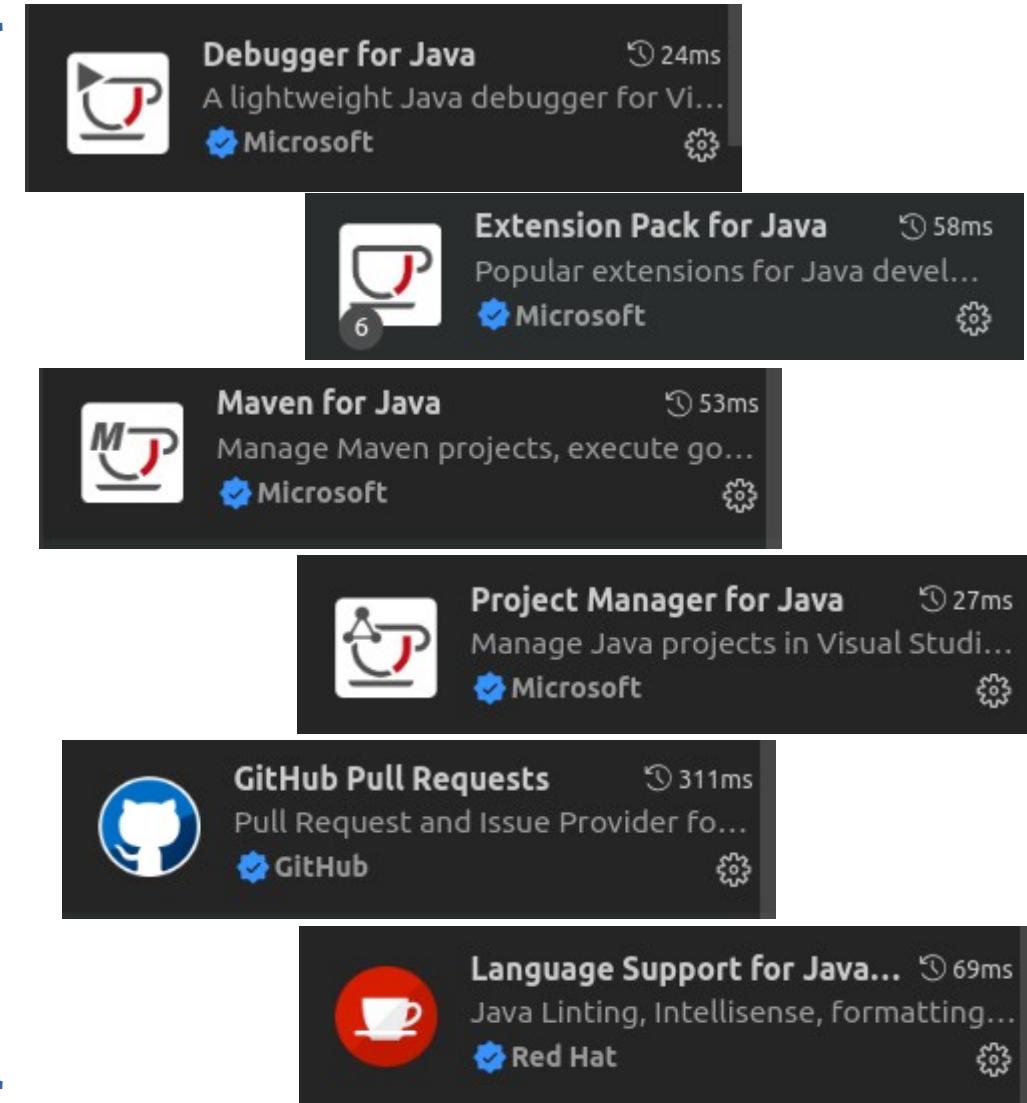
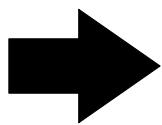
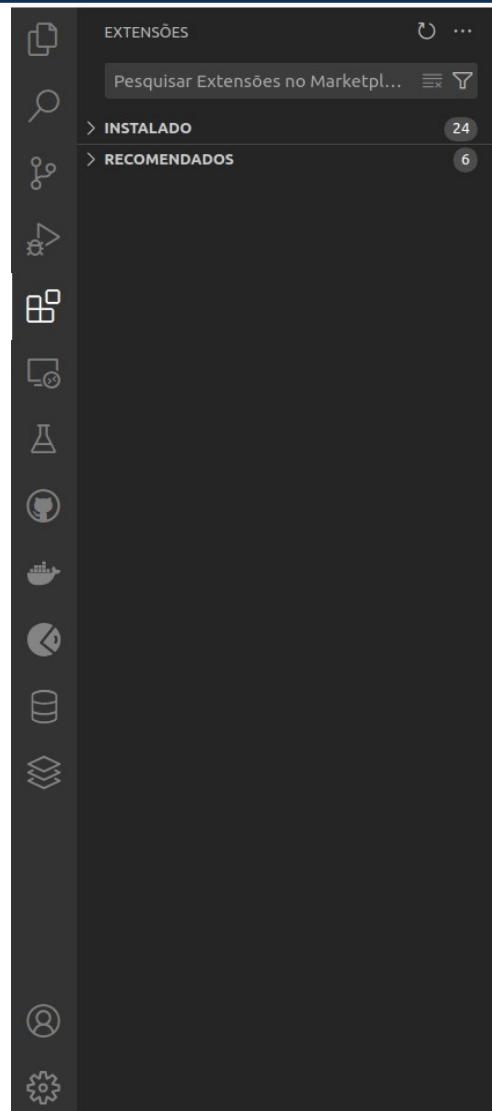
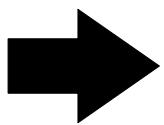
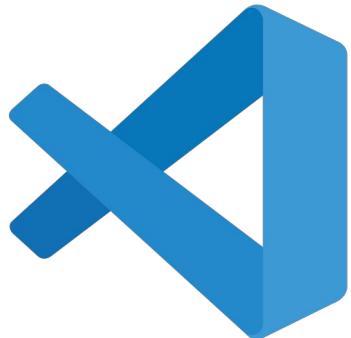
# VSCode



# VSCode

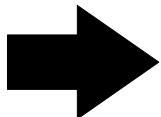


# VSCode





# GitHub



The screenshot shows a GitHub profile page for the user 'profpantoja'. At the top, there's a navigation bar with links for Overview, Repositories (10), Projects, Packages, Stars (2), and a search bar. Below the navigation is a circular profile picture of a man with rainbow hair, holding a whiteboard marker. The main content area starts with a bio section: 'Carpe diem. Make your lives extraordinary.' followed by a paragraph about the user's work in Artificial Intelligence, Robotics, Ubiquitous Computing, and the Internet of Things. It also lists current projects and collaborations. Below this is a 'Research Groups and Projects' section listing membership in CHON, OBINVEST, and the Turing Project. The 'Languages and frameworks I've interacted' section includes Java, MySQL, and other icons. At the bottom, there's a 'Follow me on Social Media' section with links to LinkedIn, ResearchGate, Google Scholar, iD, and Instagram.

profpantoja / README.md

**Carpe diem. Make your lives extraordinary.**

I work in Artificial Intelligence, more specifically in Multi-agent Systems and Software Engineering, but I have ventured into the areas of Robotics, Ubiquitous Computing, and the Internet of Things. In the management area, I work in technology dissemination (and currently social media), technological innovation, entrepreneurship, and Information Technology Governance and process mapping.

- I'm currently working on [multi-agent-system](#) [embedded-system](#) [modeling-language](#) [metamodel](#)
- I'm collaborating on [JaCaMo](#) [ChonIDE](#)

**Research Groups and Projects**

- Member of Cognitive Hardware on Networks Research Group (CHON).
- Co-coordinator of the Brazilian Investment Olympics (OBINVEST).
- Founder of the Turing Project.

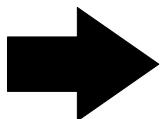
**Languages and frameworks I've interacted**

Java MySQL

**Follow me on Social Media**

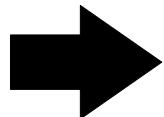
LinkedIn ResearchGate Google Scholar iD Instagram

# GitHub



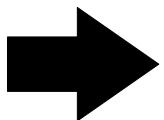
The screenshot shows a GitHub profile page for the user 'profpantoja'. At the top, there's a navigation bar with tabs: 'Overview' (selected), 'Repositories' (highlighted with a red box), 'Projects', 'Packages', and 'Stars' (2). Below the navigation bar is a circular profile picture of a man with rainbow hair, holding a whiteboard marker. The profile section includes the name 'Kadu Pantoja', the handle 'profpantoja', and a brief bio: 'Professor and researcher at CEFET/RJ, master and Ph.D in Computing at the Military Institute of Engineering (IME) and UFF with mobility at Sorbonne UPMC.' It also lists '62 followers' and '8 following'. Below the profile is a 'Repositories' section showing zero repositories. The main content area contains a 'README.md' file with the title 'Carpe diem. Make your lives extraordinary.' and a bio about working in Artificial Intelligence, Robotics, Ubiquitous Computing, and the Internet of Things. It also lists current projects and collaborations. The 'Research Groups and Projects' section mentions being a member of CHON, co-coordinator of OBINVEST, and founder of the Turing Project. The 'Languages and frameworks I've interacted' section lists Java, MySQL, and others. Finally, there's a 'Follow me on Social Media' section with links to LinkedIn, ResearchGate, Google Scholar, ID, and Instagram.

# GitHub



The screenshot shows a GitHub profile page for the user 'profpantoja'. At the top, there's a navigation bar with tabs: 'Overview' (selected), 'Repositories' (highlighted with a red box and a cursor hand icon), 'Projects', 'Packages', and 'Stars 2'. Below the navigation is a circular profile picture of a man with rainbow hair, holding a whiteboard marker. The profile section includes the name 'Kadu Pantoja', handle 'profpantoja', and a brief bio: 'Professor and researcher at CEFET/RJ, master and Ph.D in Computing at the Military Institute of Engineering (IME) and UFF with mobility at Sorbonne UPMC.' It also lists '62 followers' and '8 following'. Below the profile is a 'Edit profile' button. The main content area starts with a header 'Carpe diem. Make your lives extraordinary.' followed by a bio about working in Artificial Intelligence, Robotics, Ubiquitous Computing, and the Internet of Things. It also mentions current projects like 'multi-agent-system', 'embedded-system', 'modeling-language', and 'metamodel'. A 'Research Groups and Projects' section lists membership in CHON, co-ordination of OBINVEST, and founding the Turing Project. A 'Languages and frameworks I've interacted' section shows icons for Java, MySQL, and others. Finally, a 'Follow me on Social Media' section provides links to LinkedIn, ResearchGate, Google Scholar, iD, and Instagram.

# GitHub



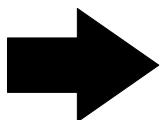
Screenshot of a GitHub user profile page for [profpantoja](https://github.com/profpantoja).

The profile page shows the user's profile picture (a person with rainbow hair holding a paintbrush), name (**Kadu Pantoja**), handle (**profpantoja**), bio (Professor and researcher at CEFET/RJ, master and Ph.D in Computing at the Military Institute of Engineering (IME) and UFF with mobility at Sorbonne UPMC.), follower count (62 followers), and following count (8 following).

The user has 10 repositories:

- java-exercicios** (Public): A repository for exercise lists related to Programming II. It was last updated yesterday and is written in Java.
- chonGame** (Private): A JavaFX game for learning object-oriented concepts. It was last updated 2 weeks ago.
- bd-exercicios** (Public): A repository for exercise lists related to Database Modeling and Databases. It was last updated on Dec 13, 2023, and has 1 star.

The page also lists the user's research groups, languages/frameworks used, and social media links.



Screenshot of a GitHub profile page for [profpantoja](https://github.com/profpantoja).

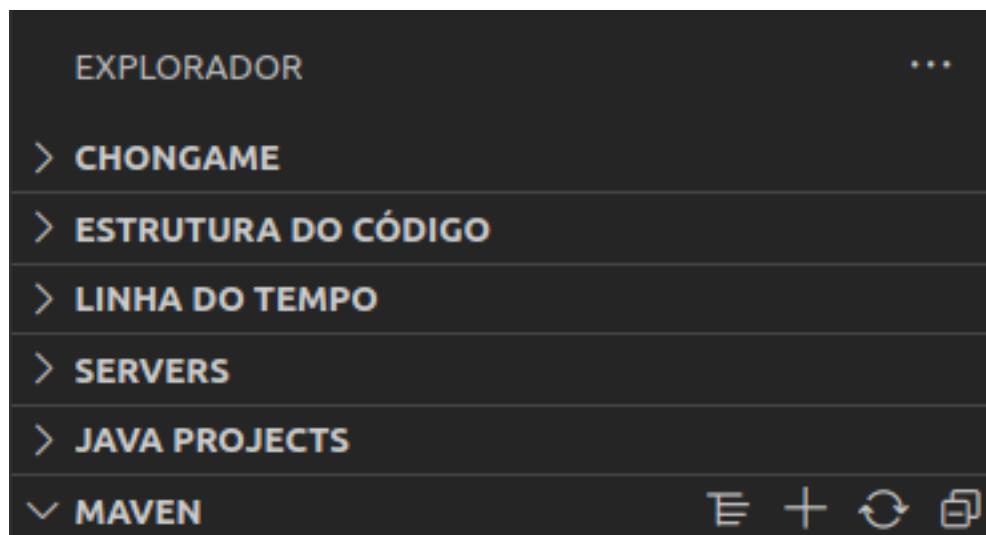
The profile page shows the user's profile picture (a person with rainbow hair holding a paintbrush), name (**Kadu Pantoja**), bio, follower count (62), and links to their CEFET/RJ profile and personal website.

The main content area displays several repositories:

- java-exercicios** (Public): A repository for exercise lists related to Programming II. It includes a note about working in Artificial Intelligence Engineering and mentions Java as the language used.
- chonGame** (Private): A JavaFX game for learning object-oriented concepts. It was last updated 2 weeks ago.
- bd-exercicios** (Public): A repository for exercise lists related to Database Modeling and Databases. It includes a note about being a member of Cognitive Hardware and co-coordinator of the Brasília Turing Project.

A red box highlights the **chonGame** repository entry.

# Creating a New Maven Project



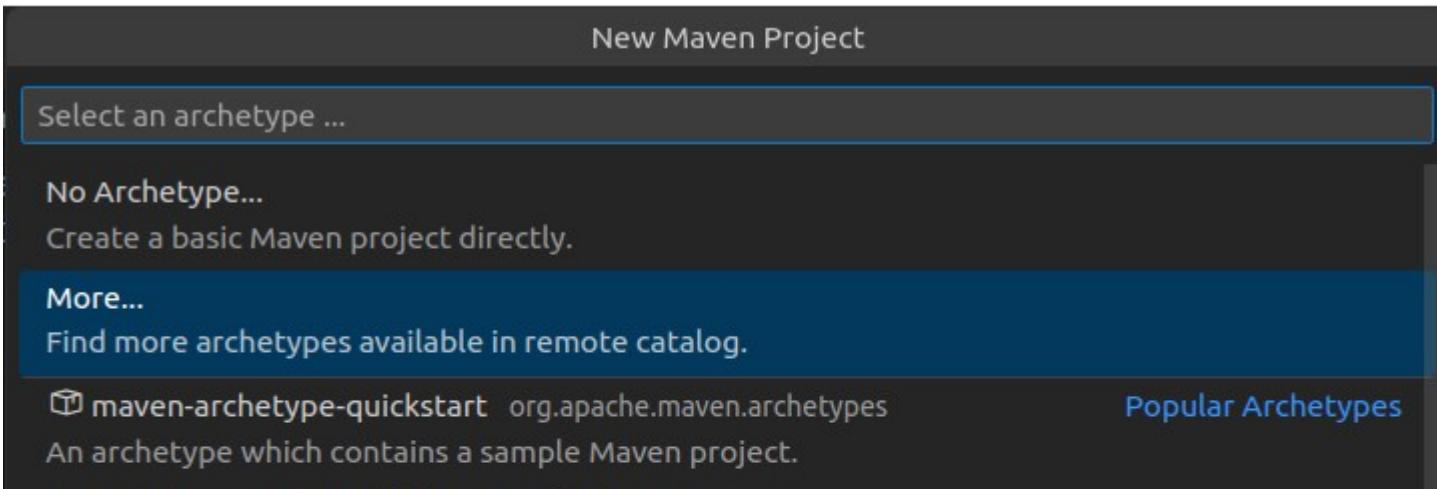
# Creating a New Maven Project



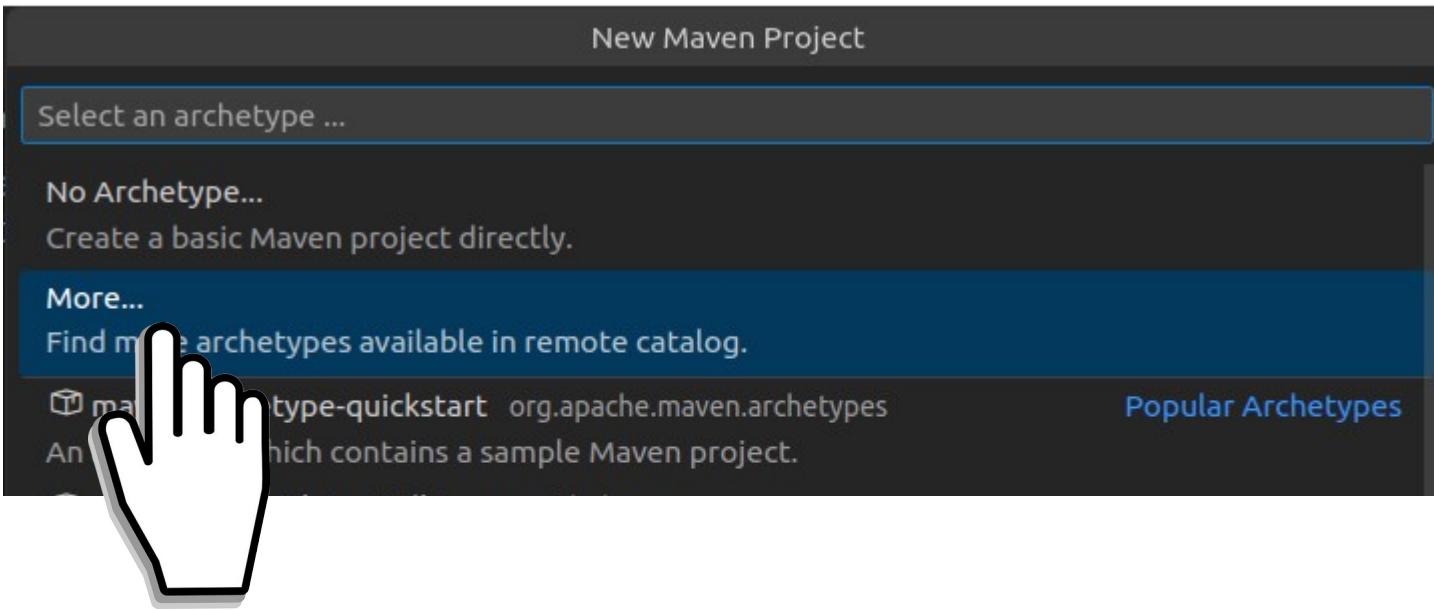
# Creating a New Maven Project



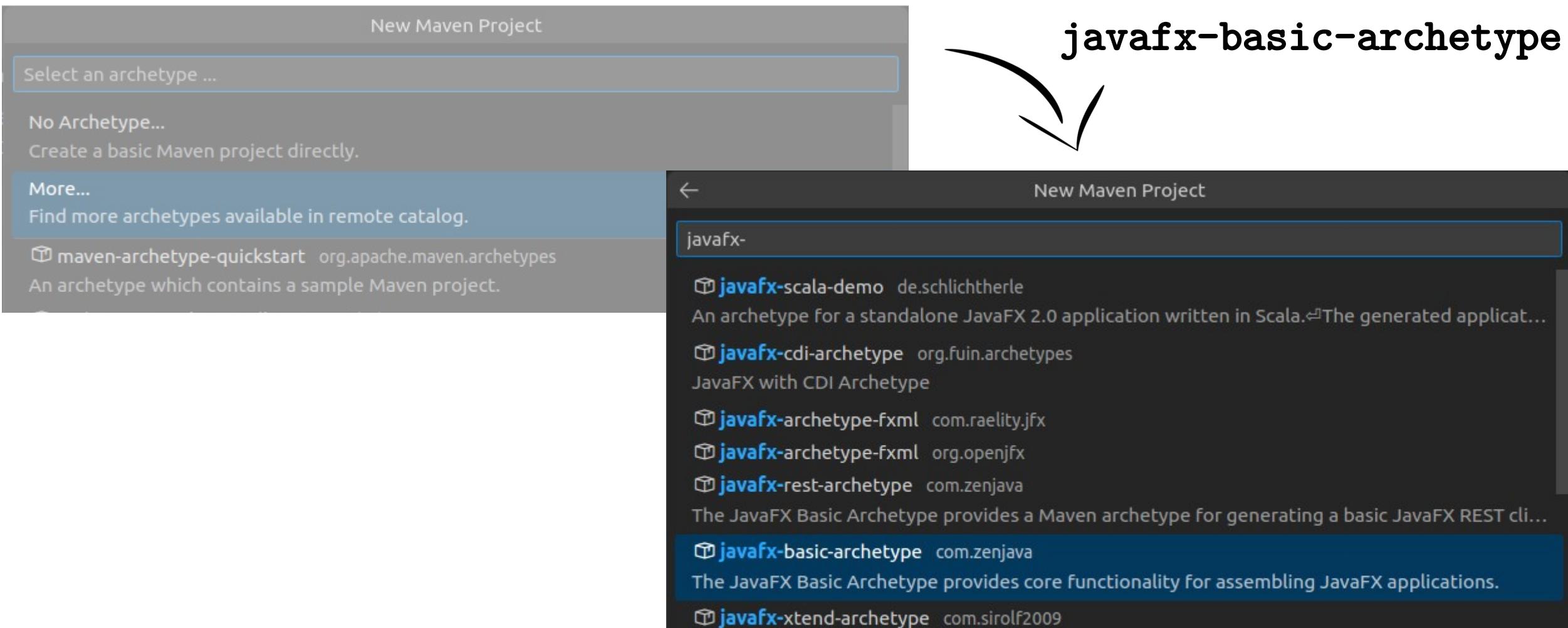
# Creating a New Maven Project



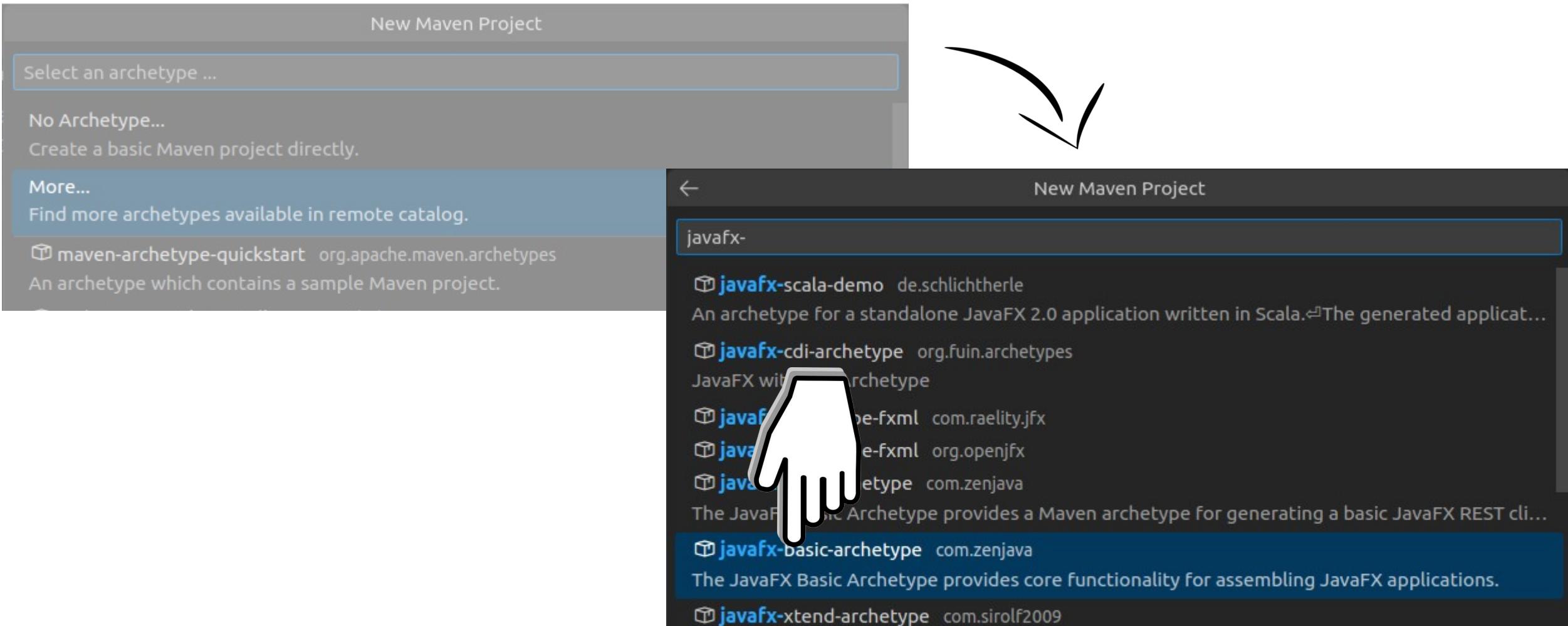
# Creating a New Maven Project



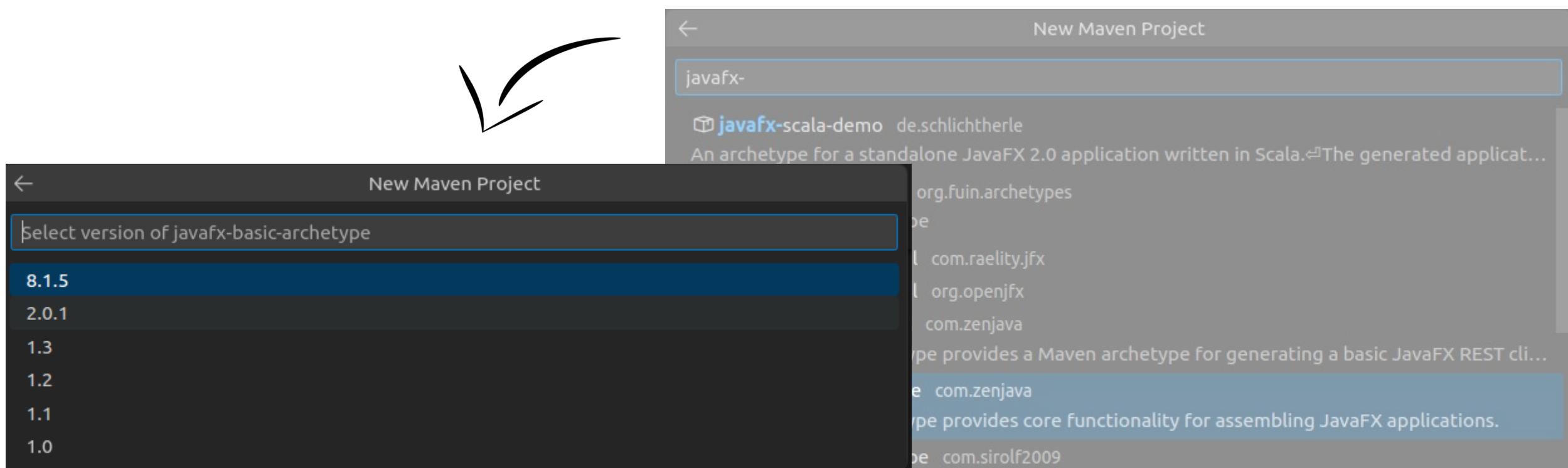
# Creating a New Maven Project



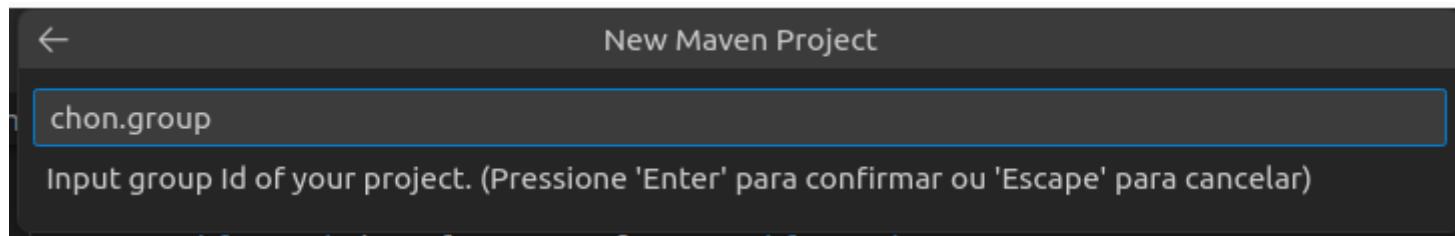
# Creating a New Maven Project



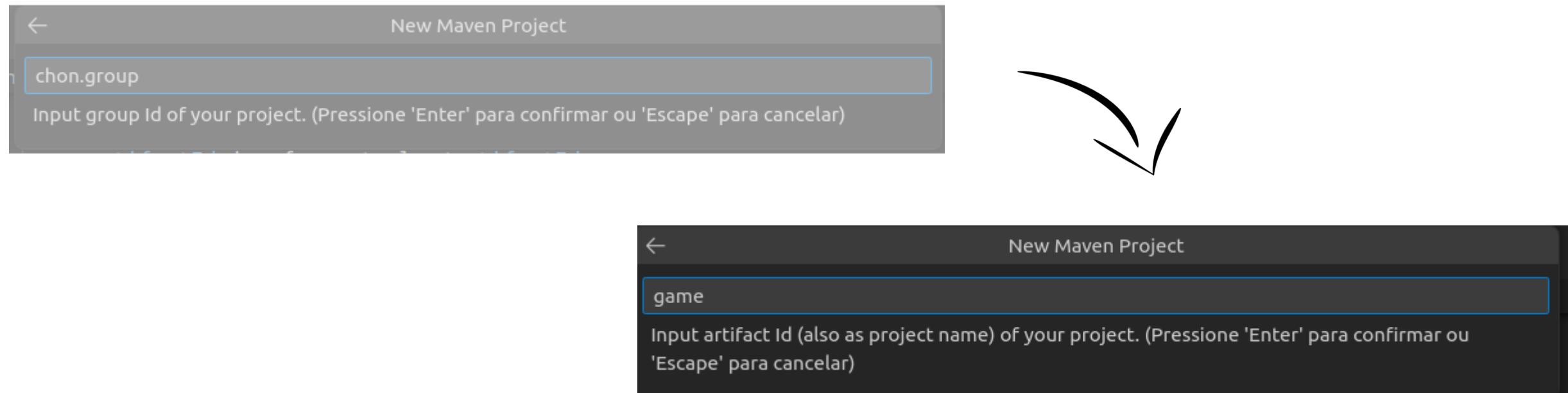
# Creating a New Maven Project



# Creating a New Maven Project



# Creating a New Maven Project



# Creating a New Maven Project

```
○ * Executando tarefa: mvn org.apache.maven.plugins:maven-archetype-plugin:3.1.2:generate -DarchetypeArtifactId="javafx-basi c-archetype" -DarchetypeGroupId="com.zenjava" -DarchetypeVersion="8.1.5" -DgroupId="chon.group" -DartifactId="game"

[INFO] Scanning for projects...
[INFO]
[INFO] -----< chon.group:game >-----
[INFO] Building game 1
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.1.2:generate (default-cli) > generate-sources @ game >>
[INFO]
[INFO] <<< maven-archetype-plugin:3.1.2:generate (default-cli) < generate-sources @ game <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.1.2:generate (default-cli) @ game ---
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [com.zenjava/javafx-basic-archetype:8.1.5] found in catalog remote
Downloading from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basic-archetype/8.1.5/javafx-basic-archety pe-8.1.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basic-archetype/8.1.5/javafx-basic-archetyp e-8.1.5.pom (6.2 kB at 17 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basic-archetype/8.1.5/javafx-basic-archety pe-8.1.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basic-archetype/8.1.5/javafx-basic-archetyp e-8.1.5.jar (145 kB at 188 kB/s)
[INFO] Using property: groupId = chon.group
[INFO] Using property: artifactId = game
Define value for property 'version' 1.0-SNAPSHOT: :
```

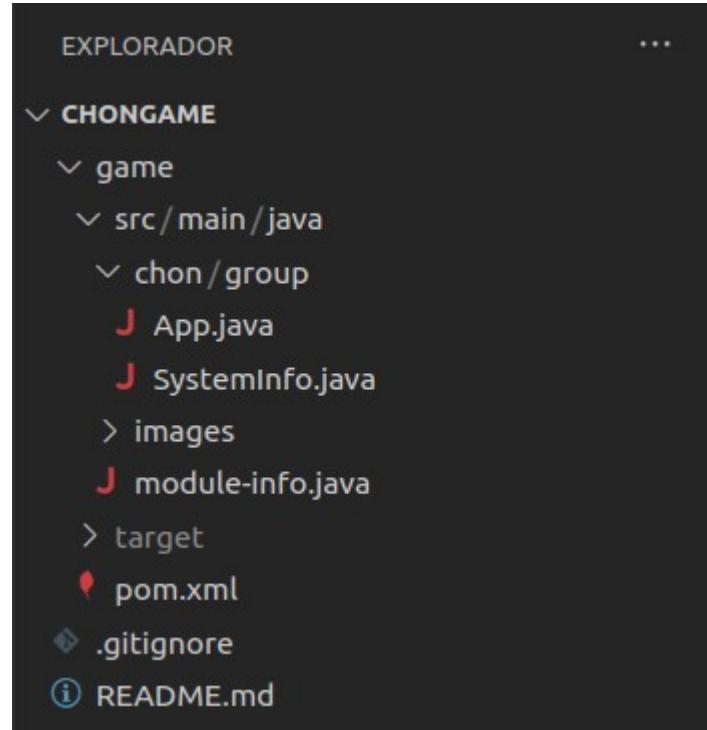
# Creating a New Maven Project

```
○ * Executando tarefa: mvn org.apache.maven.plugins:maven-archetype-plugin:3.1.2:generate -DarchetypeArtifactId="javafx-basi  
c-archetype" -DarchetypeGroupId="com.zenjava" -DarchetypeVersion="8.1.5" -DgroupId="chon.group" -DartifactId="game"  
  
[INFO] Scanning for projects...  
[INFO]  
[INFO] -----< chon.group:game >-----  
[INFO] Building game 1  
[INFO] -----[ jar ]-----  
[INFO]  
[INFO] >>> maven-archetype-plugin:3.1.2:generate (default-cli) > generate-sources @ game >>>  
[INFO]  
[INFO] <<< maven-archetype-plugin:3.1.2:generate (default-cli) < generate-sources @ game <<<  
[INFO]  
[INFO]  
[INFO] --- maven-archetype-plugin:3.1.2:generate (default-cli) @ game ---  
[INFO] Generating project in Interactive mode  
[INFO] Archetype repository not defined. Using the one from [com.zenjava/javafx-basic-archetype:8.1.5] found in catalog remote  
Downloading from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basi  
pe-8.1.5.pom  
Downloaded from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basi  
e-8.1.5.pom (6.2 kB at 17 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basi  
pe-8.1.5.jar  
Downloaded from central: https://repo.maven.apache.org/maven2/com/zenjava/javafx-basi  
e-8.1.5.jar (145 kB at 188 kB/s)  
[INFO] Using property: groupId = chon.group  
[INFO] Using property: artifactId = game  
Define value for property 'version' 1.0-SNAPSHOT: :
```

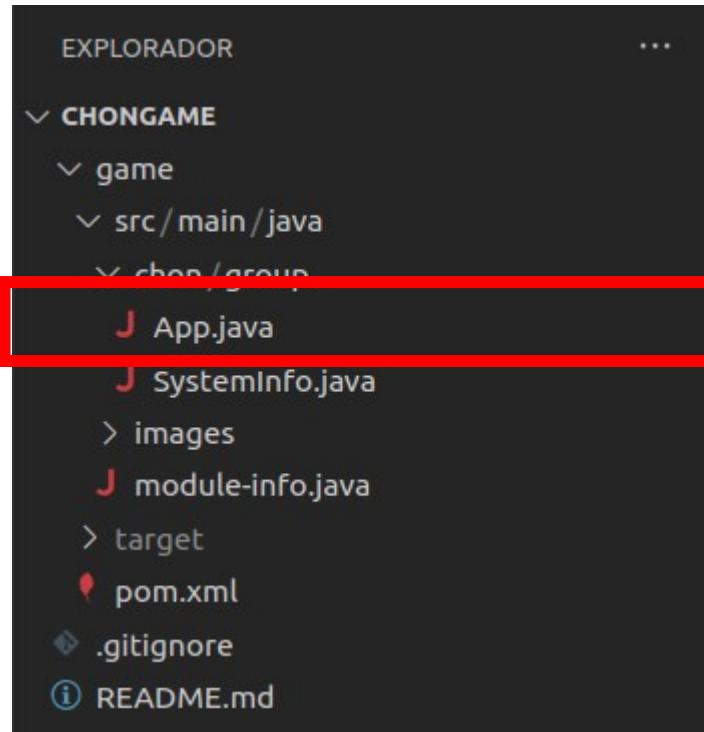


```
[INFO] Using property: groupId = chon.group  
[INFO] Using property: artifactId = game  
Define value for property 'version' 1.0-SNAPSHOT: : 1  
[INFO] Using property: package = chon.group  
Define value for property 'organizationName': chon  
Confirm properties configuration:  
groupId: chon.group  
artifactId: game  
version: 1  
package: chon.group  
organizationName: chon  
Y: : Y
```

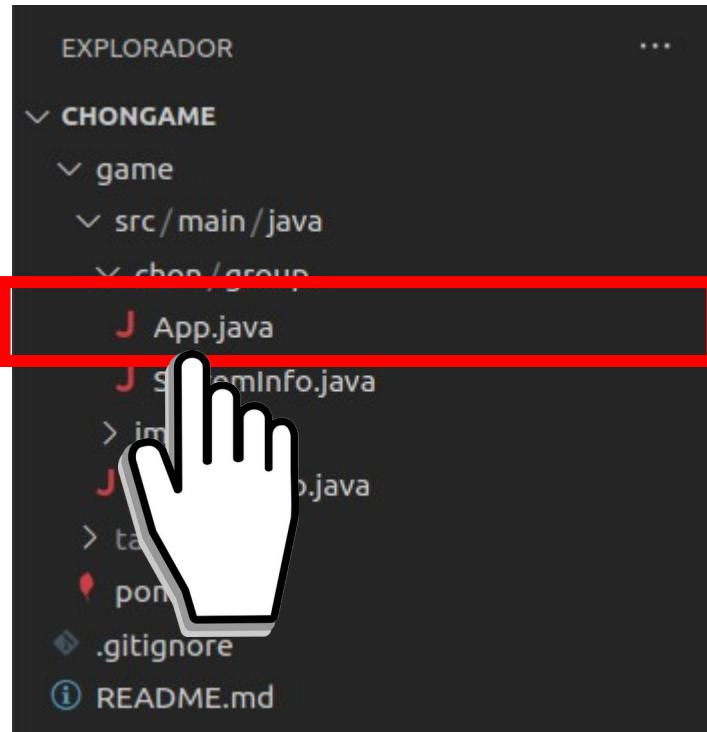
# Java FX App Running



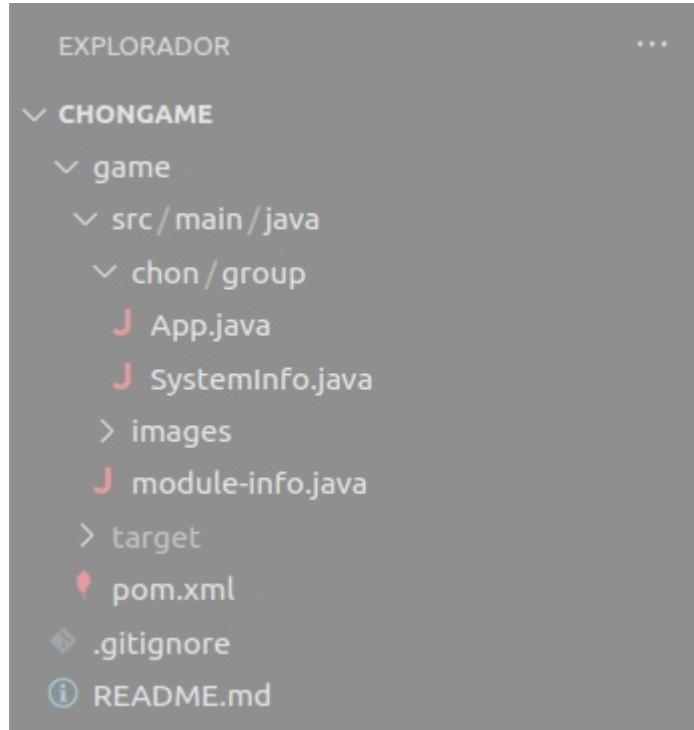
# Java FX App Running



# Java FX App Running



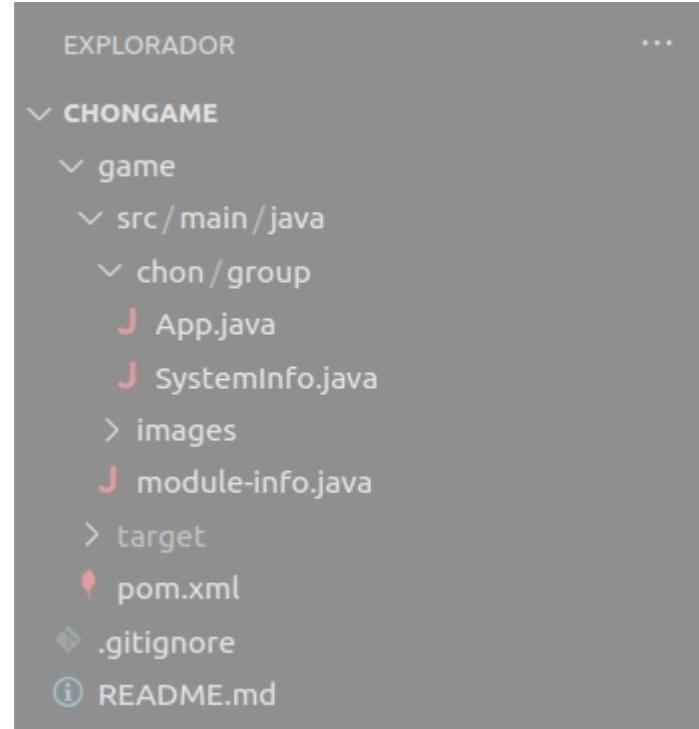
# Java FX App Running



A screenshot of an IDE editor showing the 'App.java' file. The code defines a JavaFX application that prints the Java and JavaFX versions to a window. The code uses Java 8's var keyword for variable declarations.

```
J App.java M X
game > src > main > java > chon > group > J App.java > App > start(Stage)
1 package chon.group;
2
3 import javafx.application.Application;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Label;
6 import javafx.scene.layout.StackPane;
7 import javafx.stage.Stage;
8
9 /**
10 * JavaFX App
11 */
12 public class App extends Application {
13
14     @Override
15     public void start(Stage stage) {
16         var javaVersion = SystemInfo.javaVersion();
17         var javafxVersion = SystemInfo.javafxVersion();
18
19         var label = new Label("Hello, JavaFX " + javafxVersion + ", running on Java " + javaVersion + ".");
20         var scene = new Scene(new StackPane(label), width:640, height:480);
21         stage.setScene(scene);
22         stage.show();
23     }
24
25     public static void main(String[] args) {
26         launch();
27     }
28 }
29 }
```

# Java FX App Running

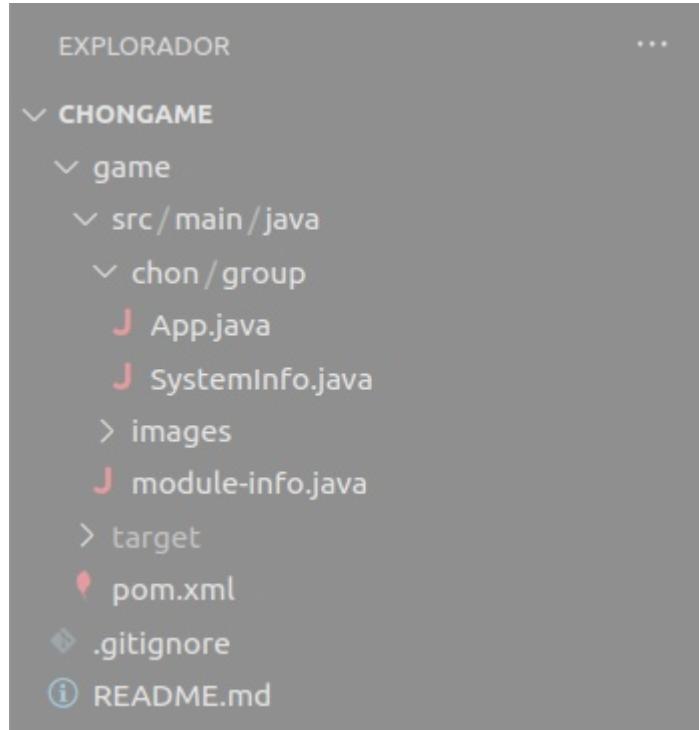


The image shows a code editor window with the file 'App.java' open. The code defines a JavaFX application that prints the Java and JavaFX versions to a window.

```
App.java M X
game > src > main > java > chon > group > J App.java > App > start(Stage)
1 package chon.group;
2
3 import javafx.application.Application;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Label;
6 import javafx.scene.layout.StackPane;
7 import javafx.stage.Stage;
8
9 /**
10 * JavaFX App
11 */
12 public class App extends Application {
13
14     @Override
15     public void start(Stage stage) {
16         var javaVersion = SystemInfo.javaVersion();
17         var javafxVersion = SystemInfo.javafxVersion();
18
19         var label = new Label("Hello, JavaFX " + javafxVersion + ", running on Java " + javaVersion + ".");
20         var scene = new Scene(new StackPane(label), width:640, height:480);
21         stage.setScene(scene);
22         stage.show();
23     }
24
25     public static void main(String[] args) {
26         launch();
27     }
28 }
29 }
```

A red box highlights the 'Run | Debug' button located at the bottom of the code editor window.

# Java FX App Running

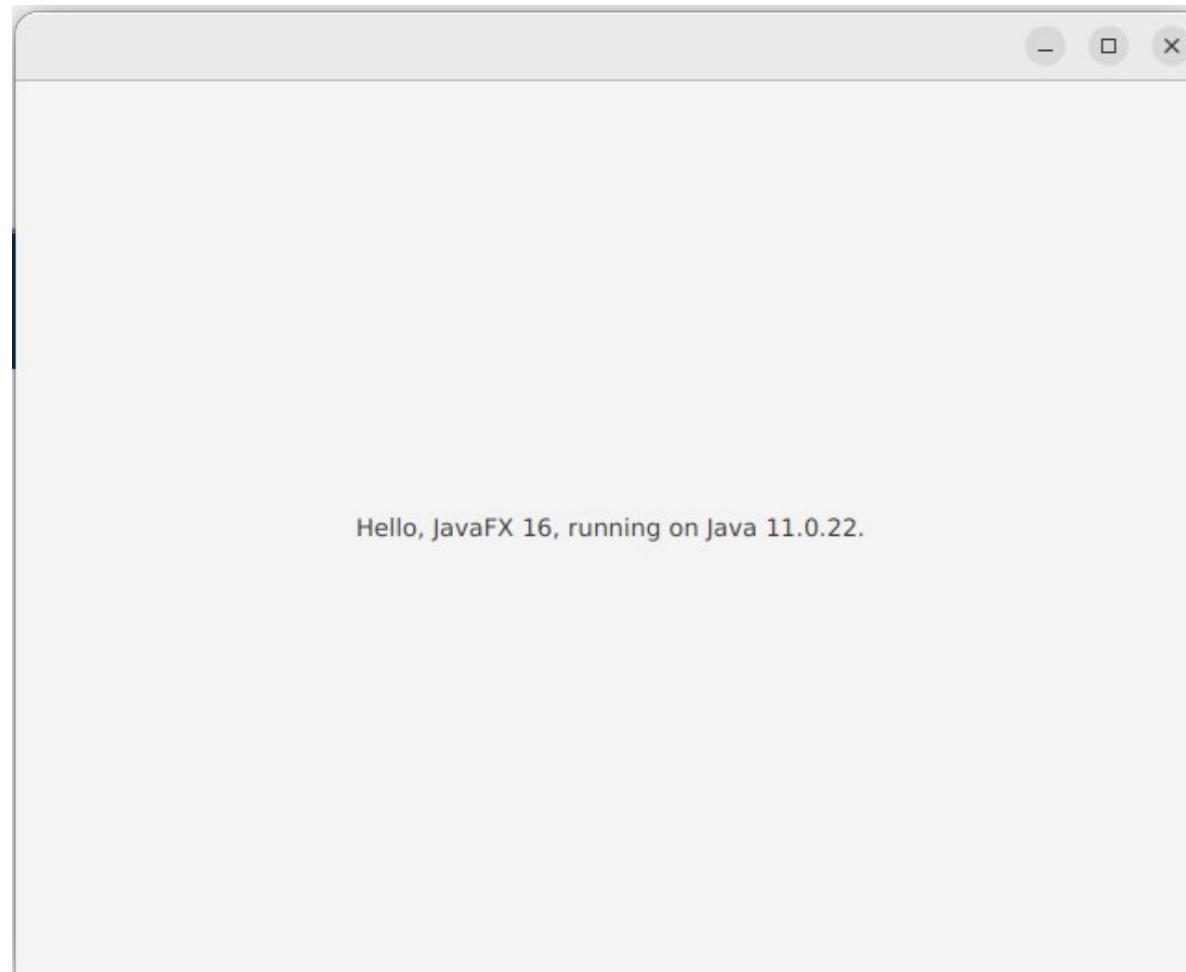


The code in 'App.java' is as follows:

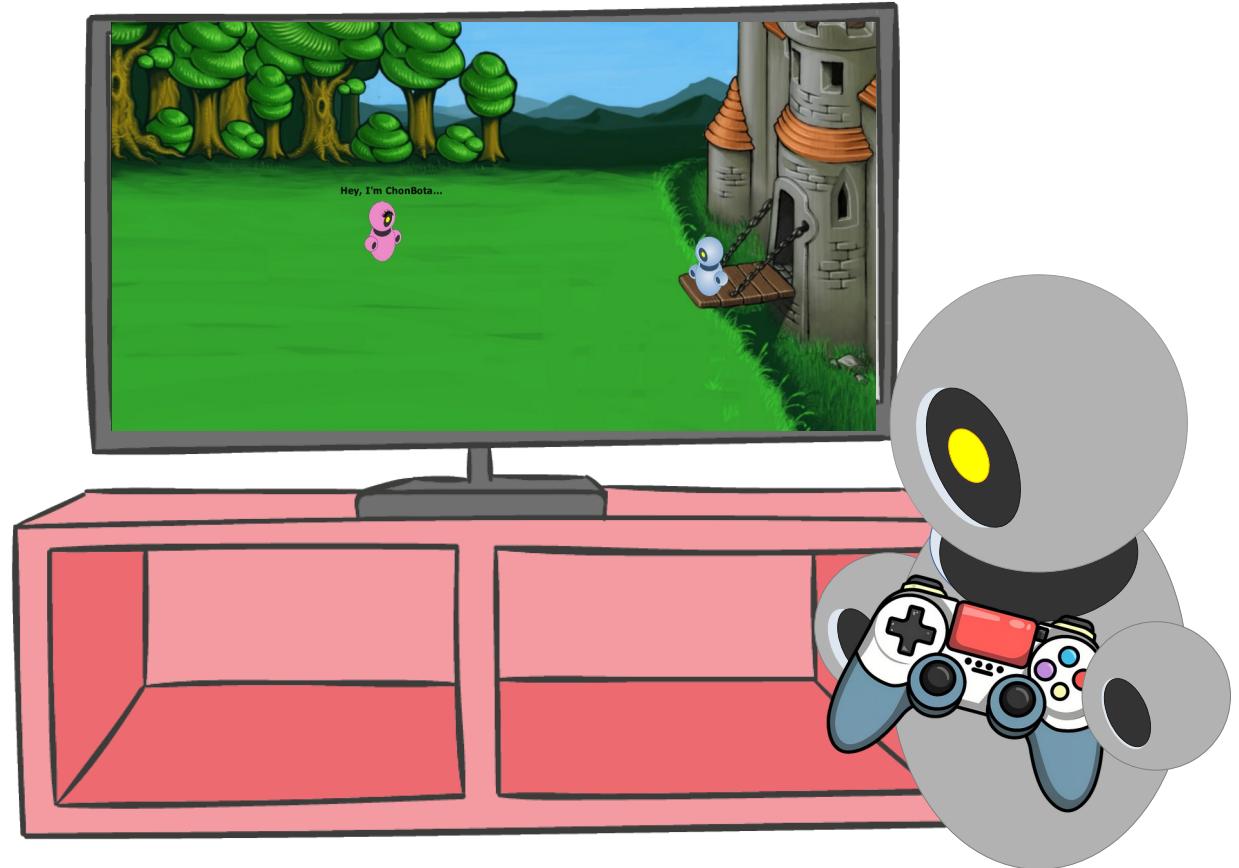
```
App.java M X
game > src > main > java > chon > group > J App.java > App > start(Stage)
1 package chon.group;
2
3 import javafx.application.Application;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Label;
6 import javafx.scene.layout.StackPane;
7 import javafx.stage.Stage;
8
9 /**
10 * JavaFX App
11 */
12 public class App extends Application {
13
14     @Override
15     public void start(Stage stage) {
16         String javaVersion = SystemInfo.javaVersion();
17         String javafxVersion = SystemInfo.javafxVersion();
18
19         Label label = new Label("Hello, JavaFX " + javafxVersion + ", running on Java " + javaVersion + ".");
20         StackPane scene = new StackPane(label), width:640, height:480);
21         stage.setScene(scene);
22         stage.show();
23
24     }
25     public static void main(String[] args) {
26         launch();
27     }
28 }
29 }
```

A hand cursor icon is positioned over the code area, and a red rectangular box highlights the 'Run | Debug' button at the bottom of the code editor window.

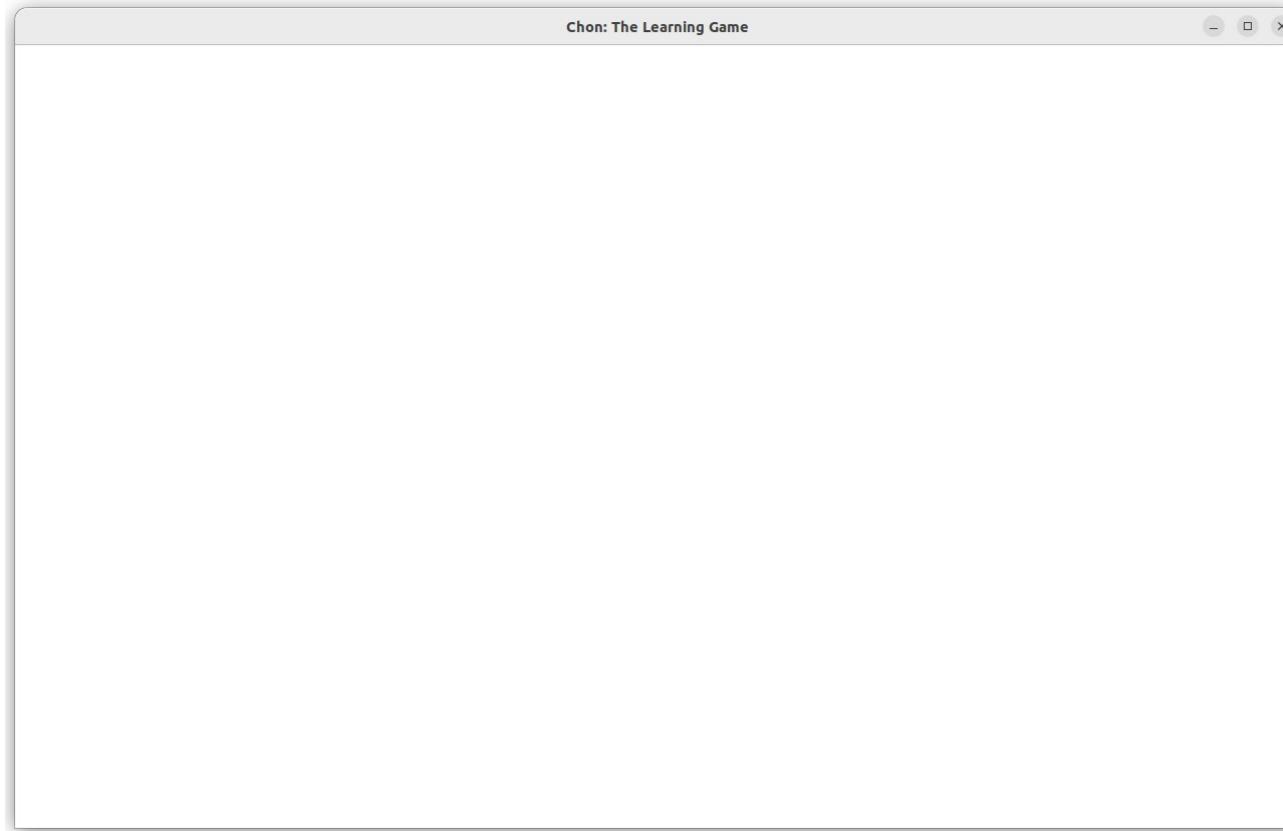
# Java FX App Running



# MANIPULATING GRAPHICAL ELEMENTS

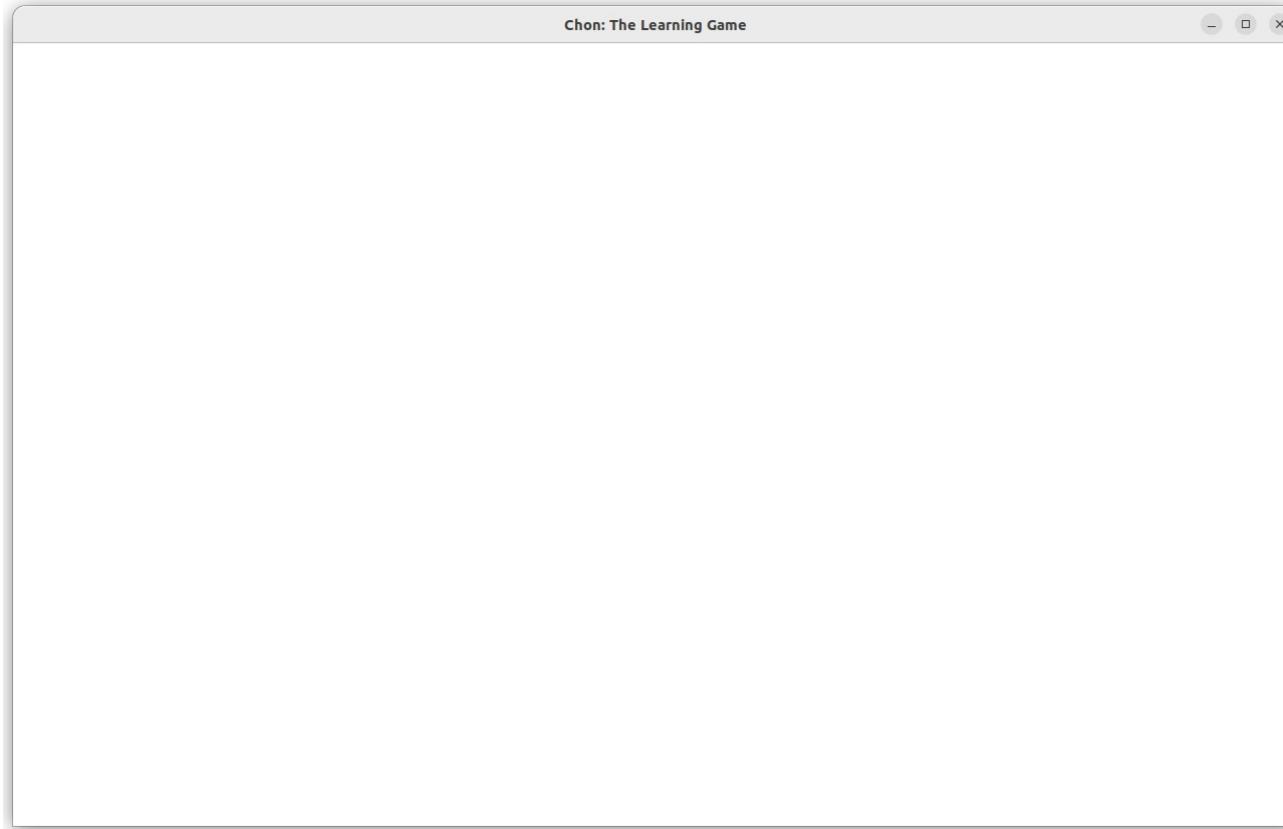


# StackPane



# StackPane

pane



# StackPane

pane

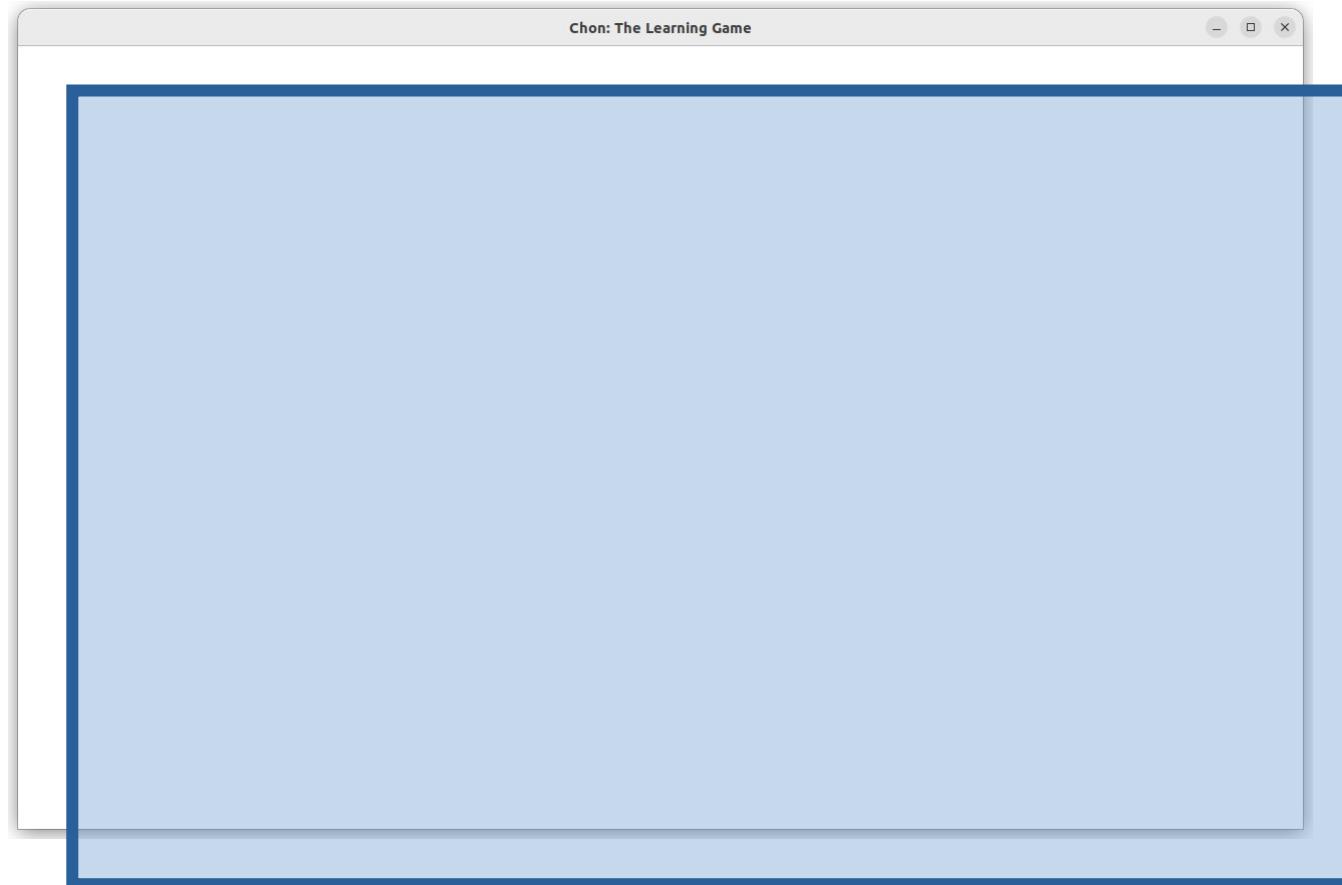


The window  
that is  
currently  
displayed on  
the screen.



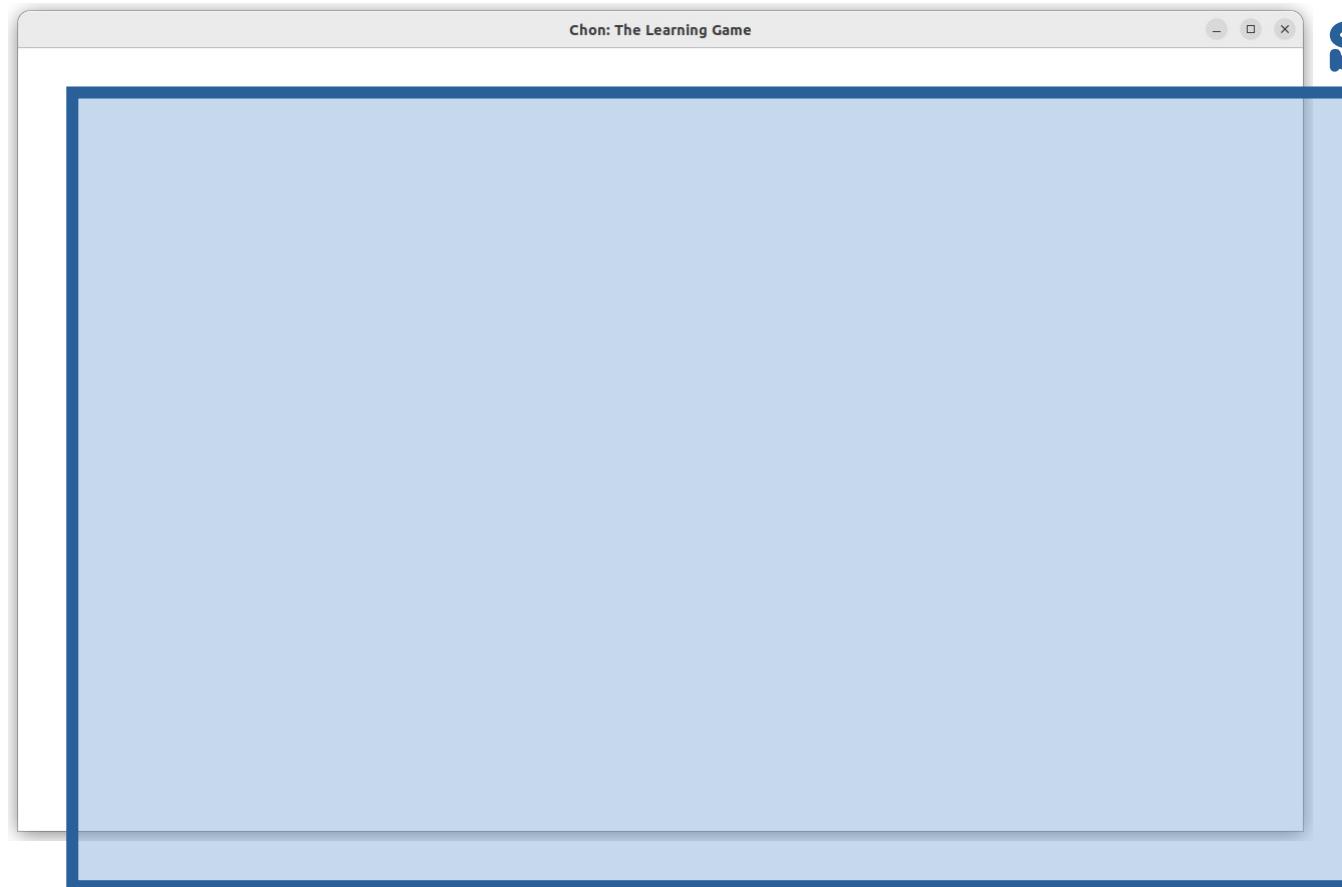
# Scene

pane



# Scene

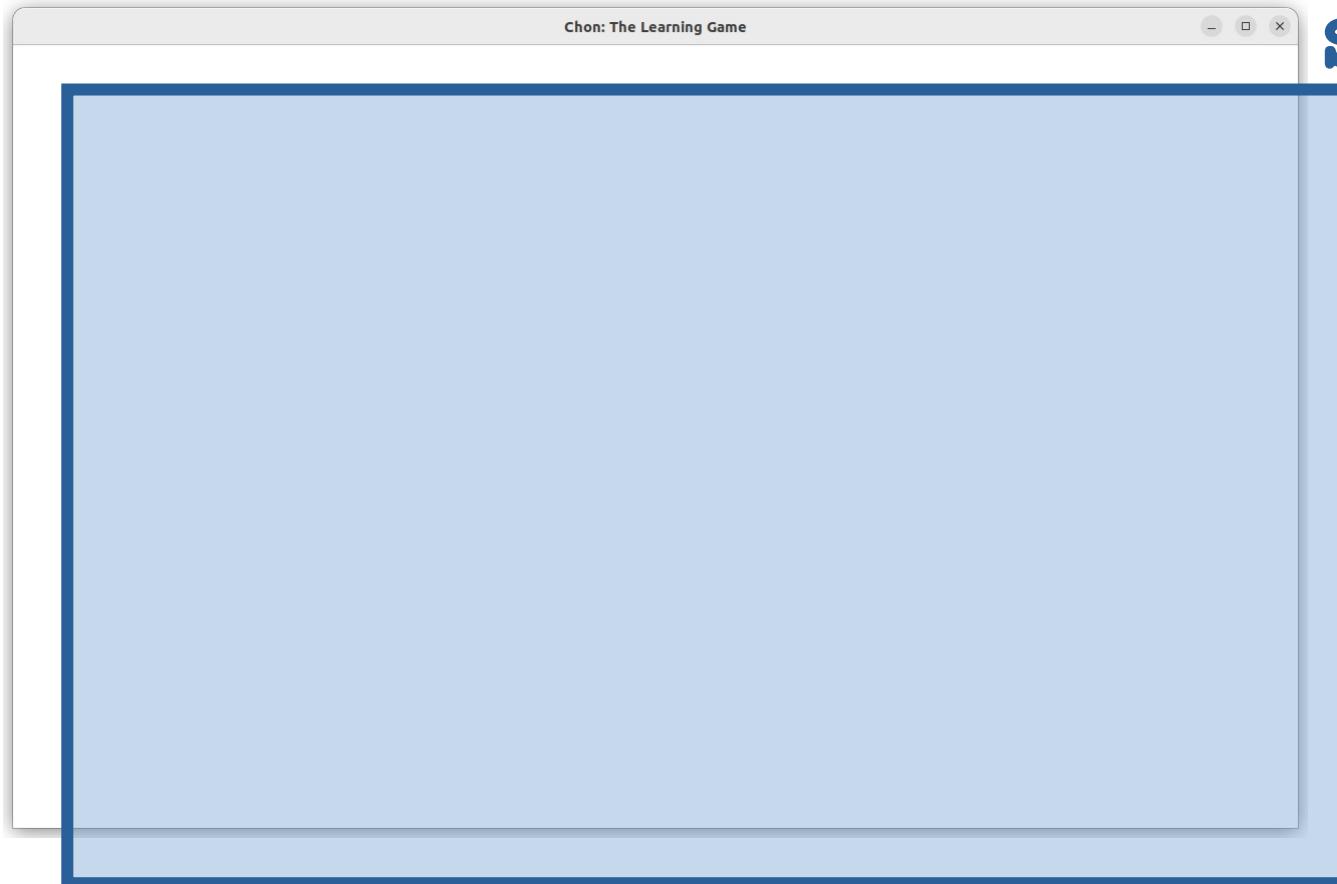
pane



scene

# Scene

pane



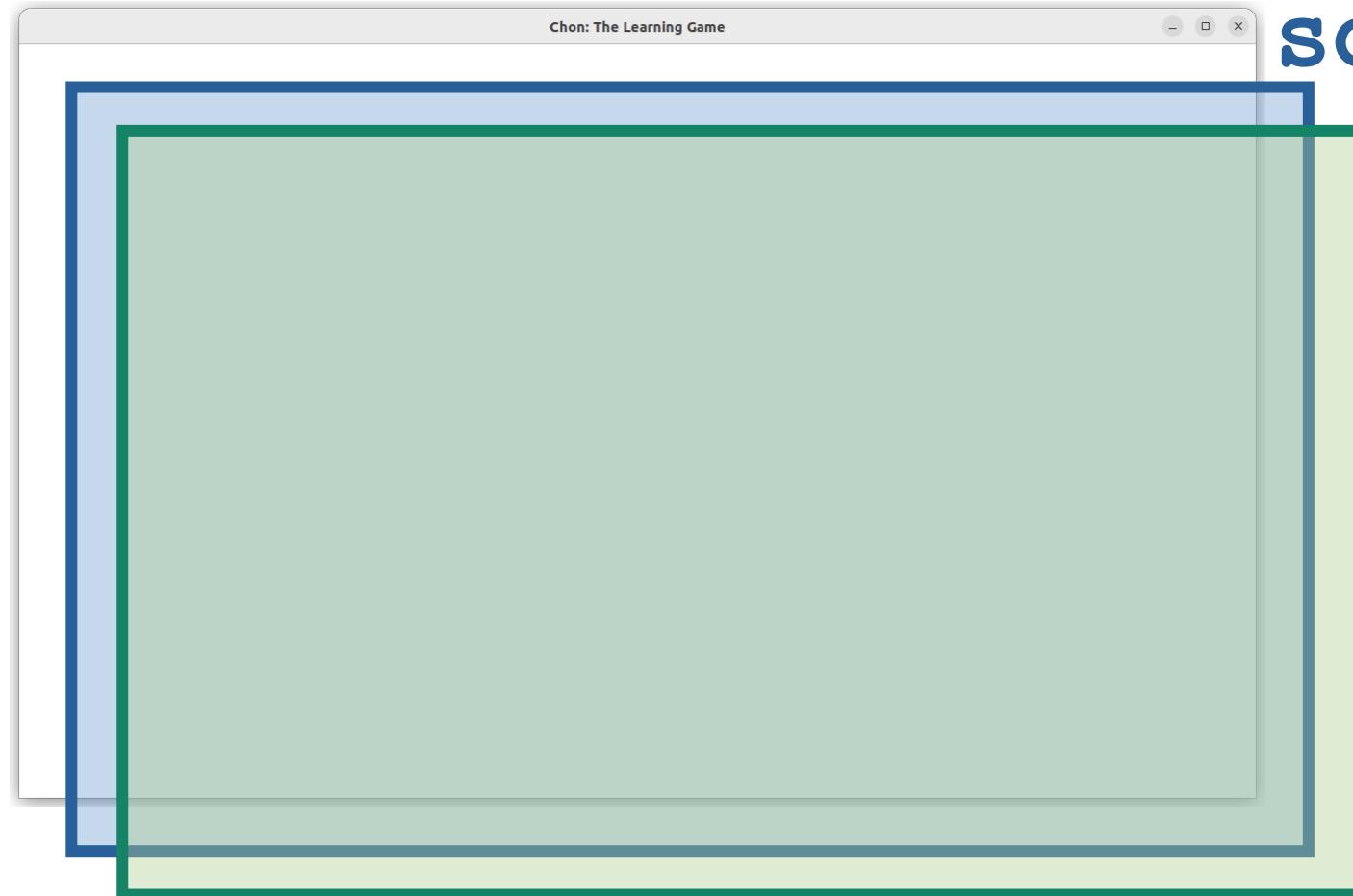
scene



It refers to  
the initial  
visible  
dimensions  
of a pane.

# Canvas

pane



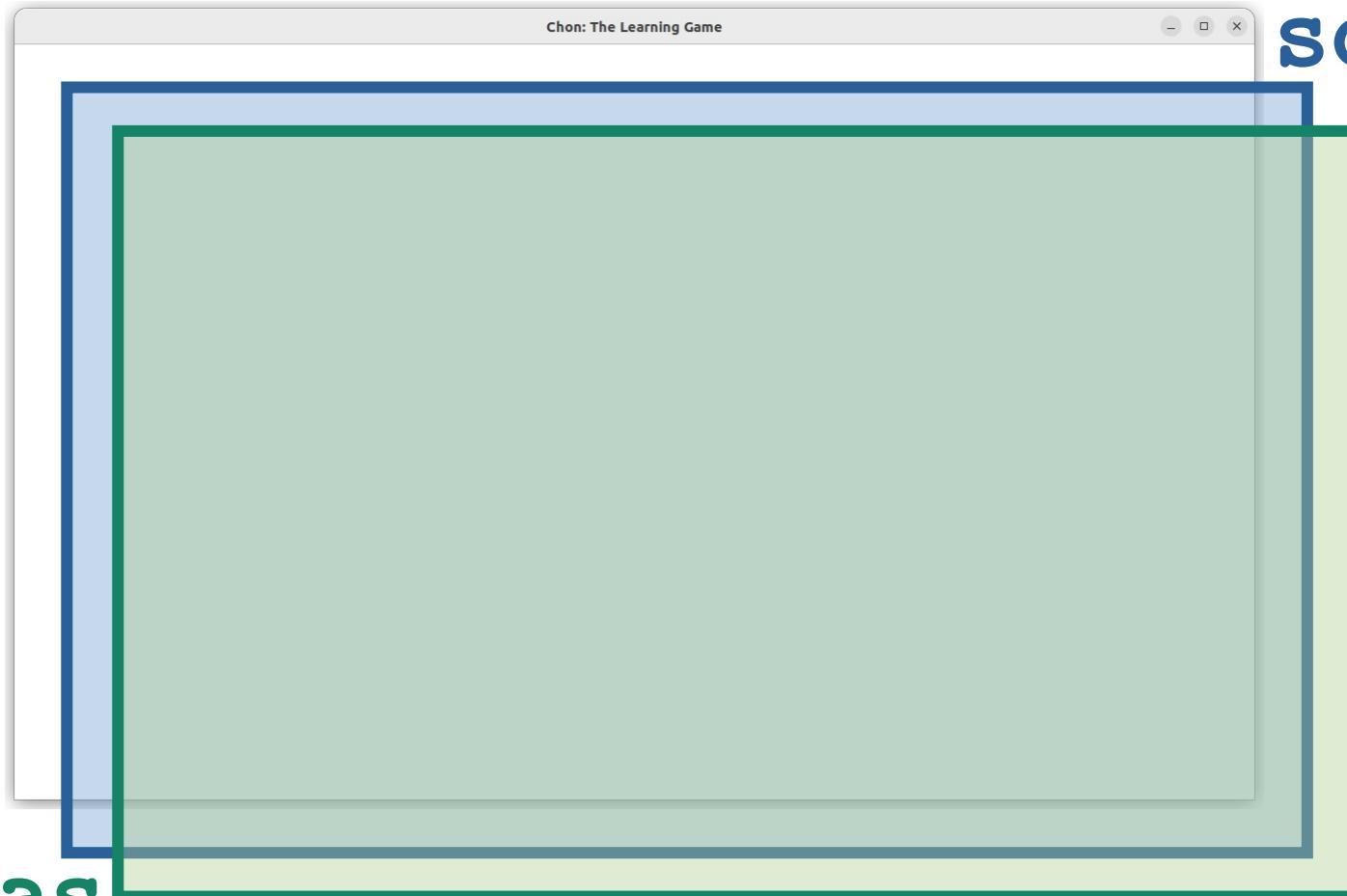
scene

# Canvas

pane

scene

canvas



# Canvas

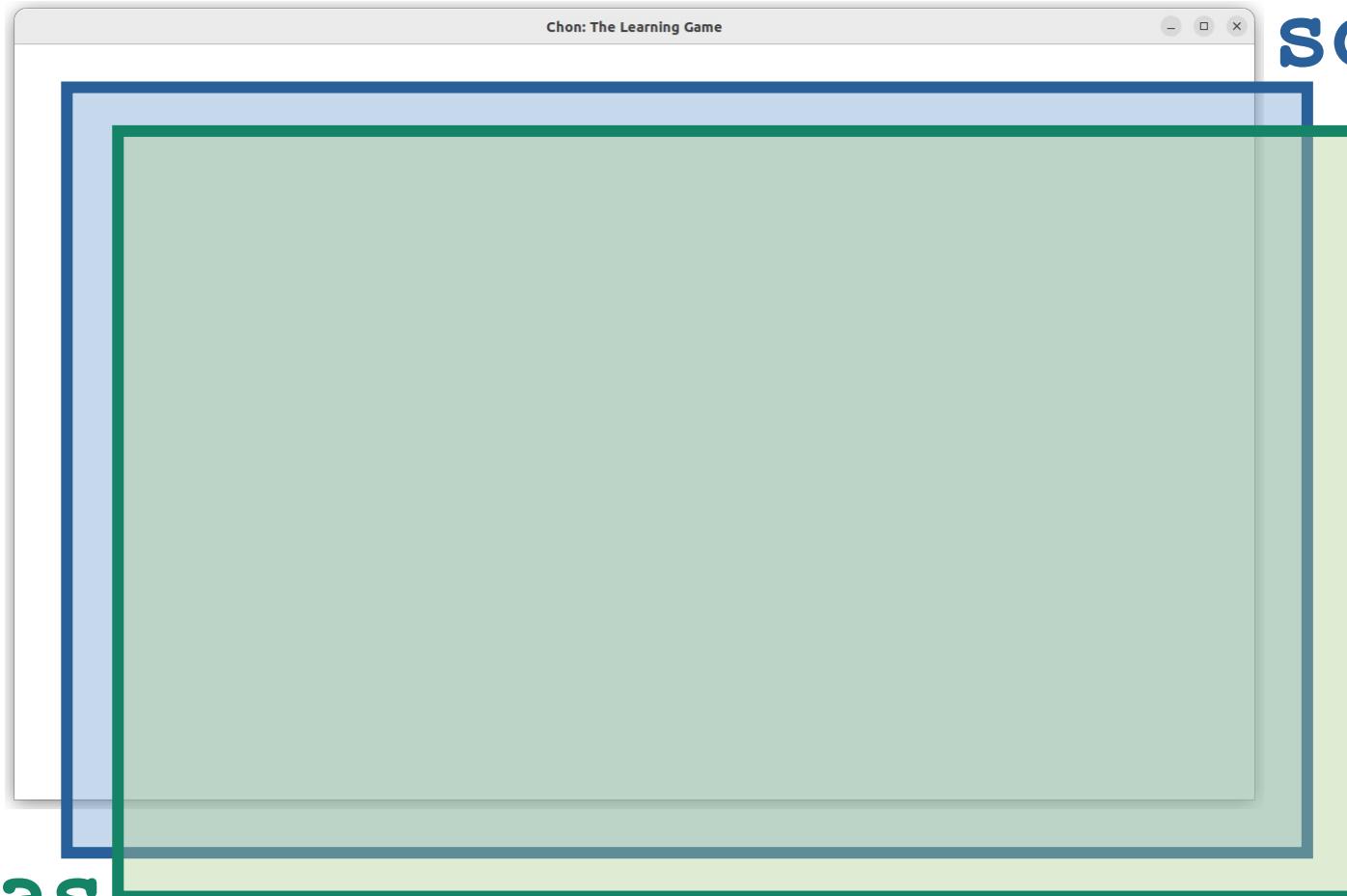
pane

The canvas is  
the part of  
the screen  
where graphics  
are displayed  
and printed.



canvas

scene



# Object's Dimension: Width and Height

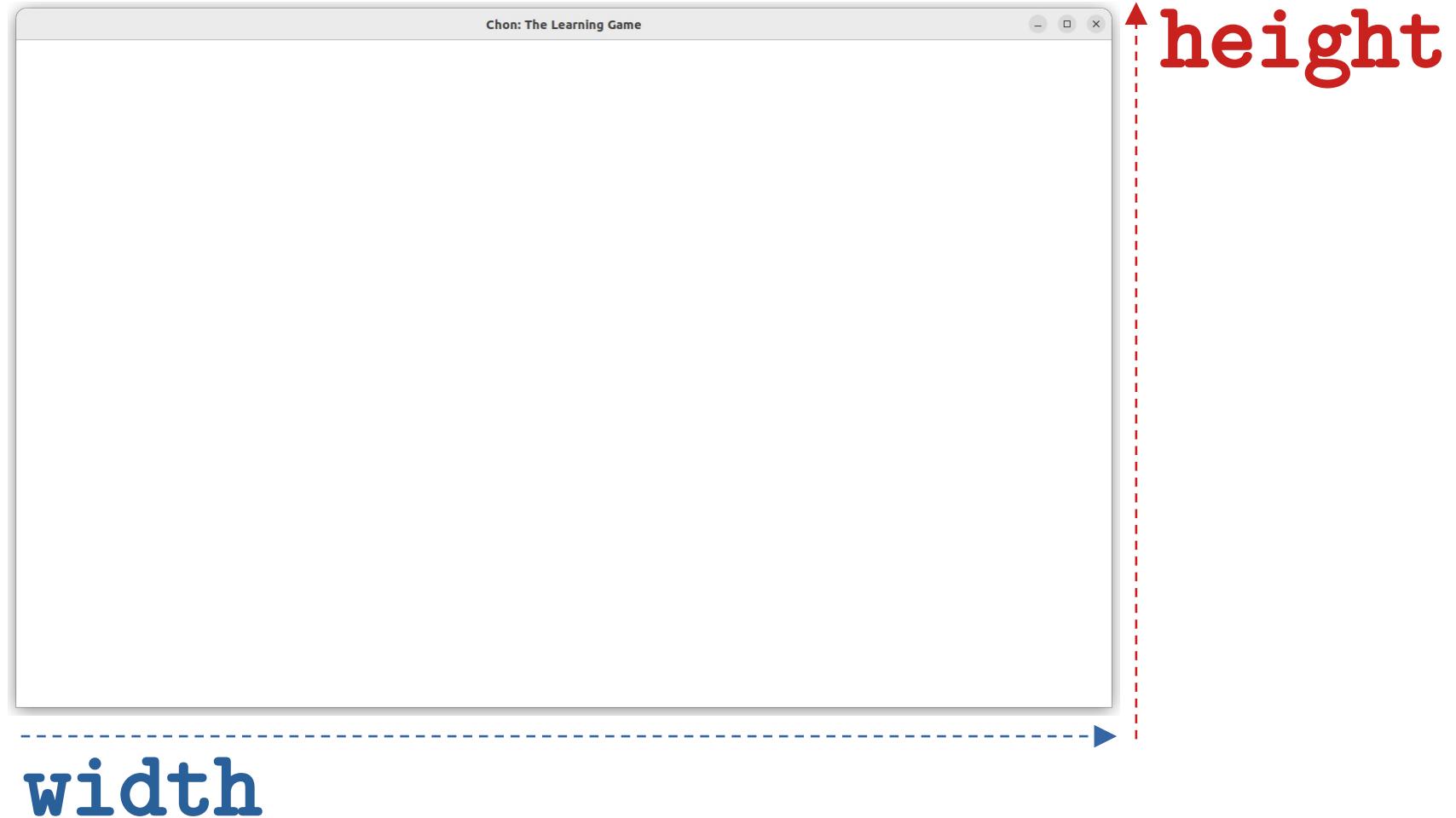


# Object's Dimension: Width and Height

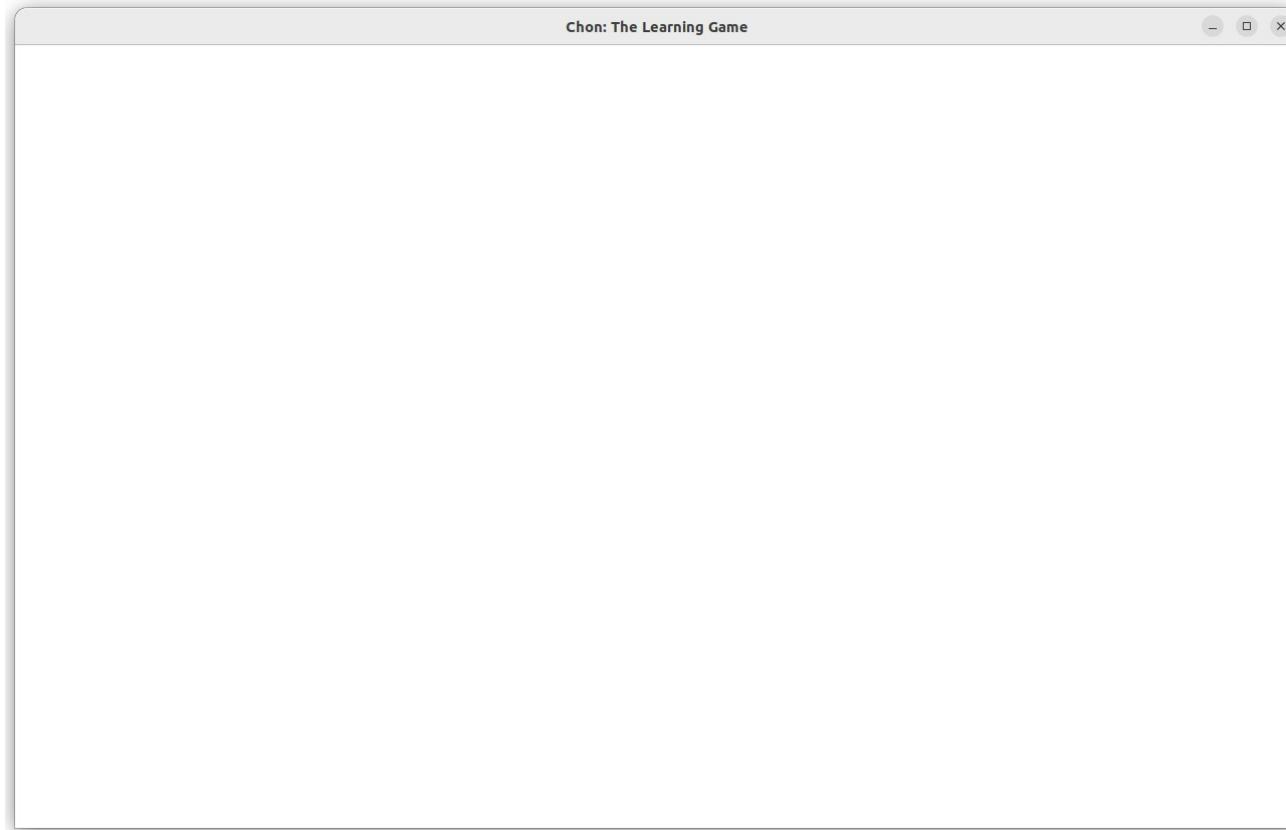


width

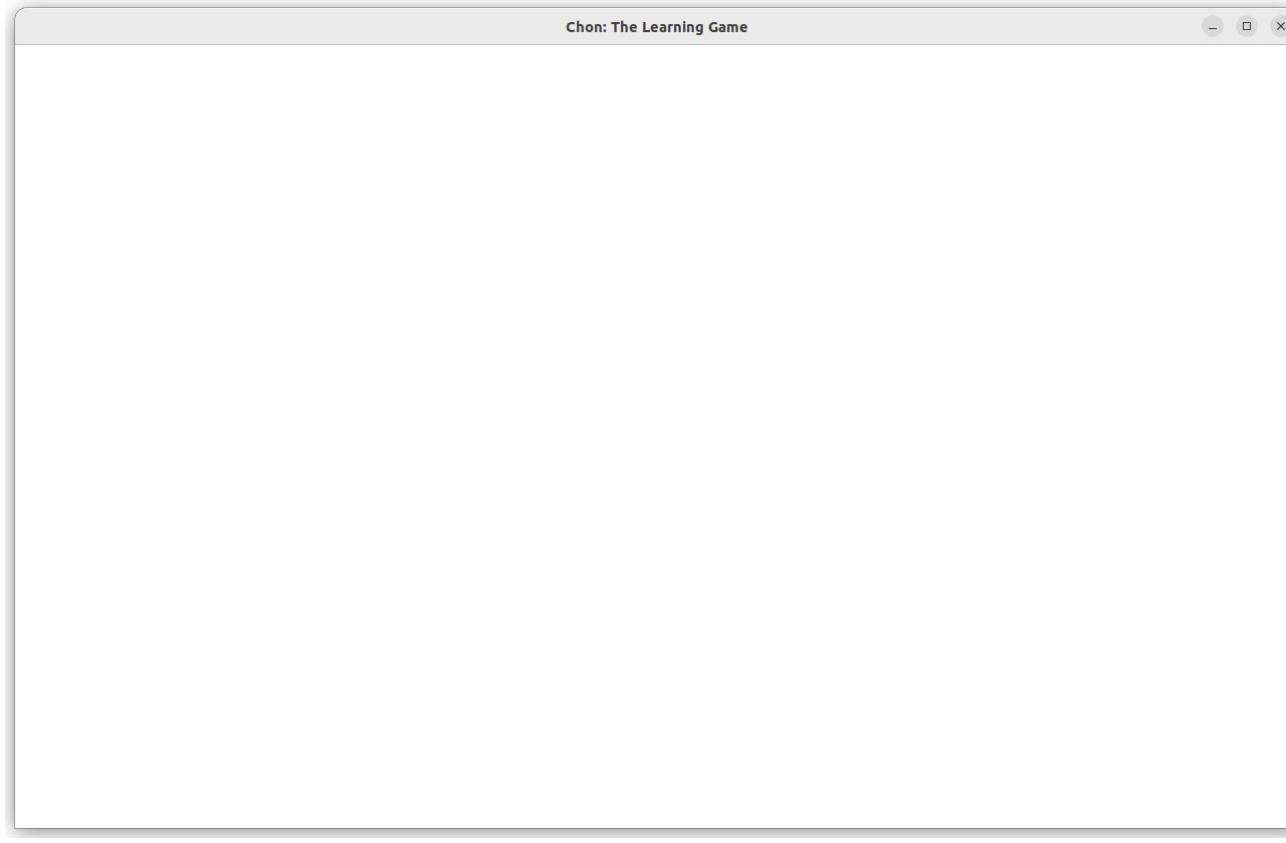
# Object's Dimension: Width and Height



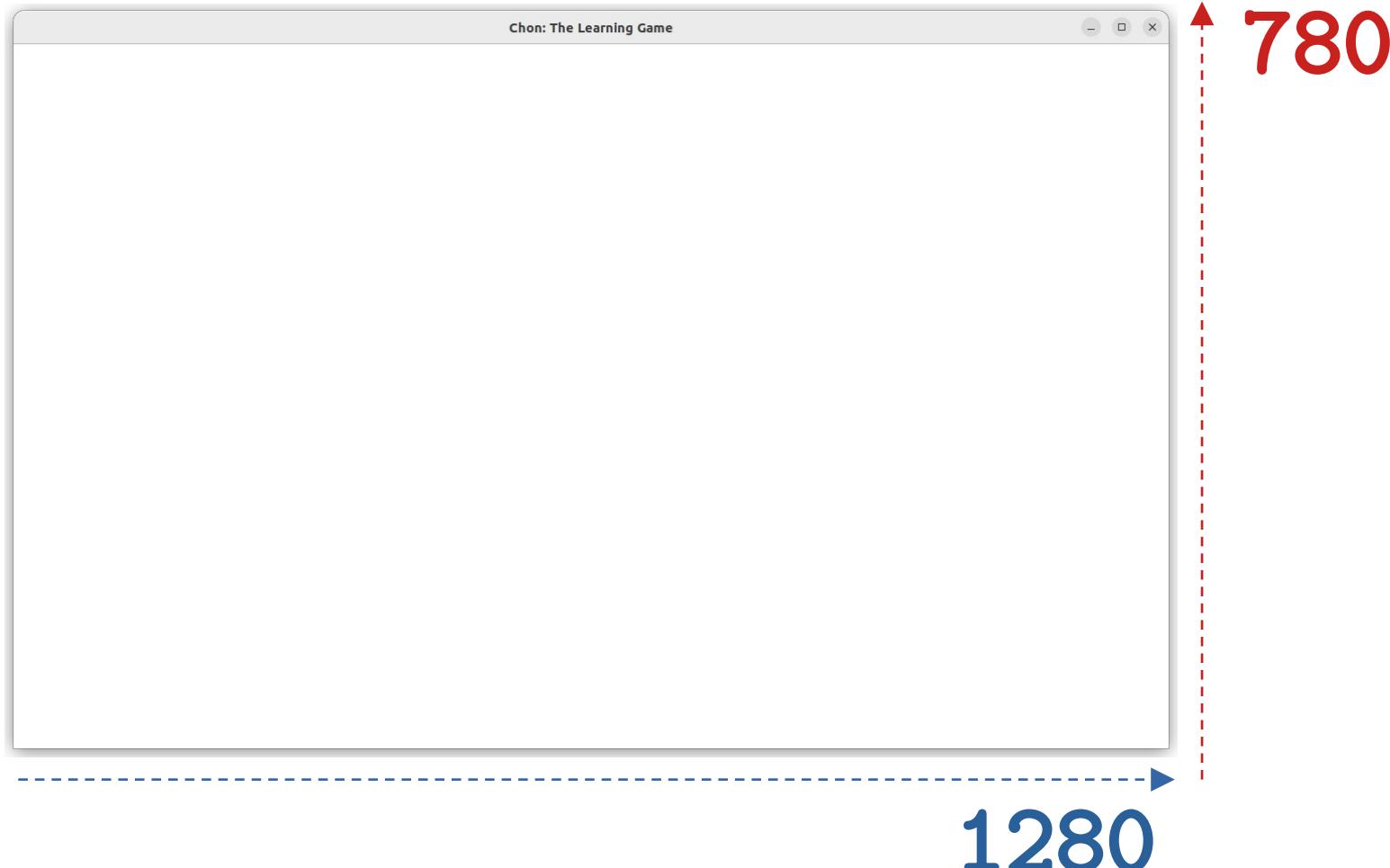
# Object's Dimension: Width and Height



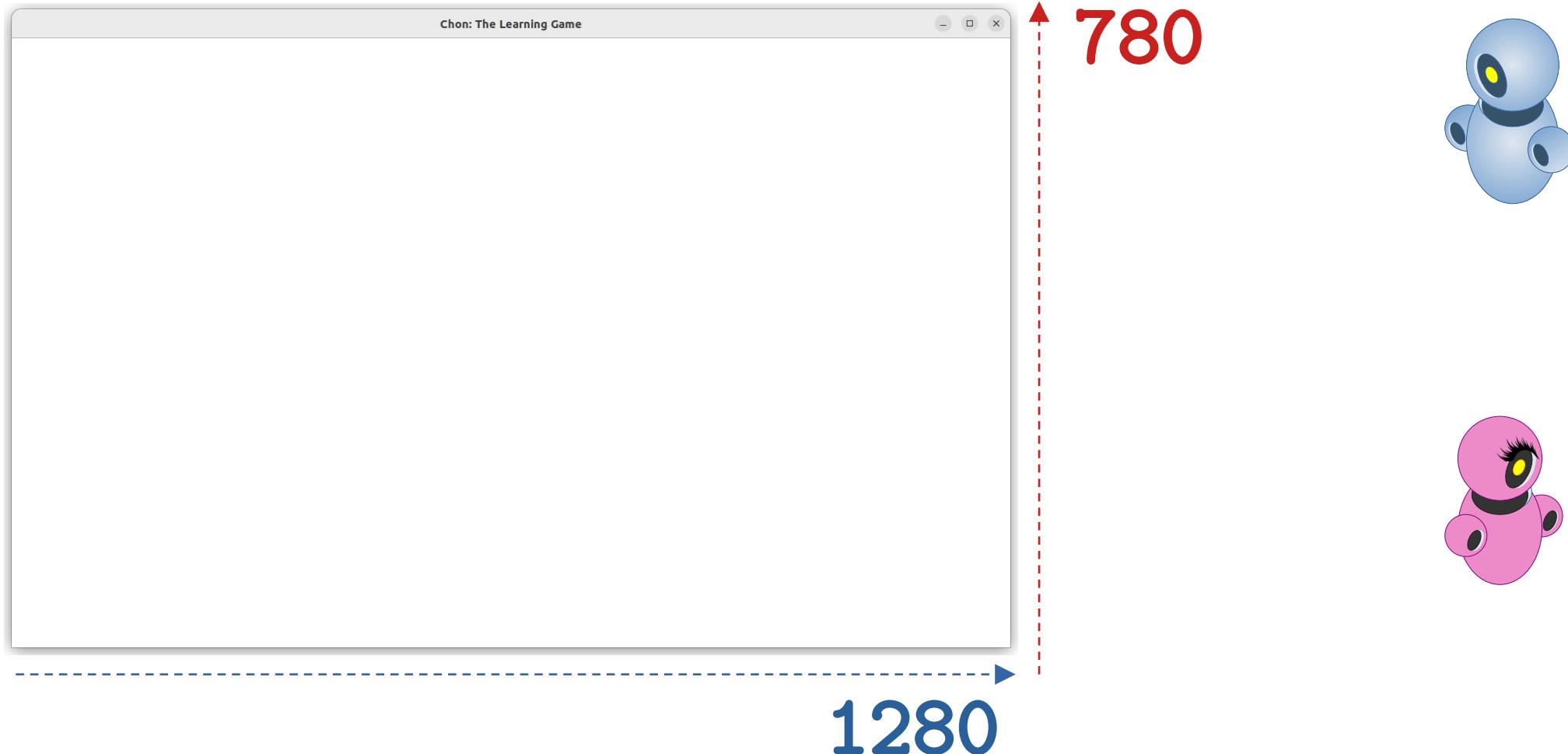
# Object's Dimension: Width and Height



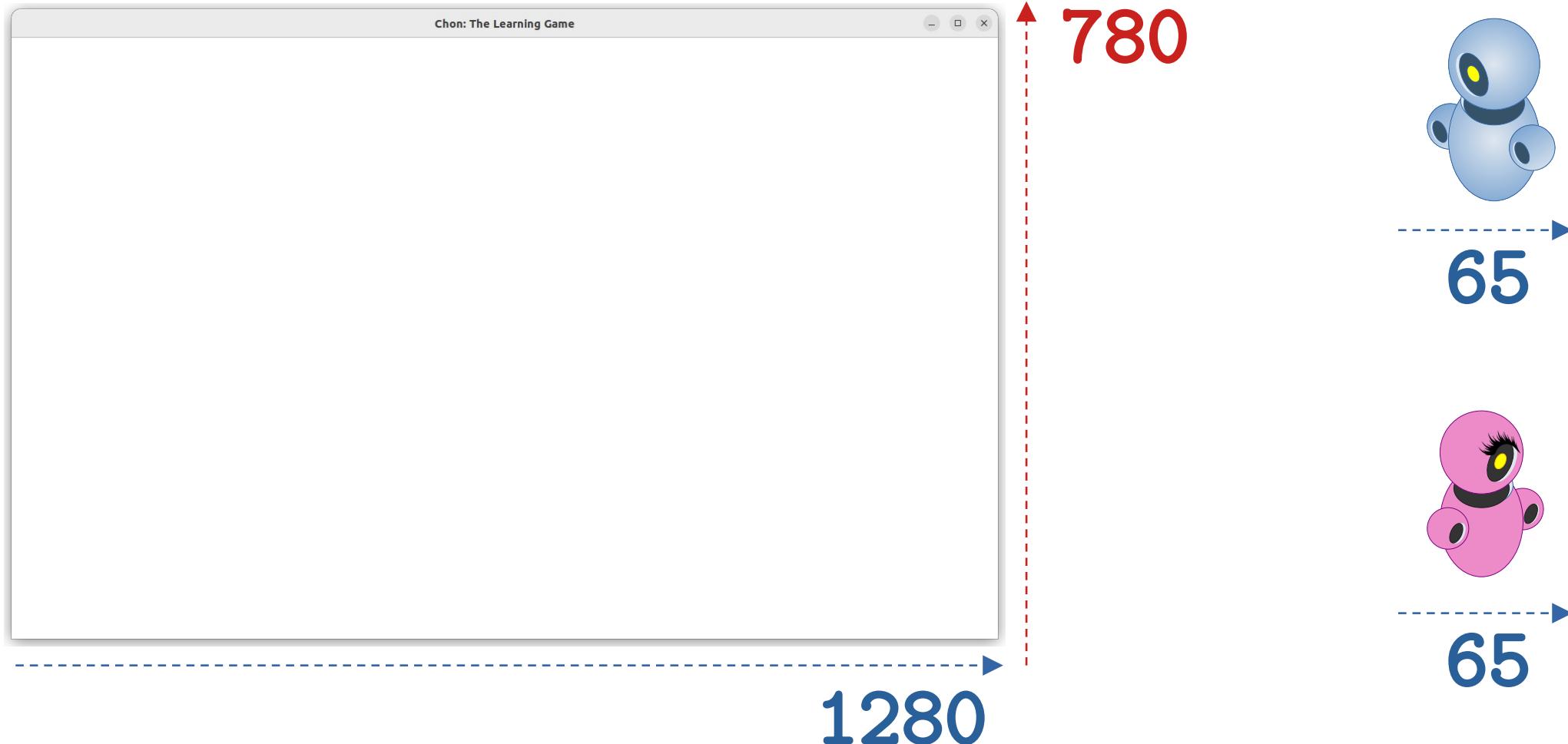
# Object's Dimension: Width and Height



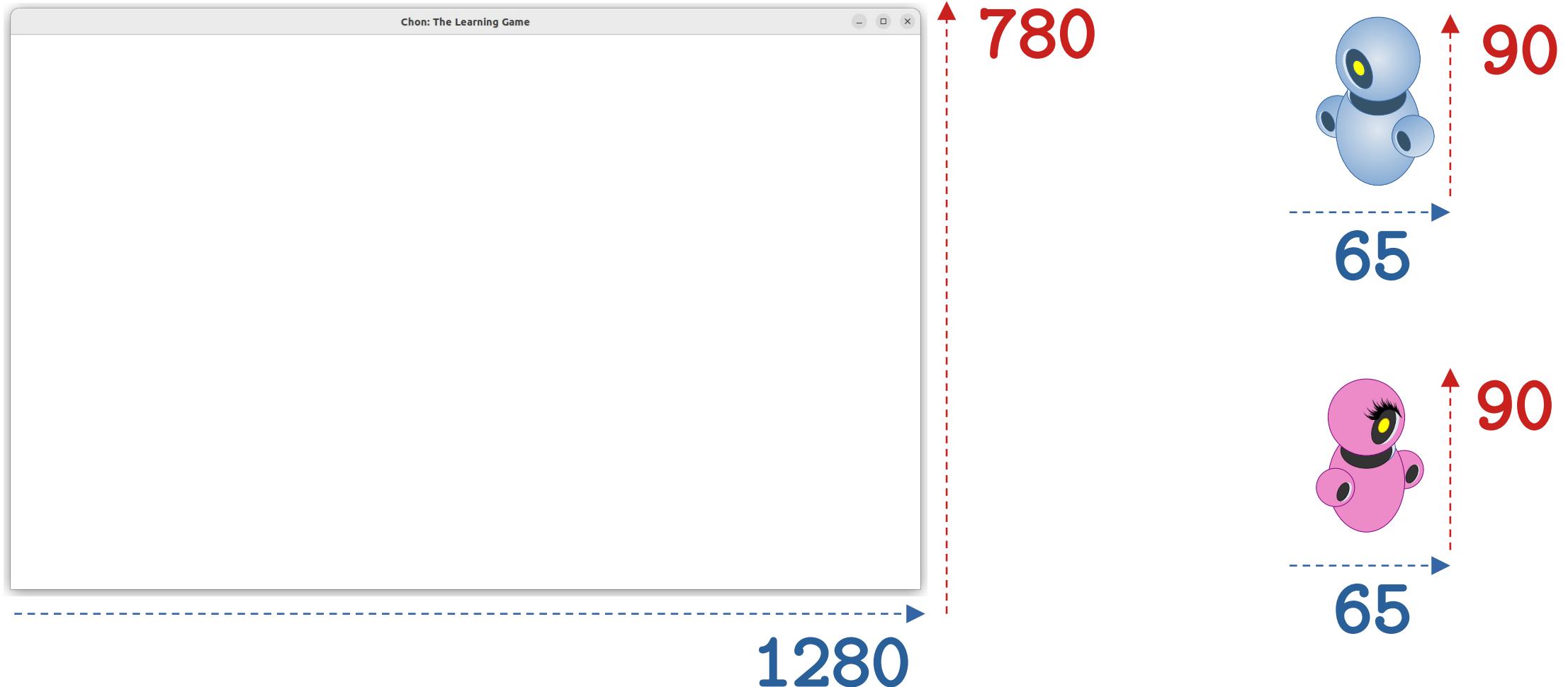
# Object's Dimension: Width and Height



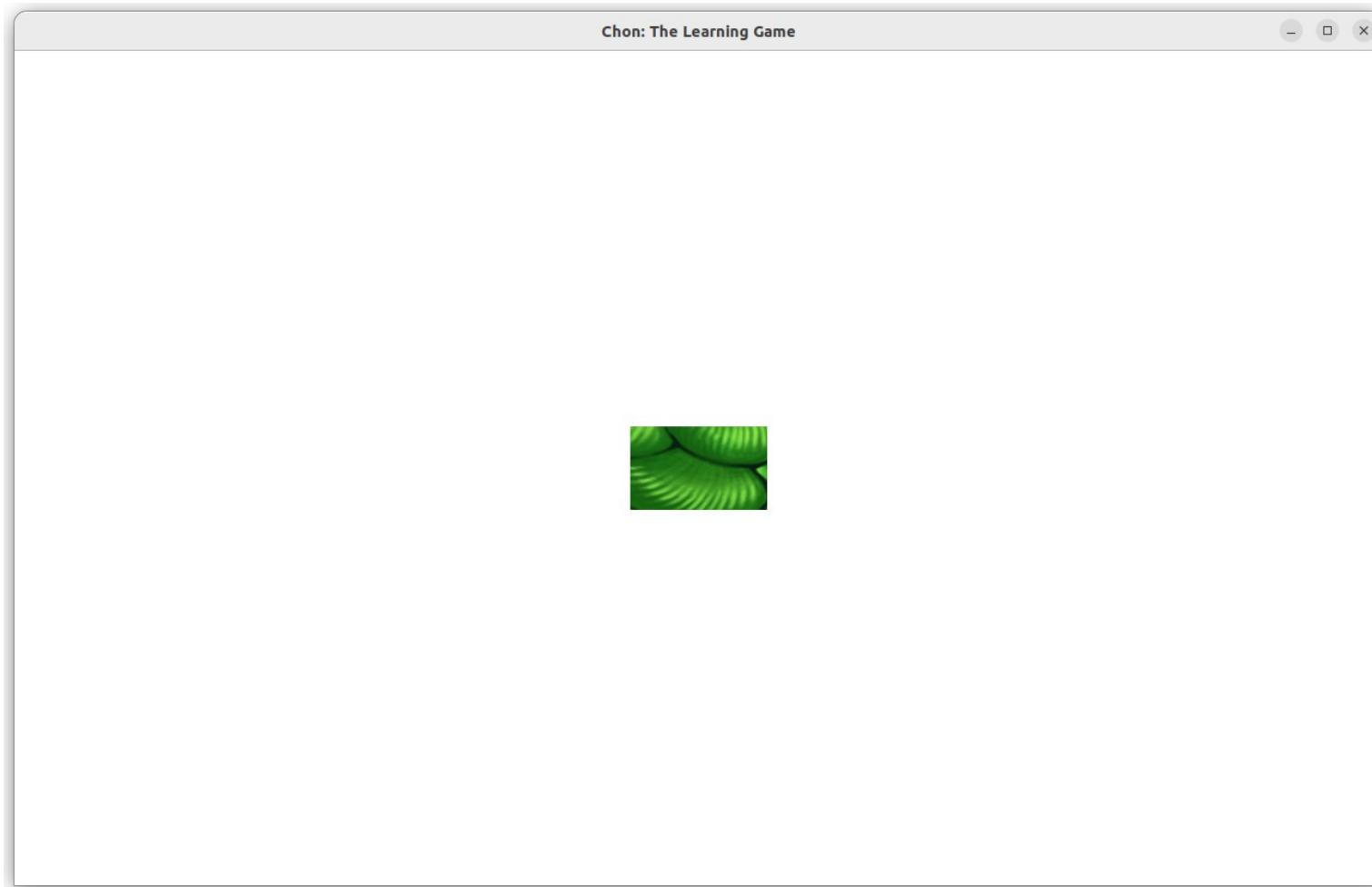
# Object's Dimension: Width and Height



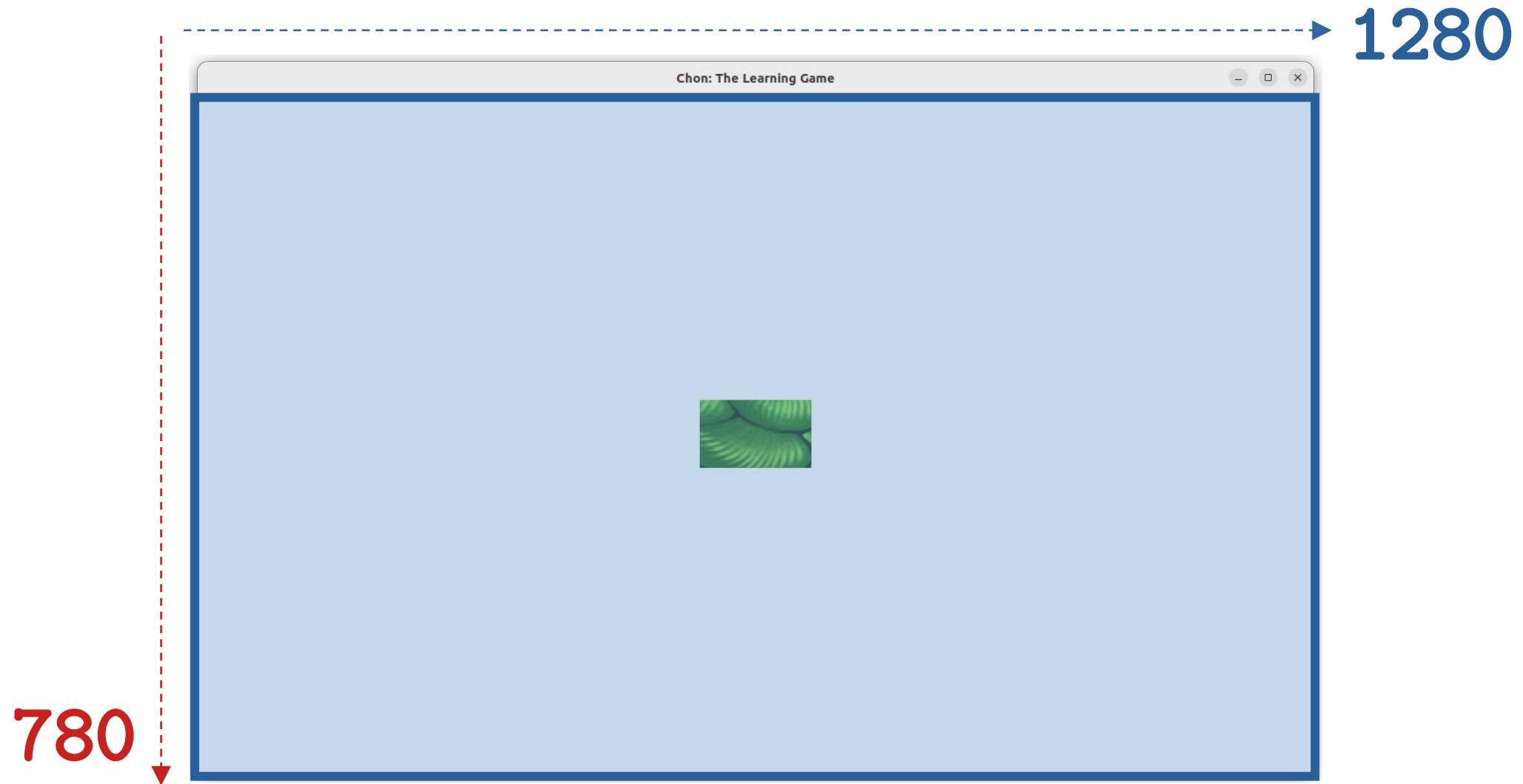
# Object's Dimension: Width and Height



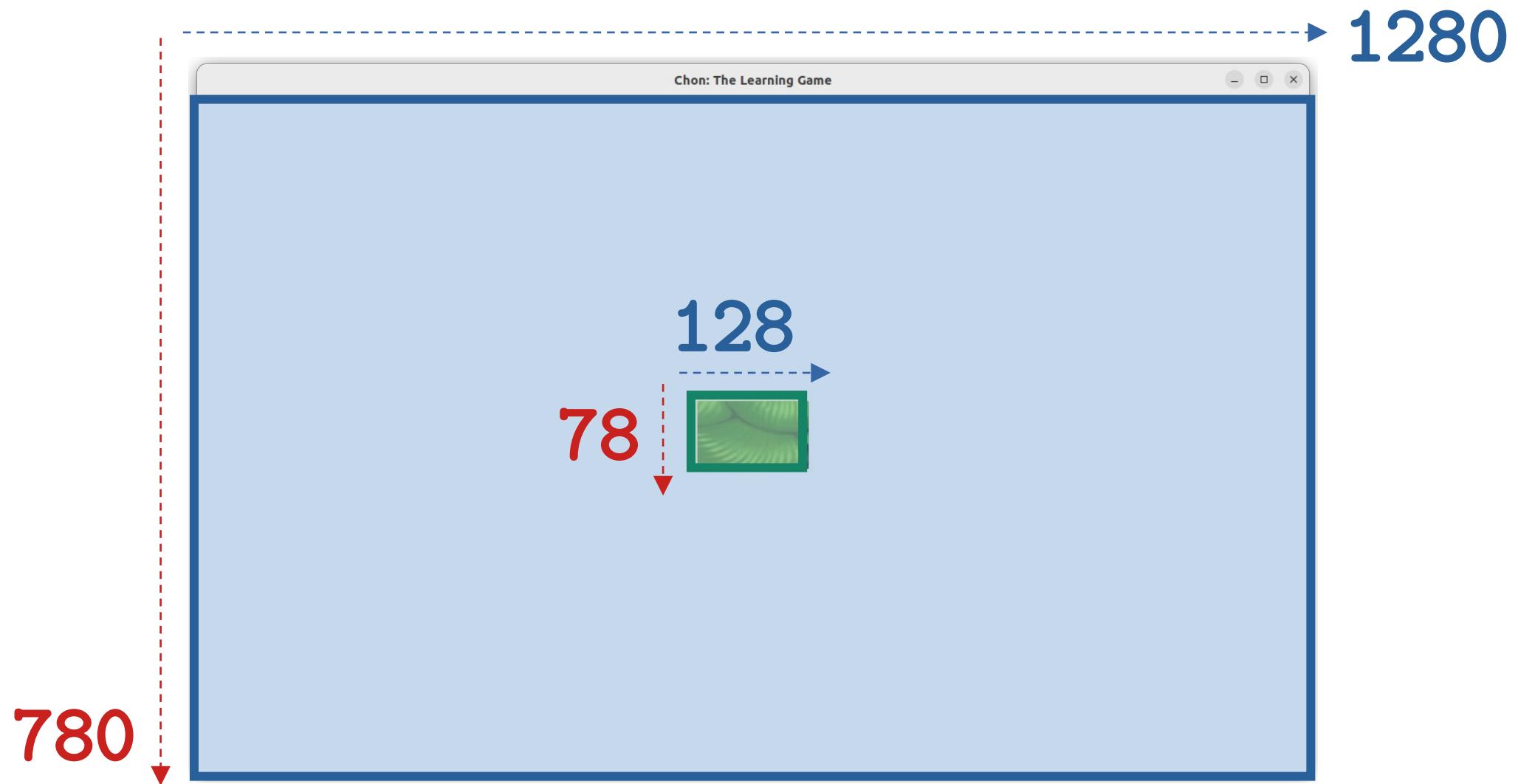
# Canvas Example



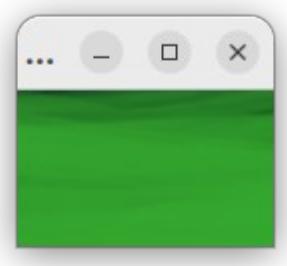
# Canvas Example



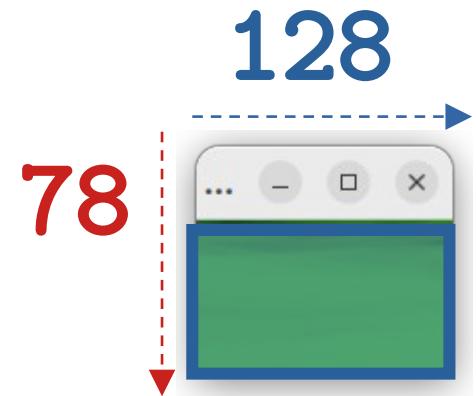
# Canvas Example



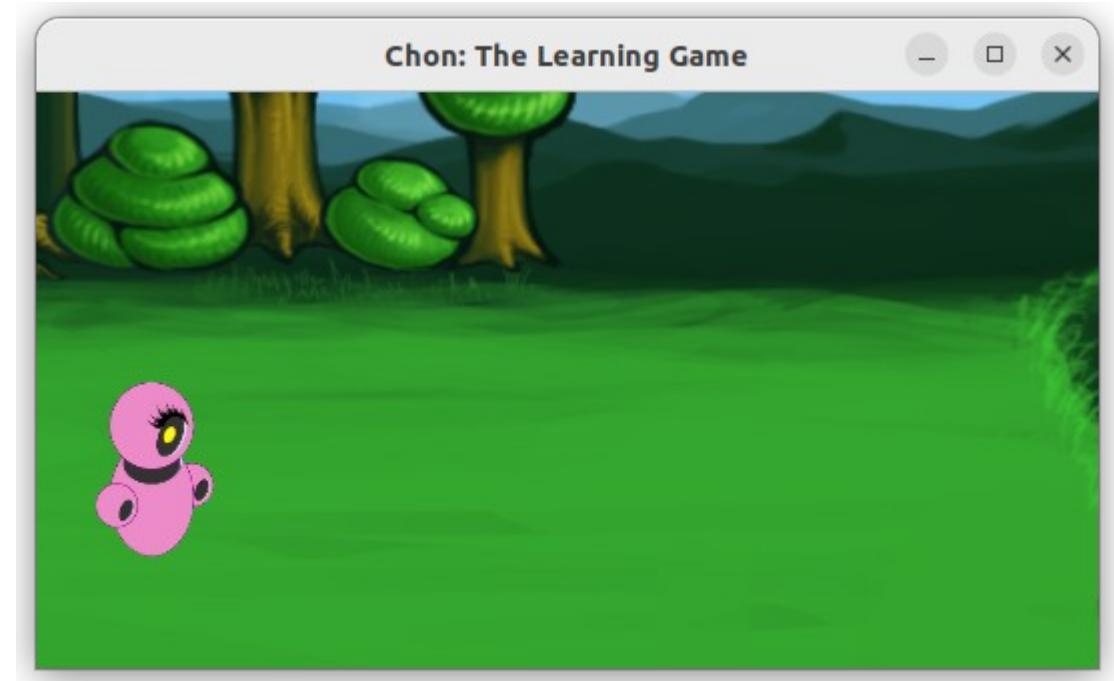
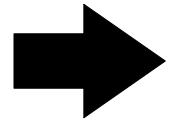
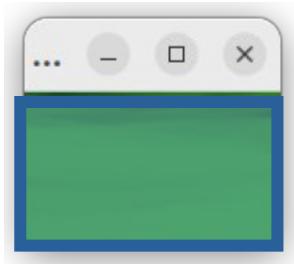
# Scene Example



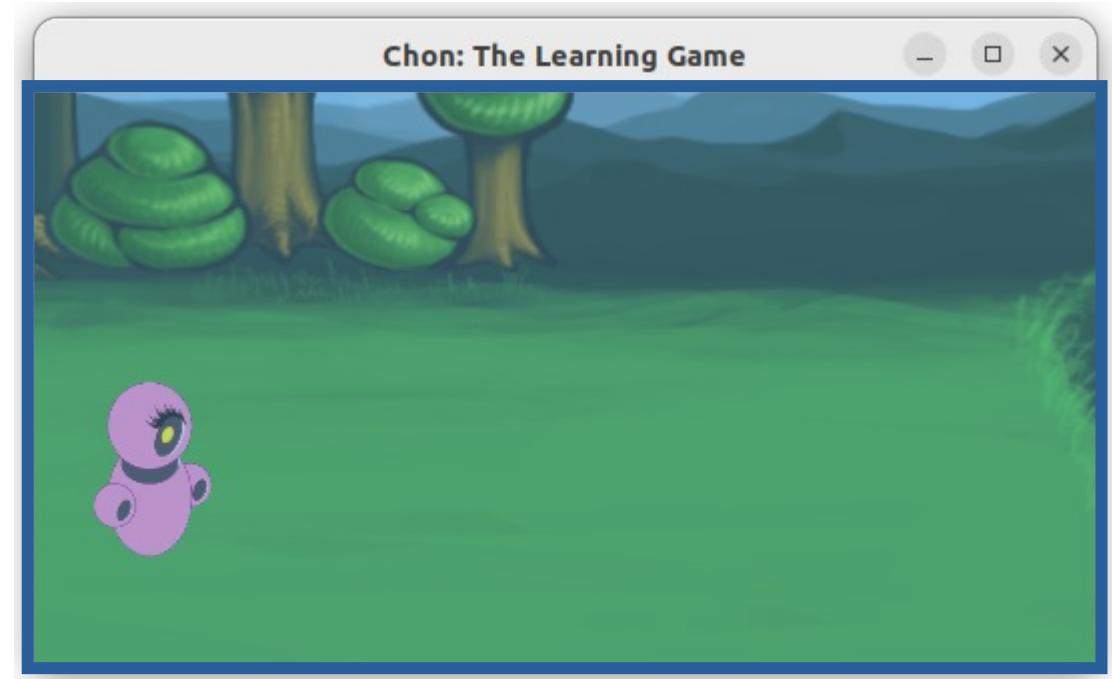
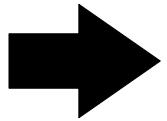
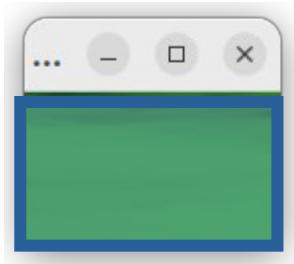
# Scene Example



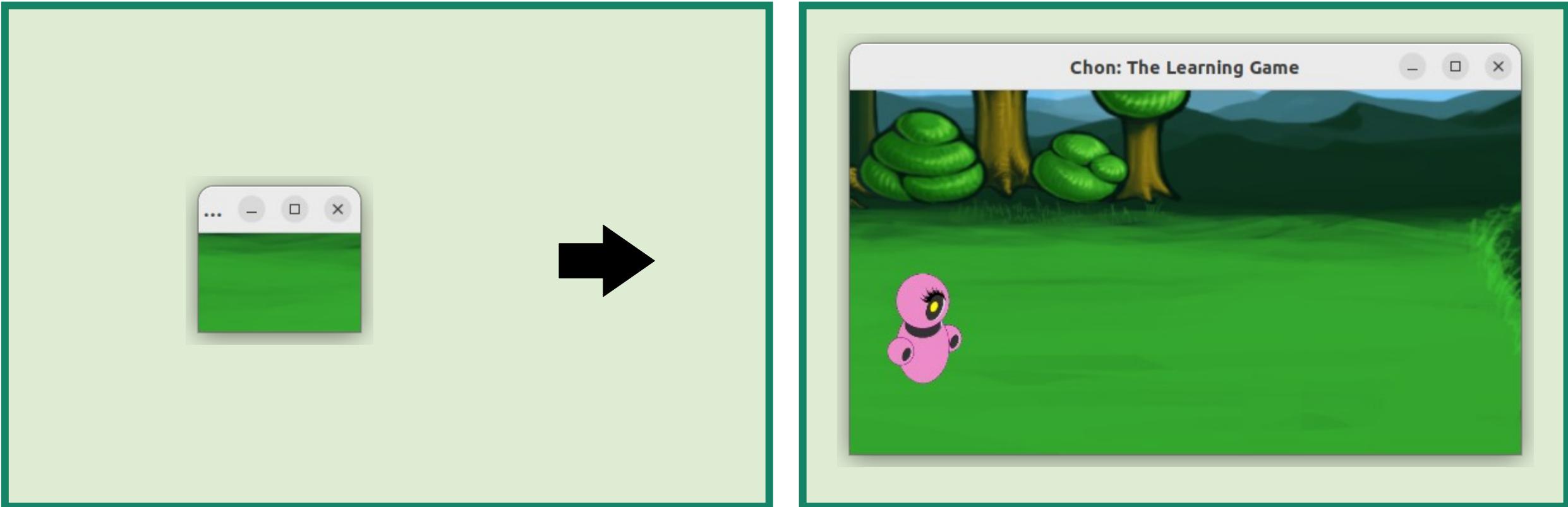
# Scene Example



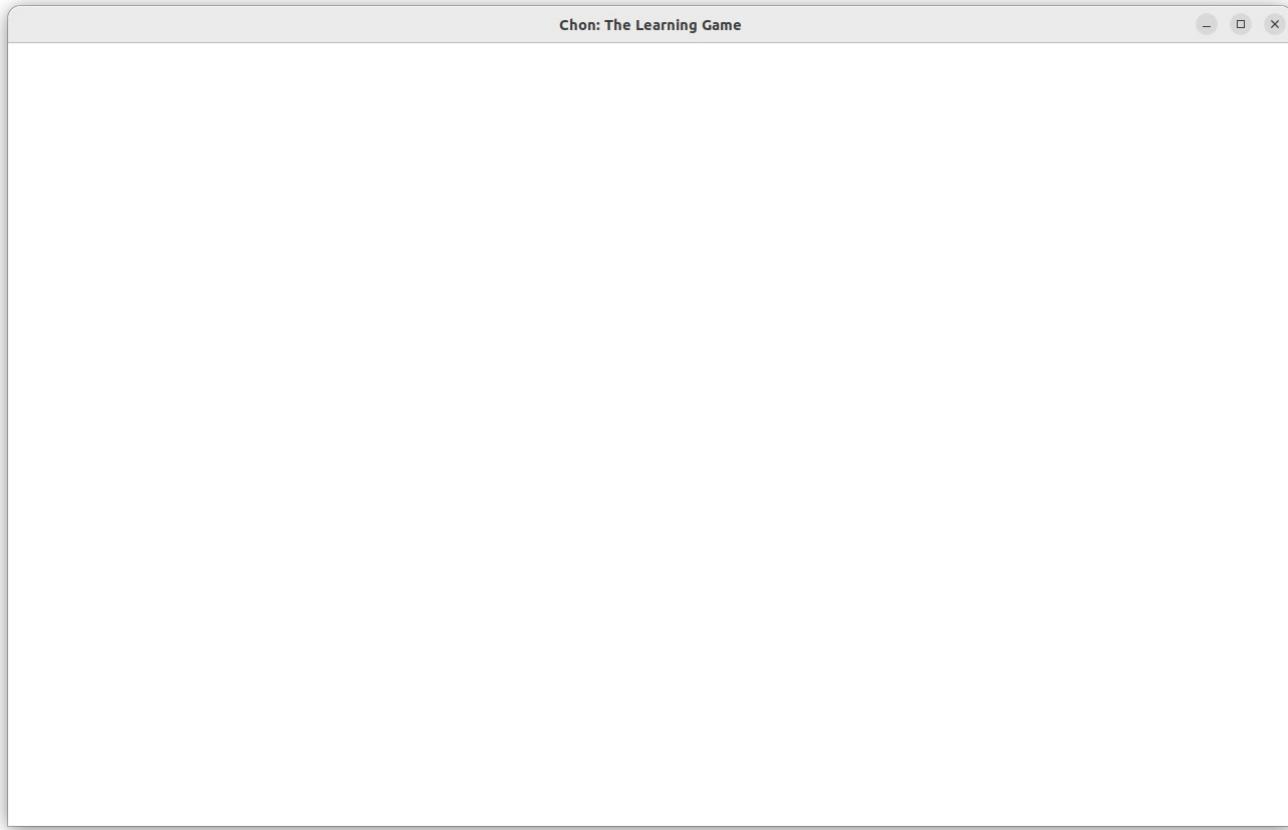
# Scene Example



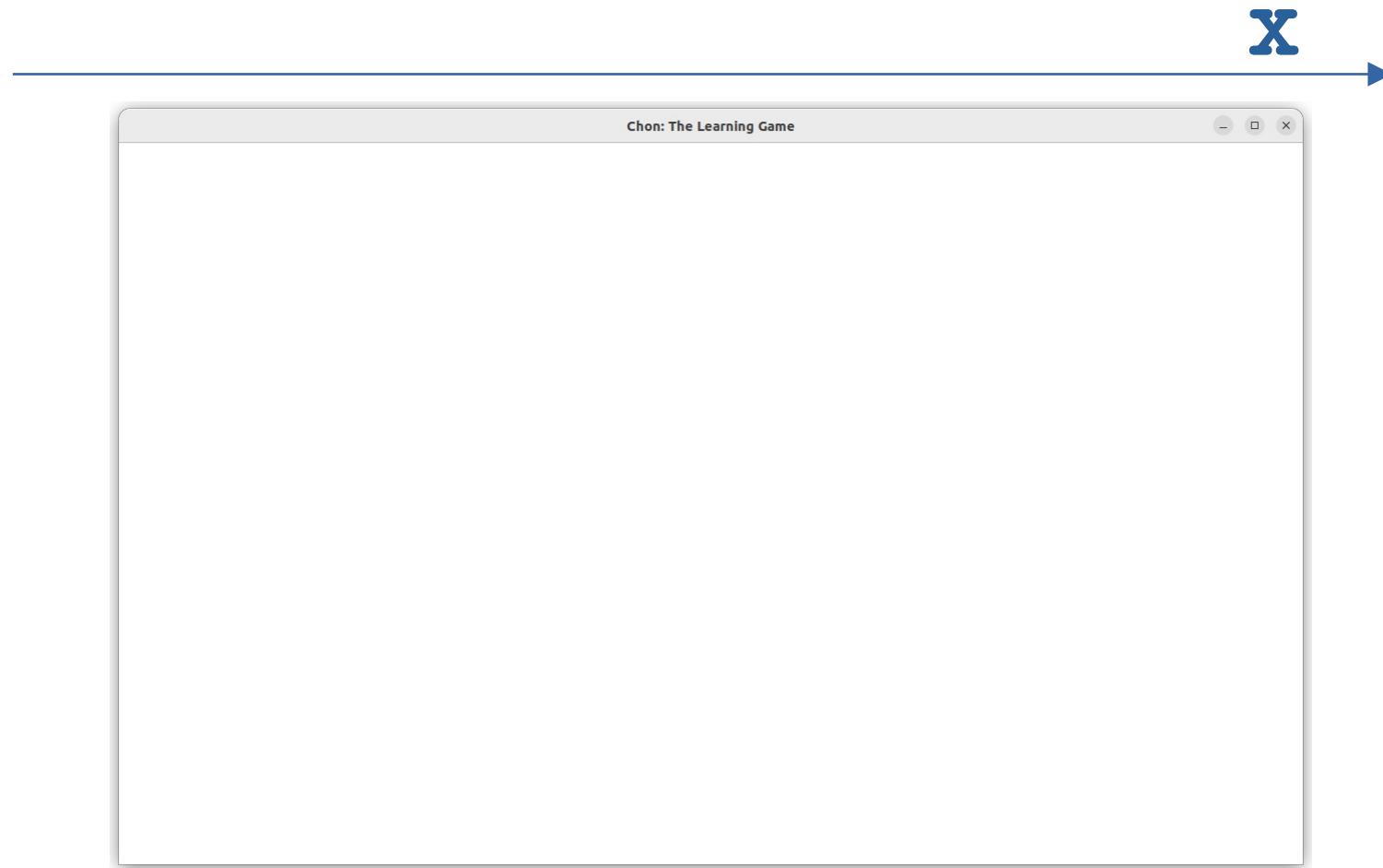
# Scene Example



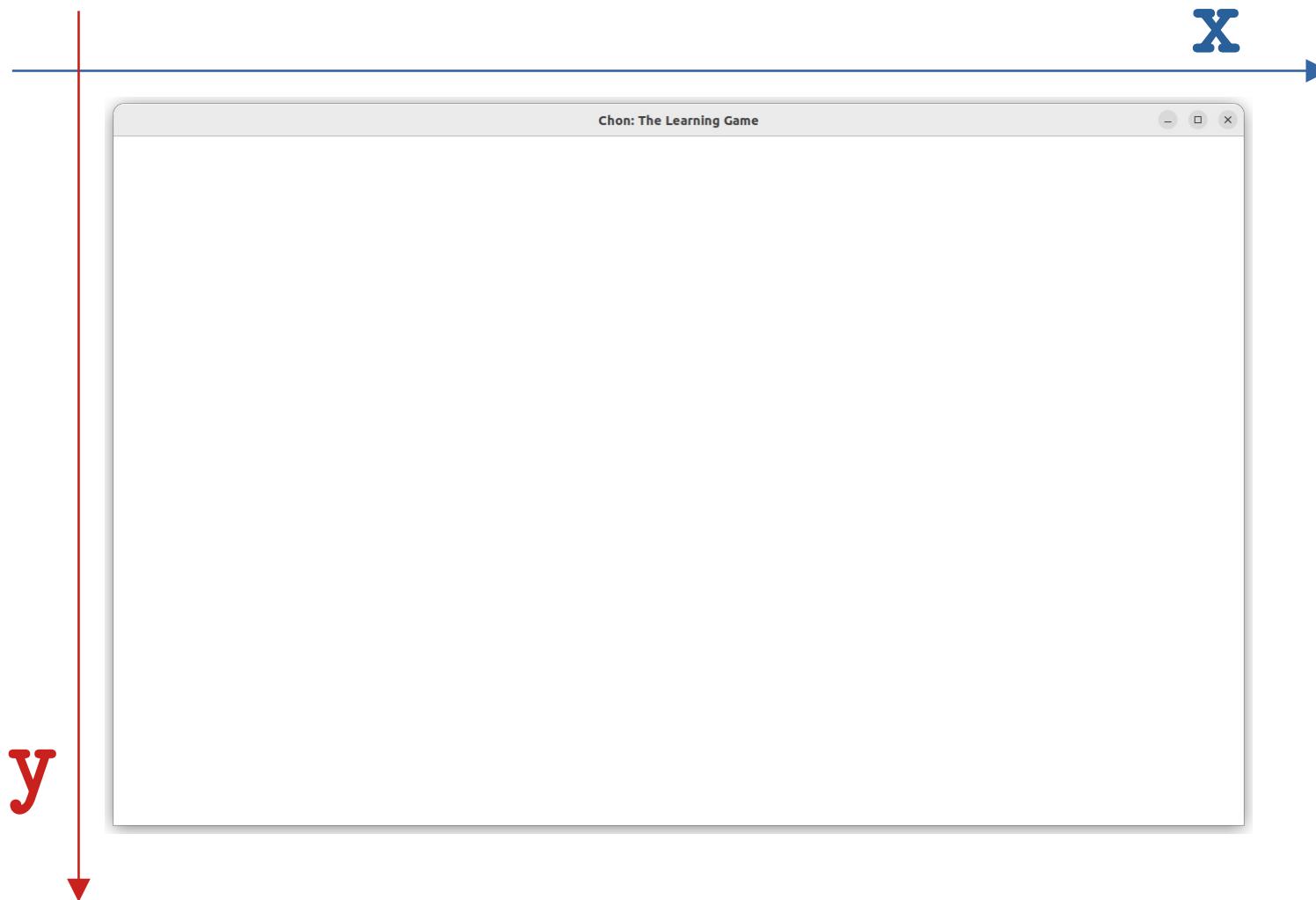
# Positioning System



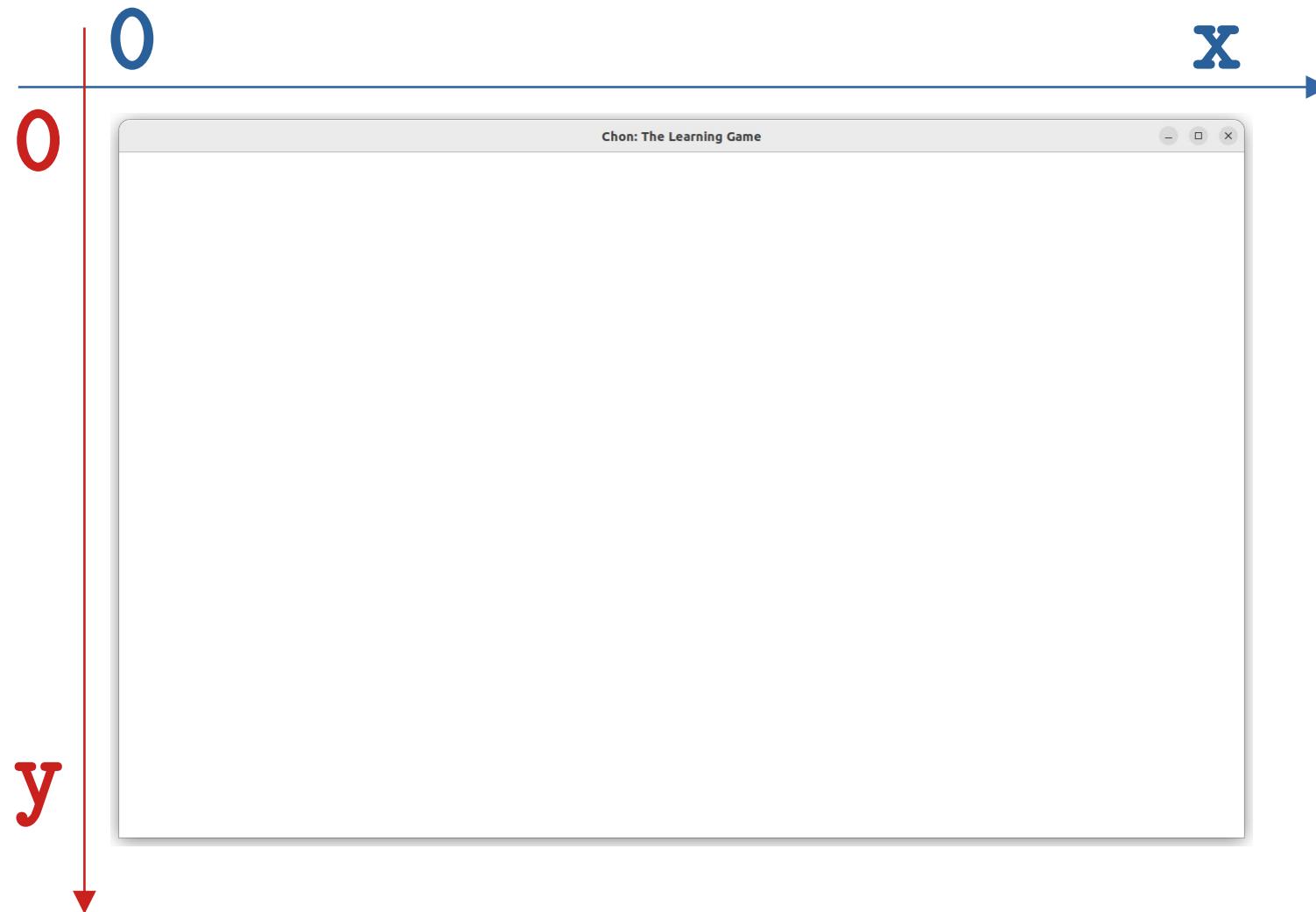
# Positioning System



# Positioning System



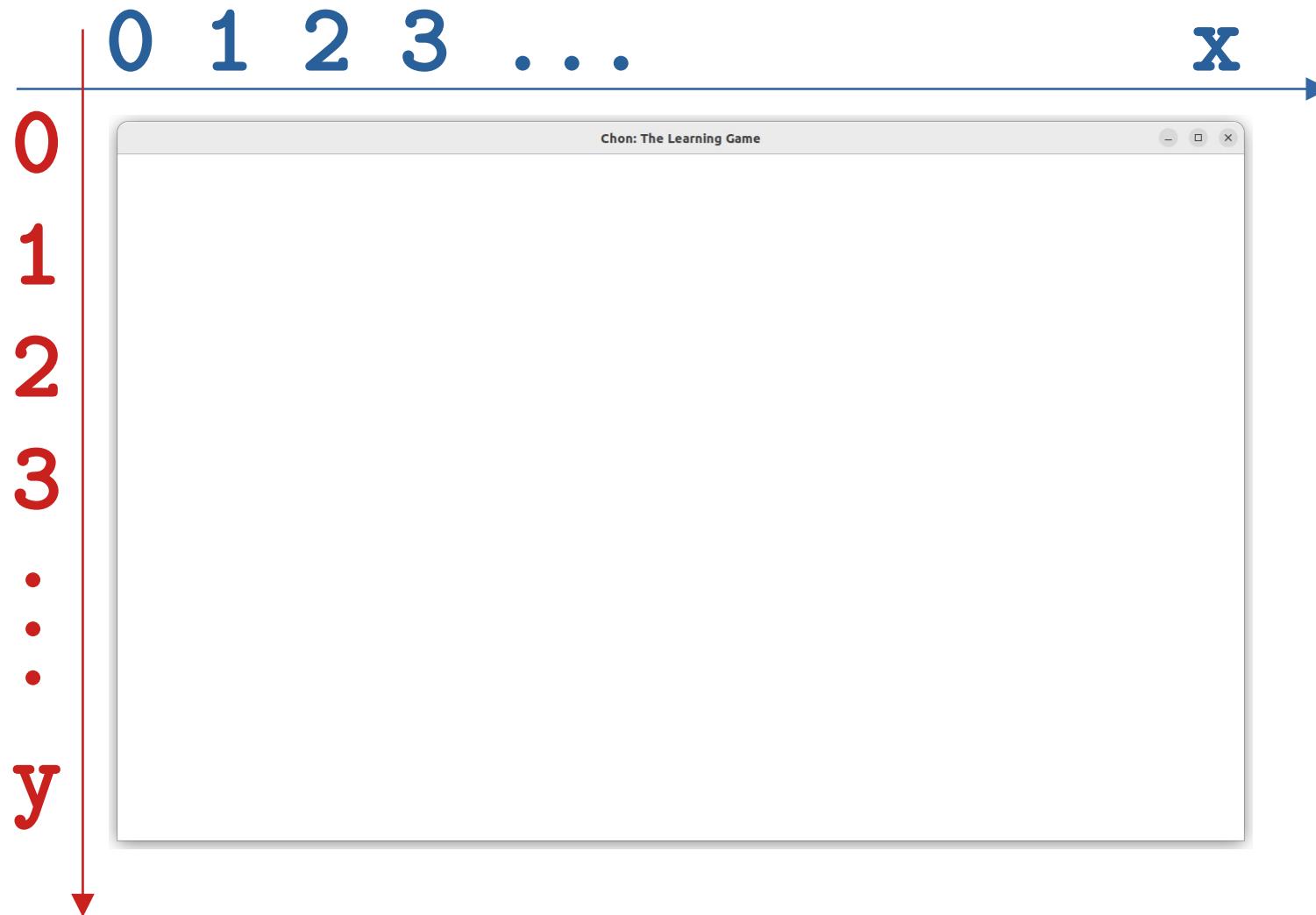
# Positioning System



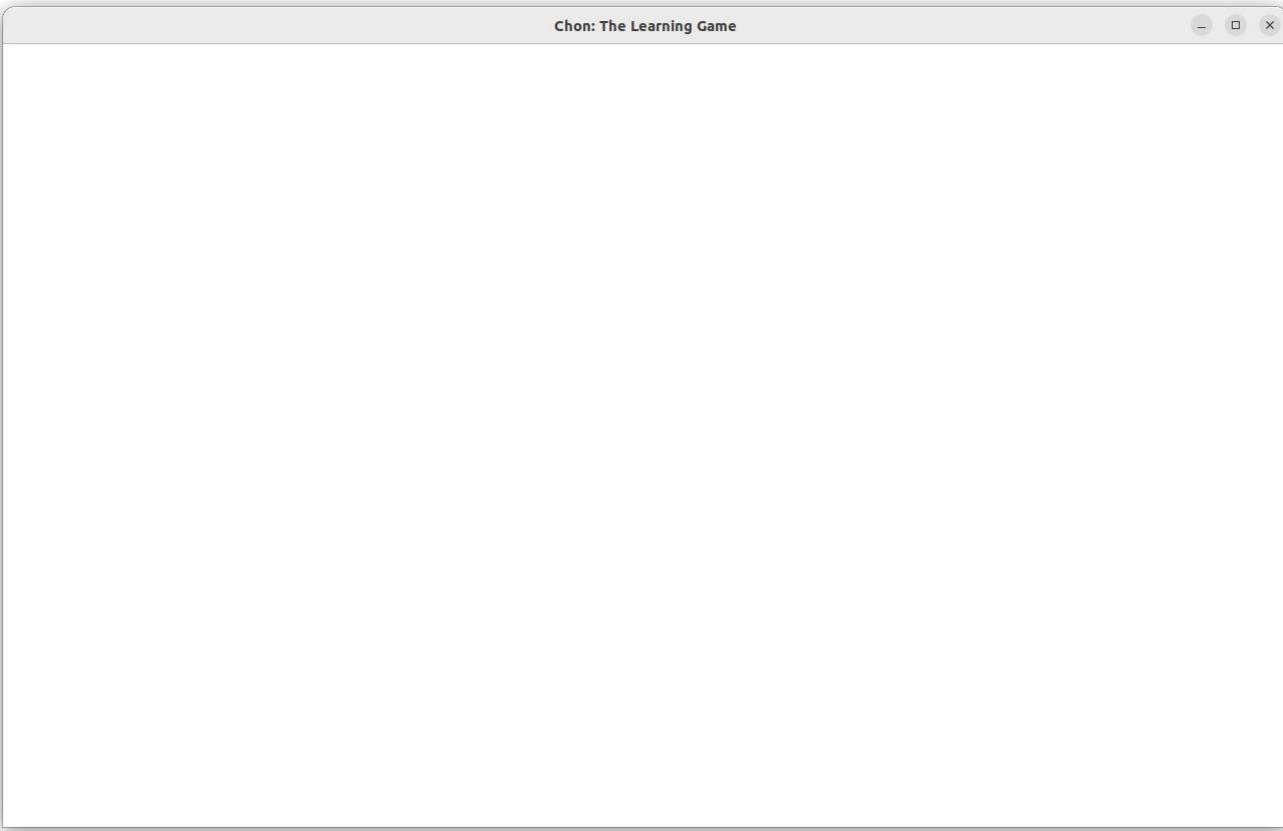
# Positioning System



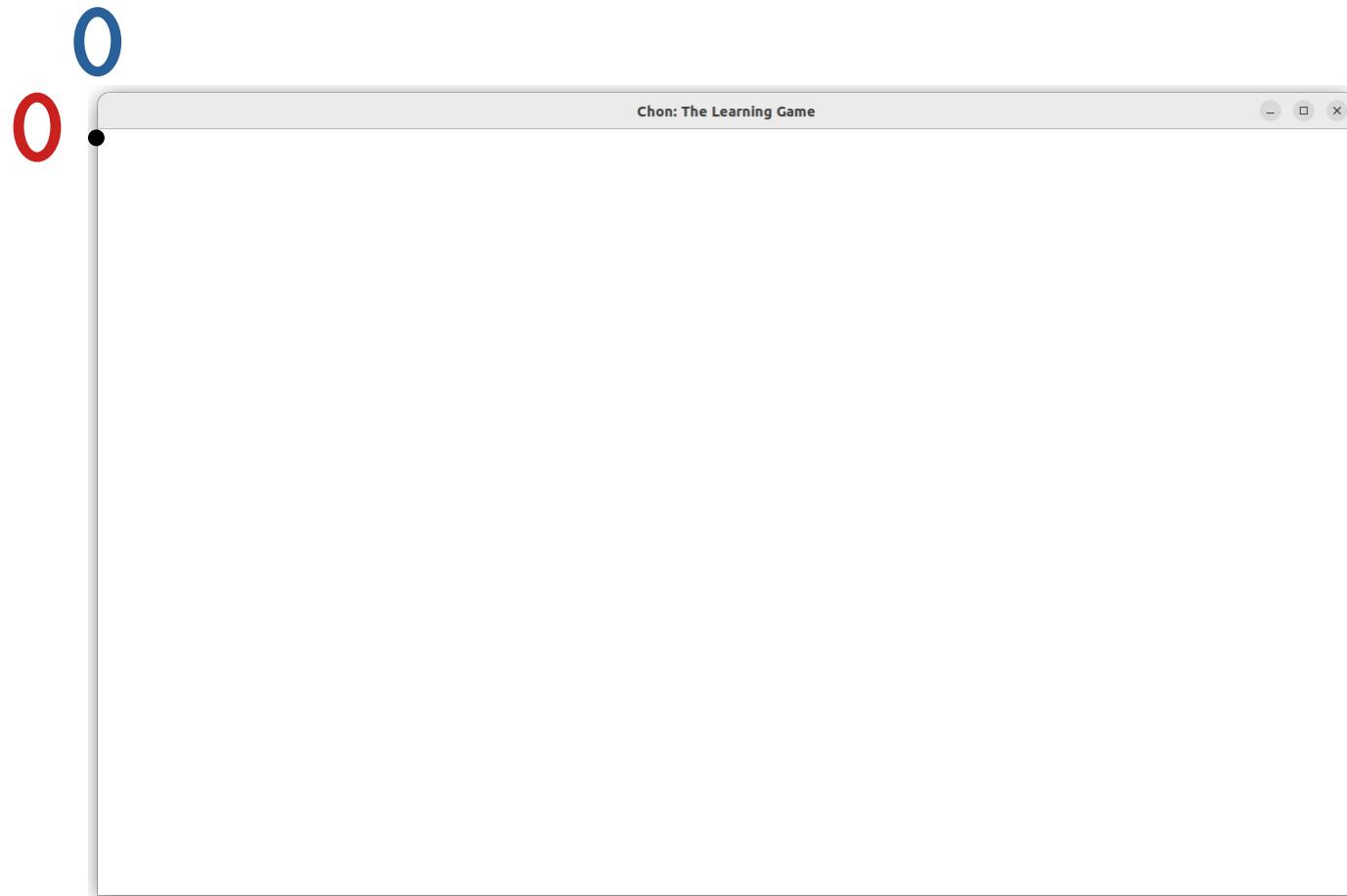
# Positioning System



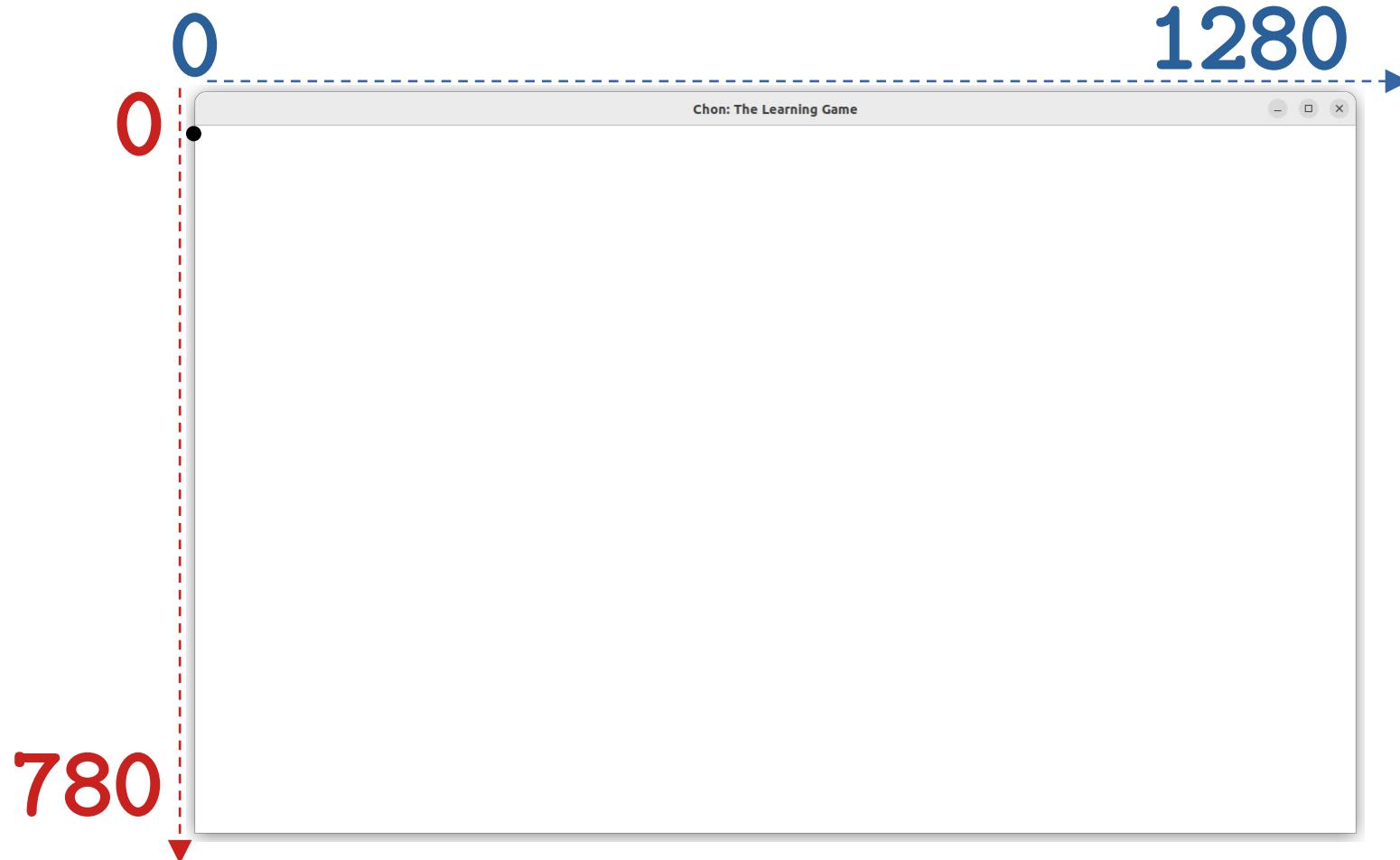
# Drawing the Background



# Drawing the Background



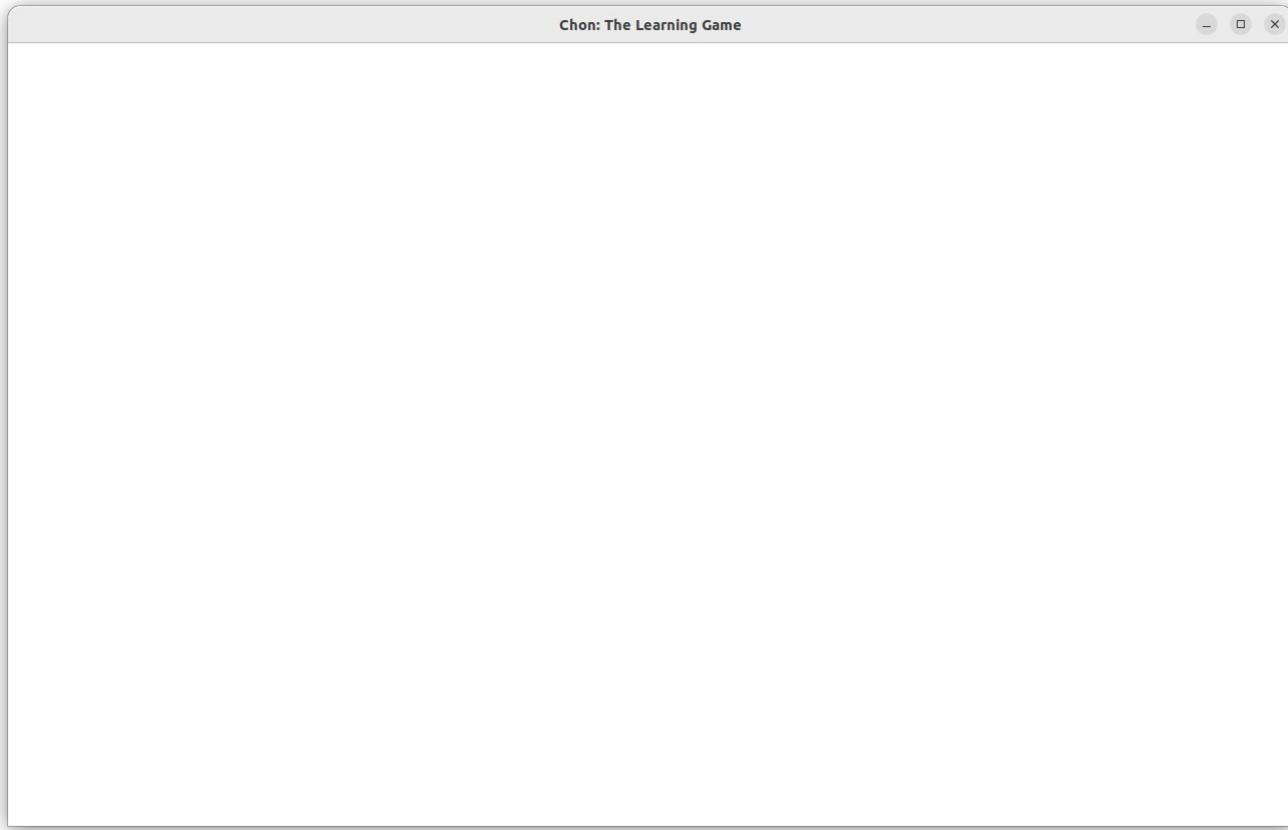
# Drawing the Background



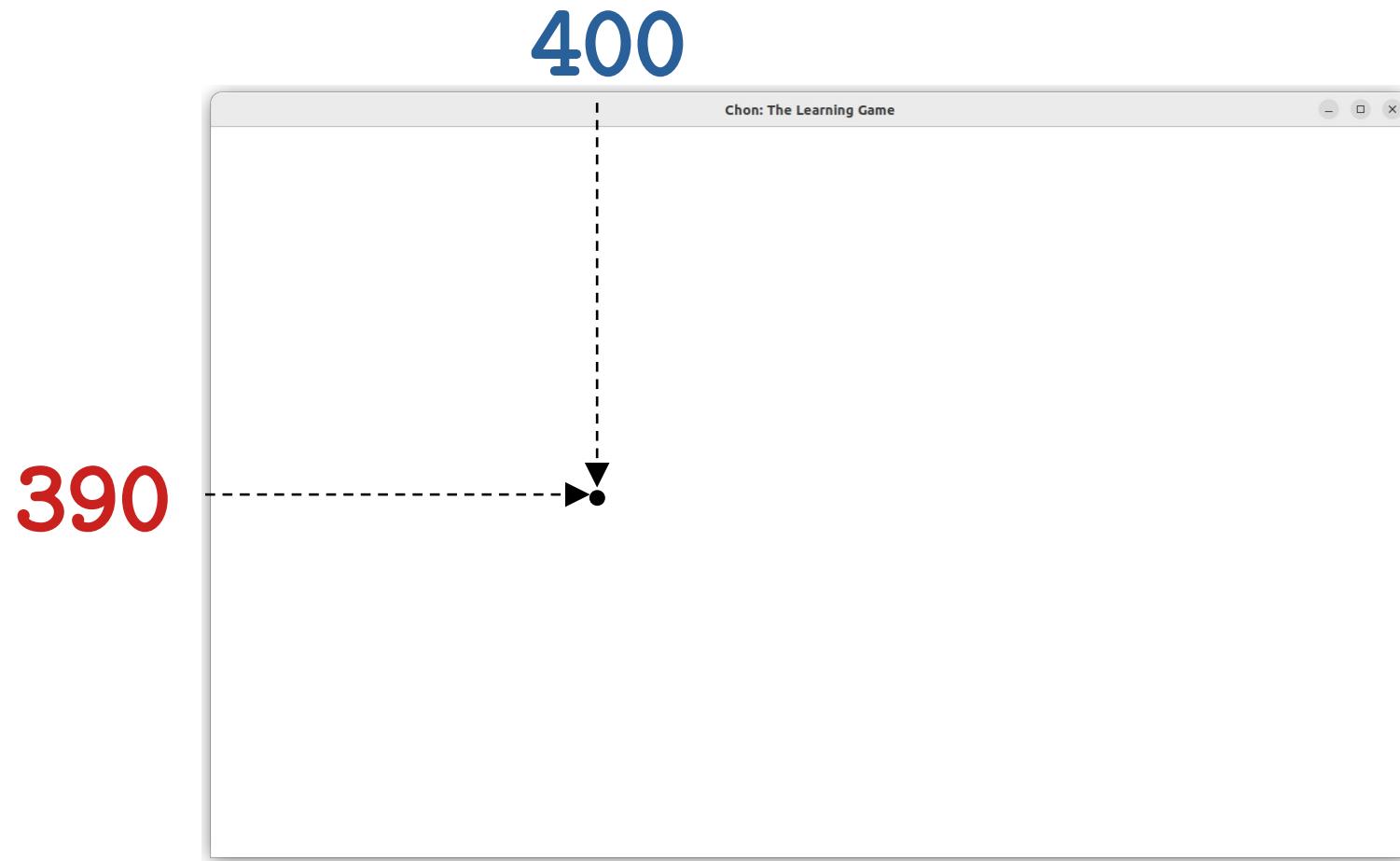
# Drawing the Background



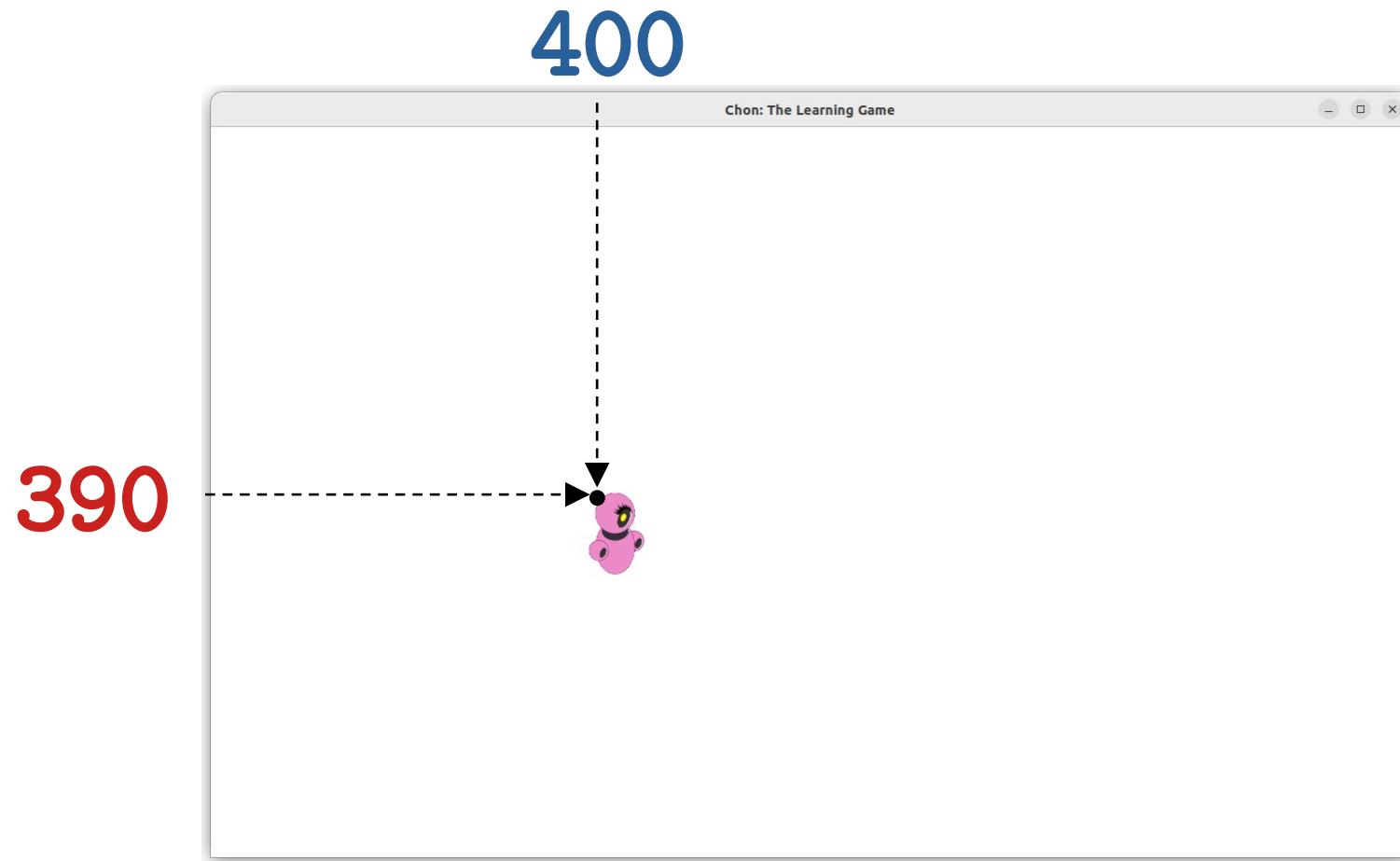
# Drawing the Characters



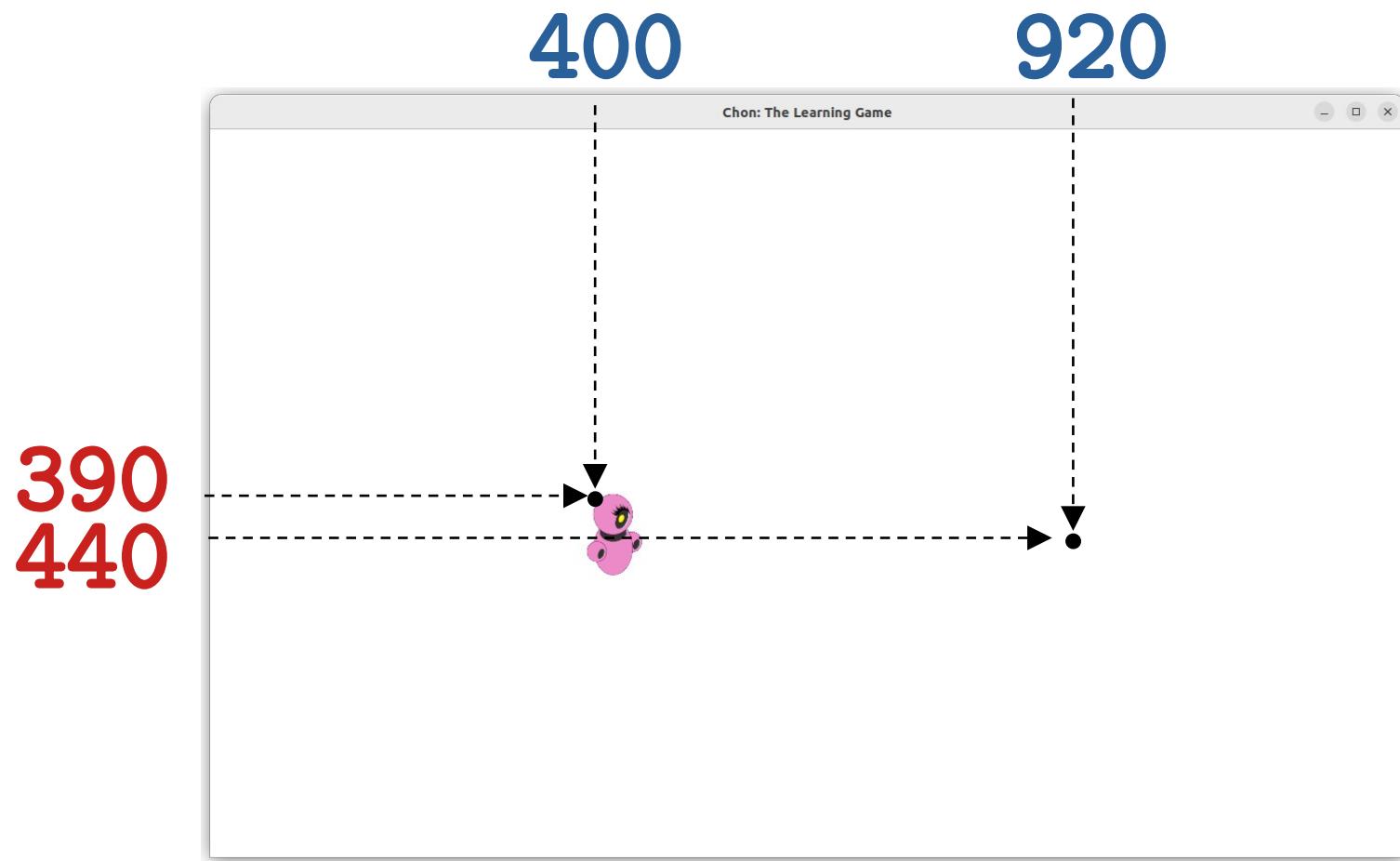
# Drawing the Characters



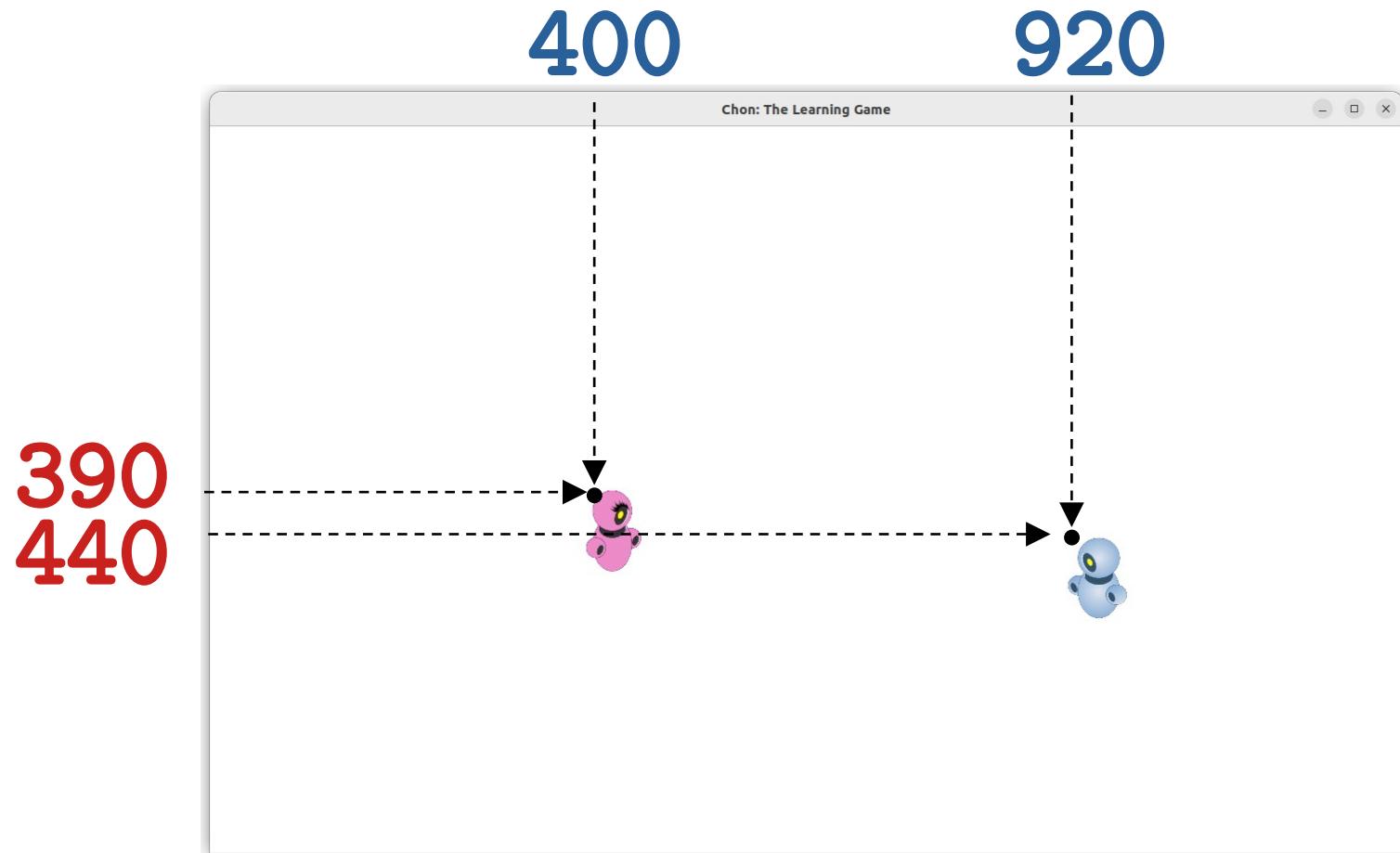
# Drawing the Characters



# Drawing the Characters



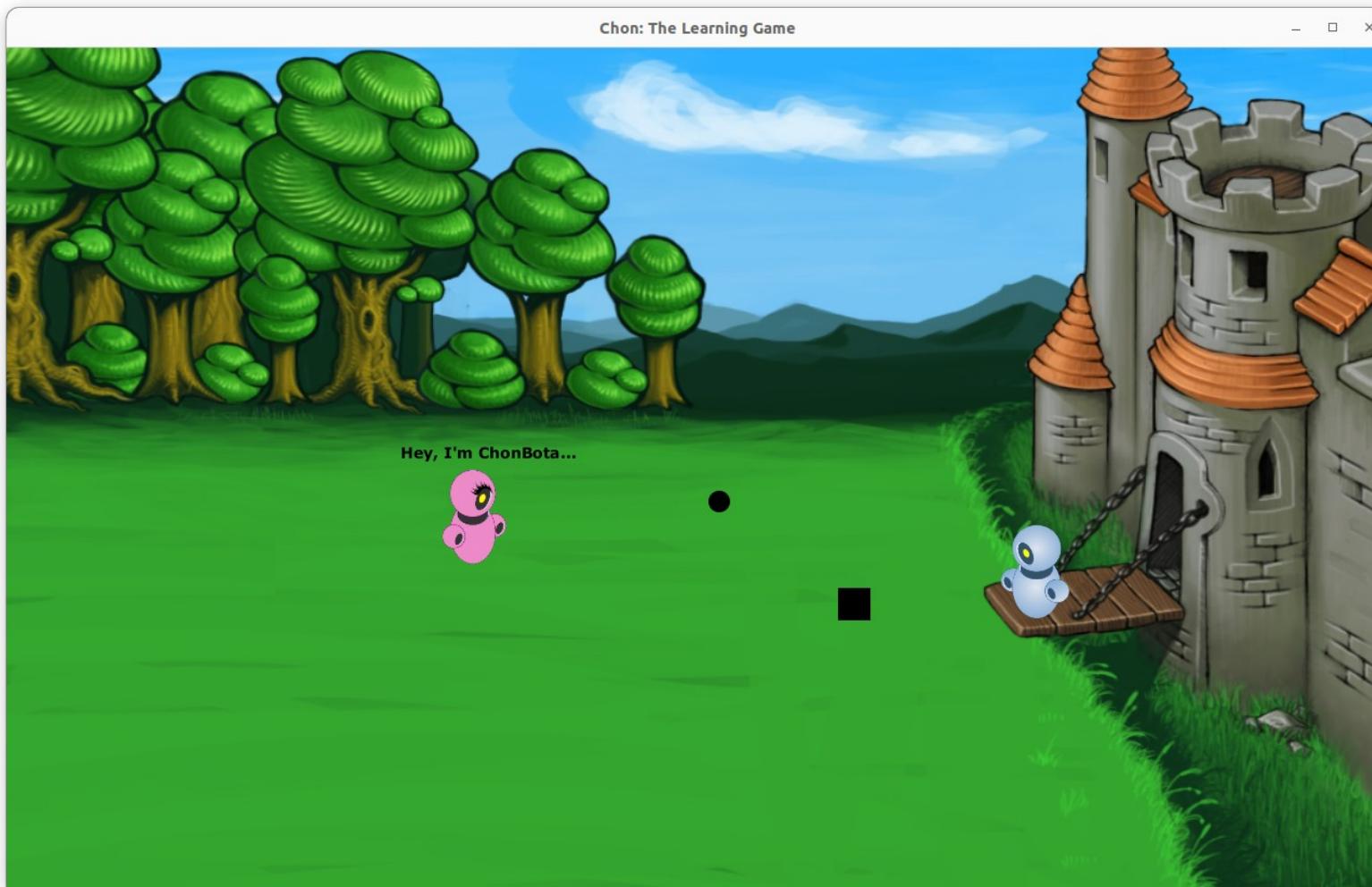
# Drawing the Characters



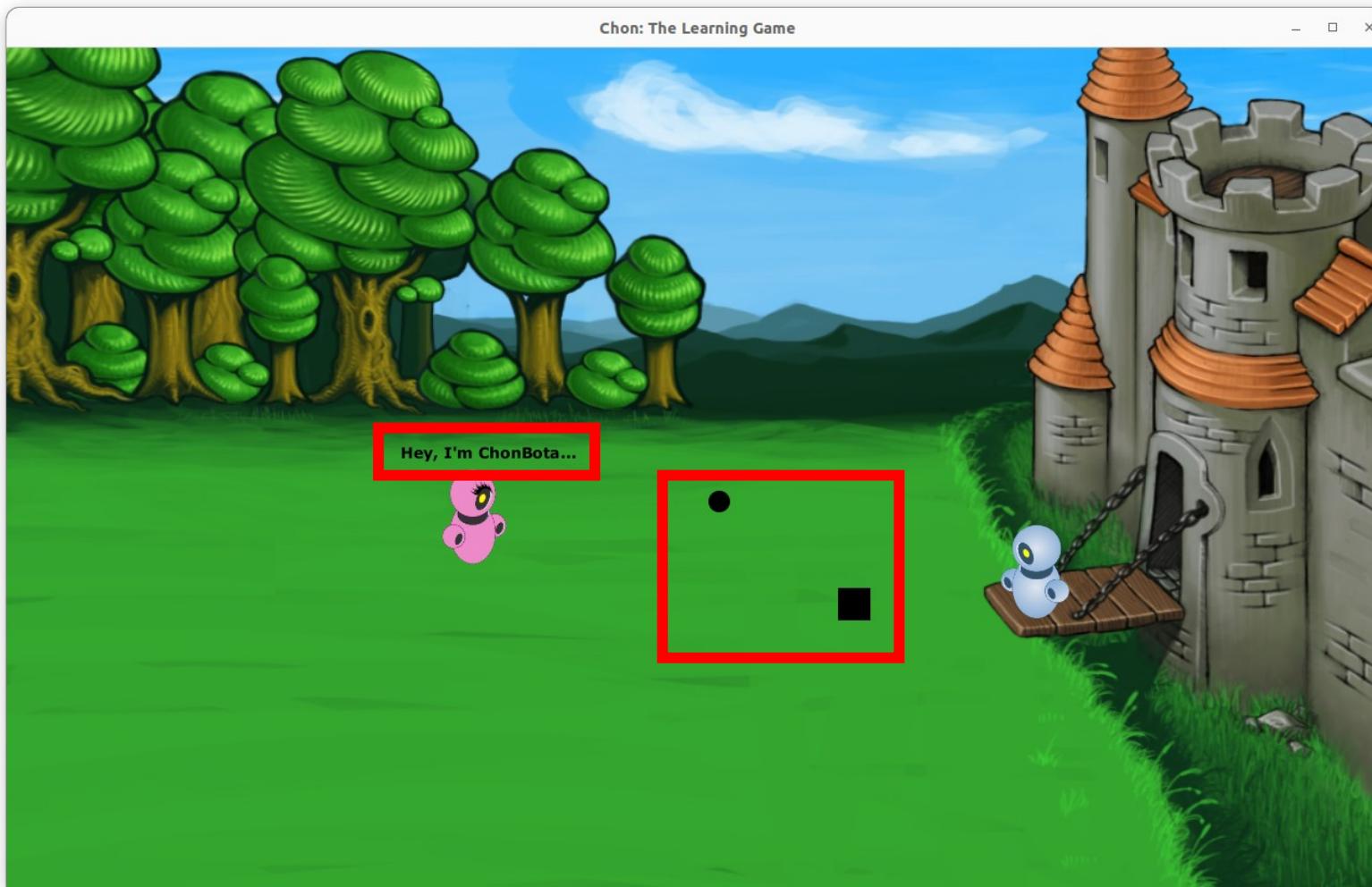
# Drawing the Characters



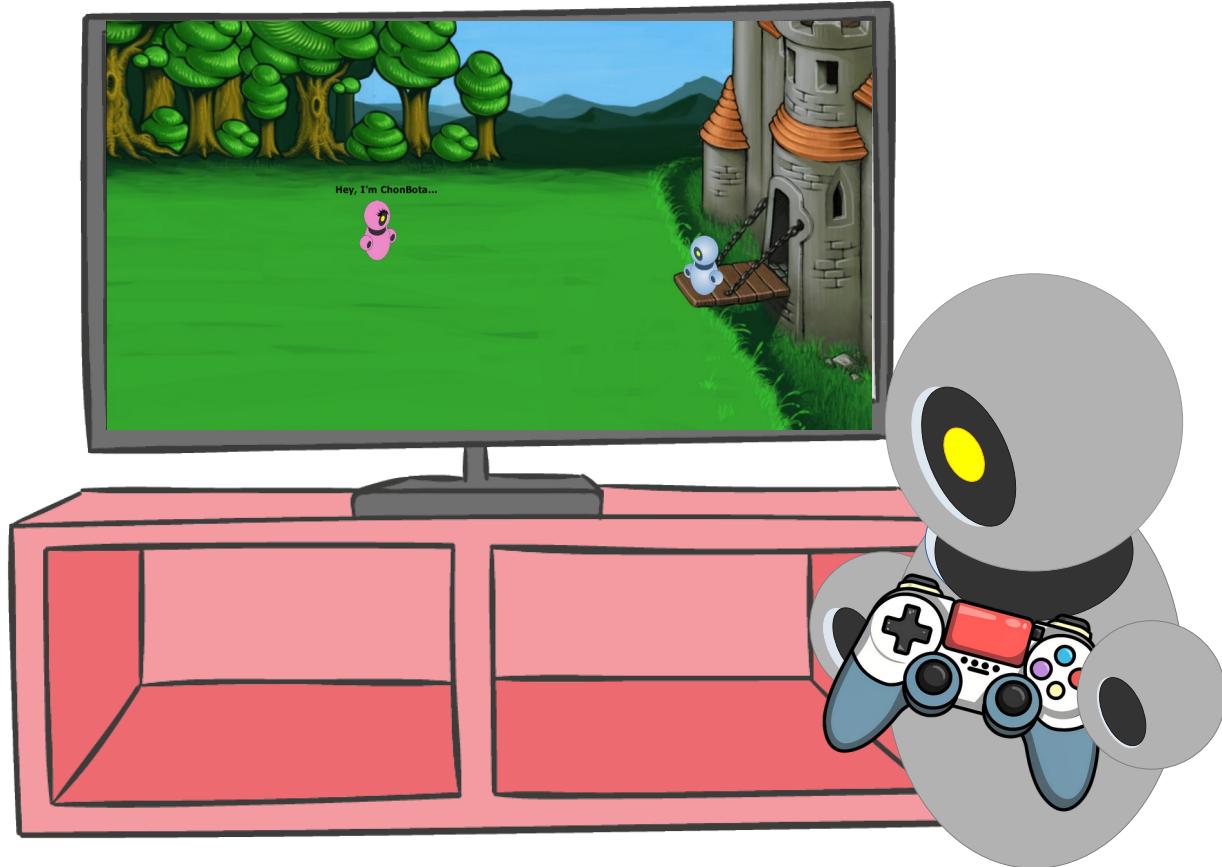
# Drawing Other Objects



# Drawing Other Objects



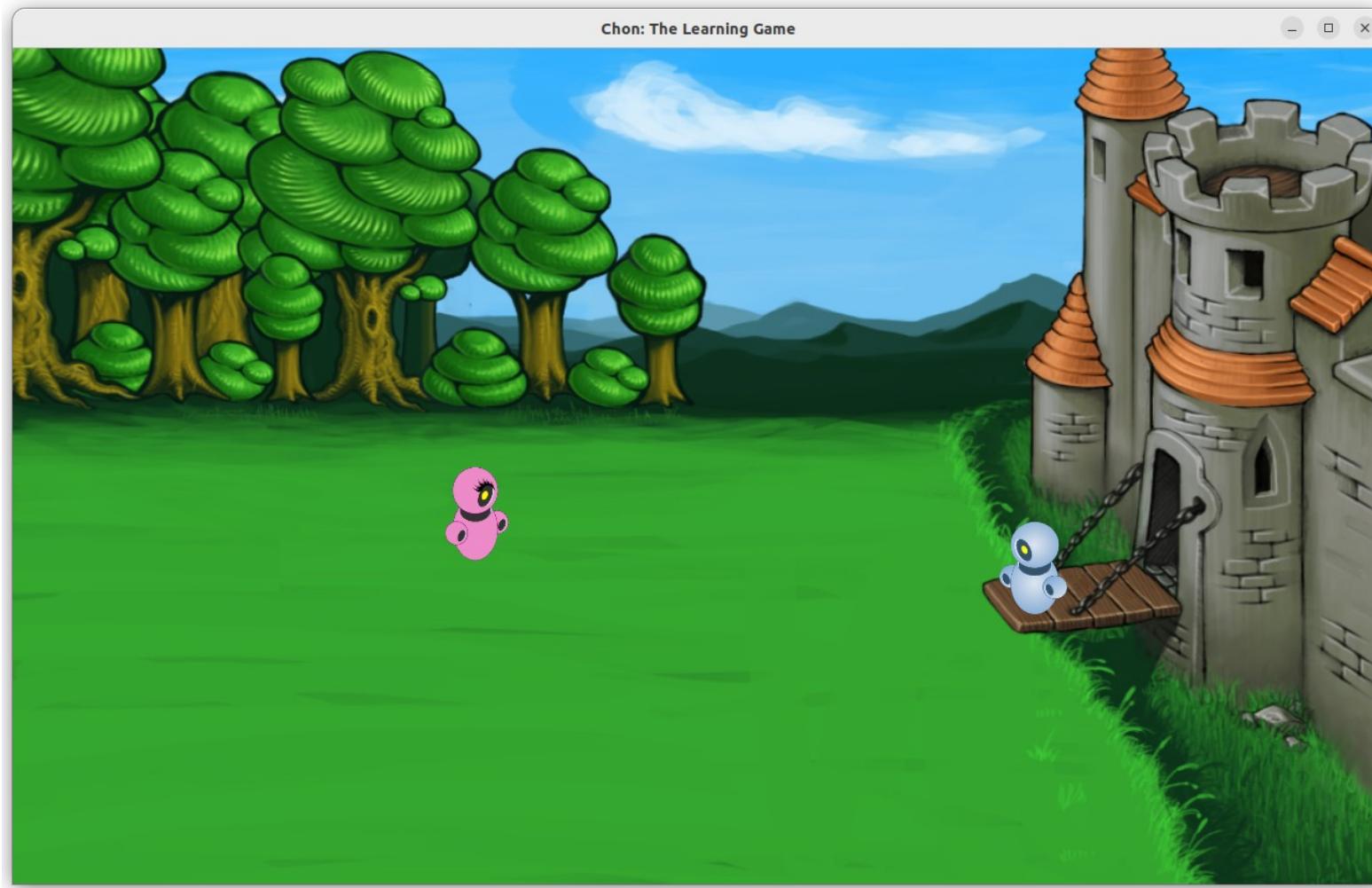
# MOVING OBJECTS IN THE CANVAS



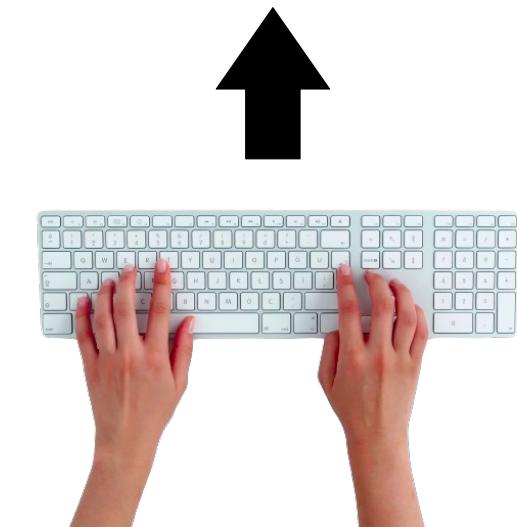
# Scene's Events

In JavaFX, **Events** represent interactions or changes that happen while running a JavaFX application, such as **user inputs** (mouse clicks, key presses), **window actions** (resizing, closing), or **internal changes** (animation updates).

# Events



# Events

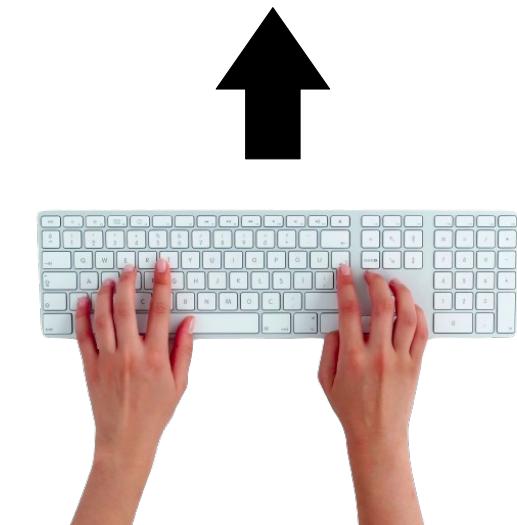


# Events



- Pressed: UP  
Released: UP
- Pressed: DOWN  
Released: DOWN
- Pressed: RIGHT  
Released: RIGHT
- Pressed: SPACE  
Released: SPACE
- Pressed: UP  
Released: UP

□



# Animation

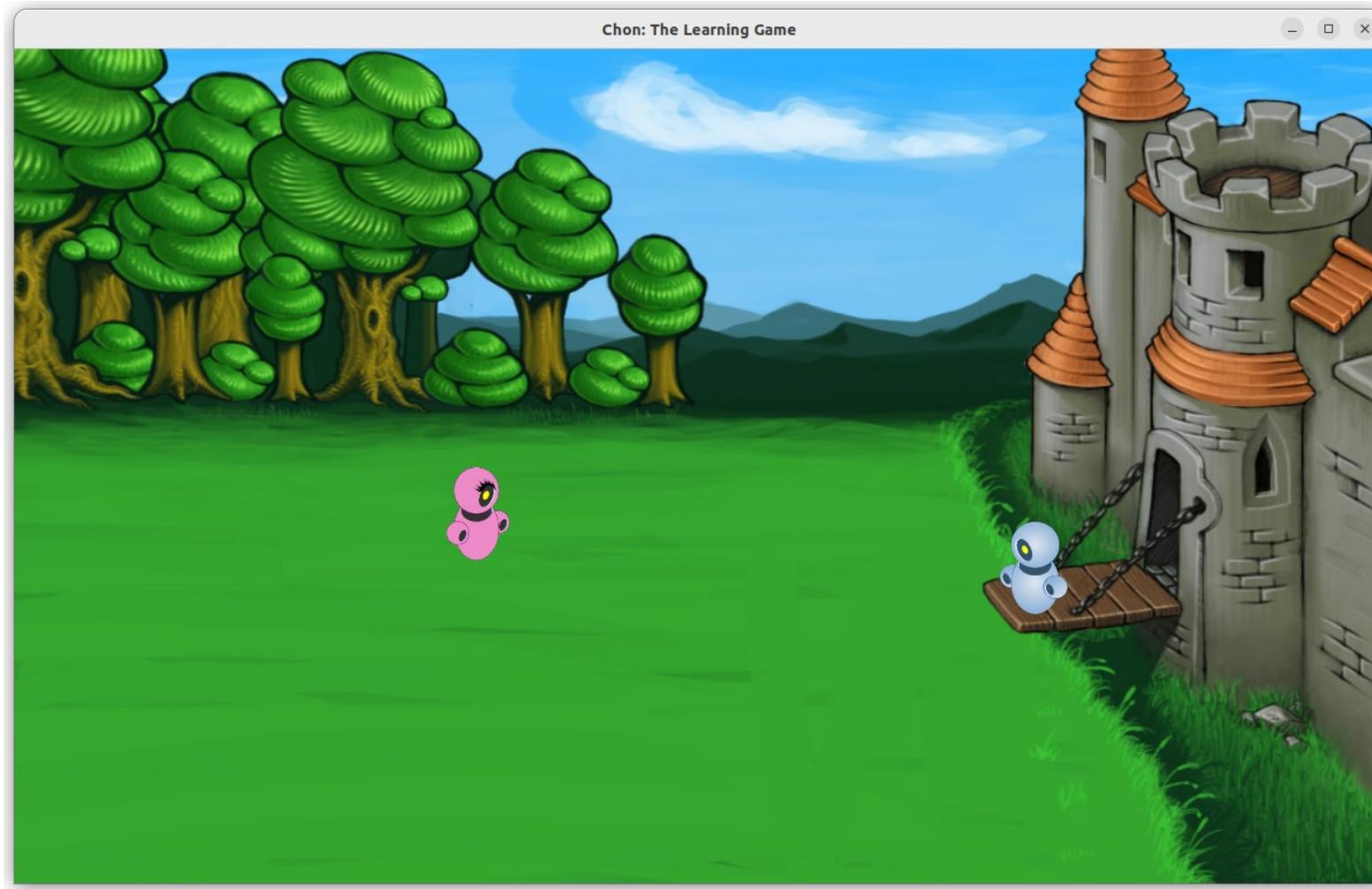
The **AnimationTimer** allows the creation of animations by repeatedly executing a code block at each frame.

# Animation

The **AnimationTimer** allows the creation of animations by repeatedly executing a code block at each frame.

It provides a way to perform updates and render graphics, making it suitable for **creating animations, game loops, and other time-based tasks**.

# Animation



# Animation



○ Pressed: RIGHT  
Released: RIGHT  
□



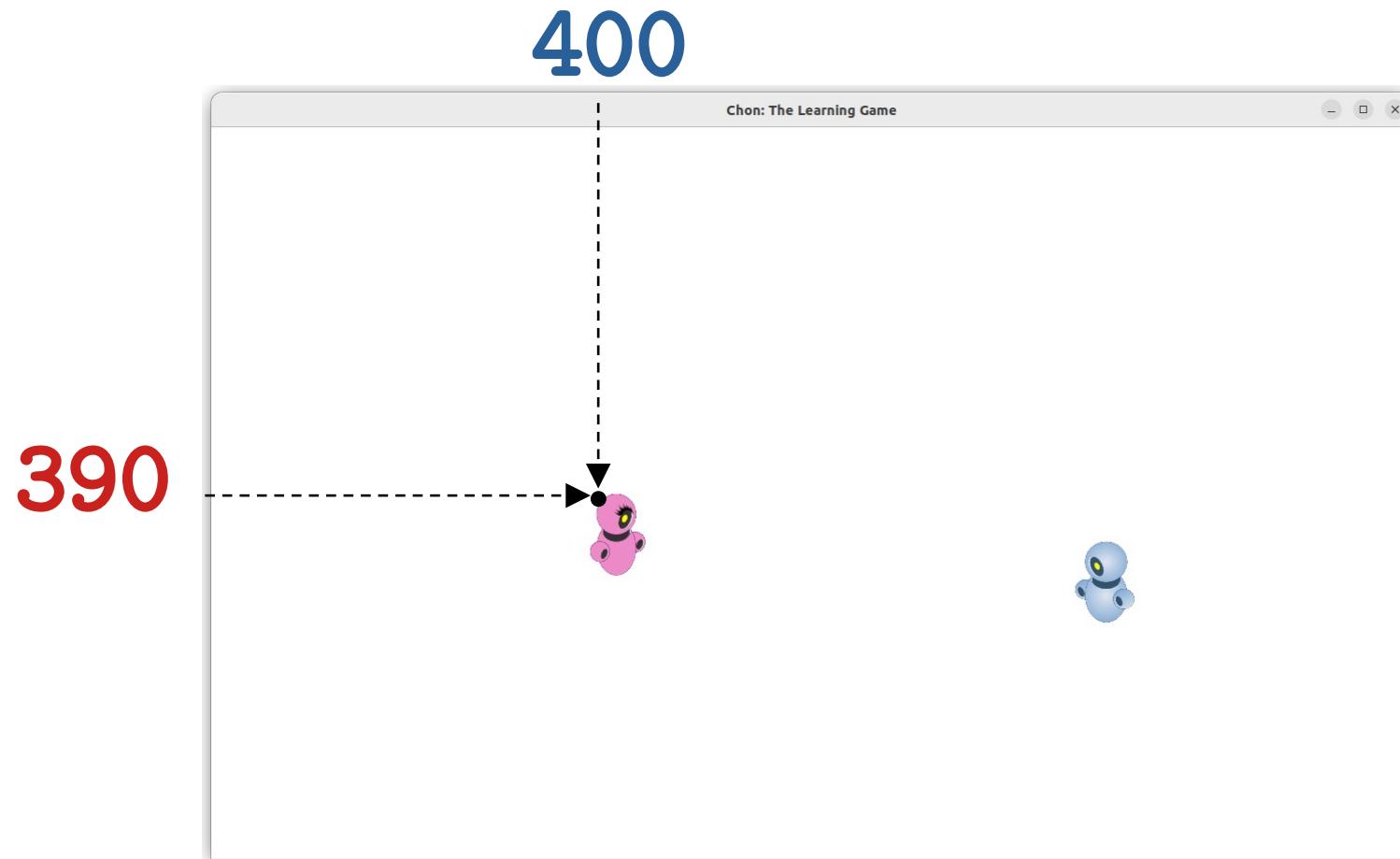
# Animation



○ Pressed: RIGHT  
Released: RIGHT  
□



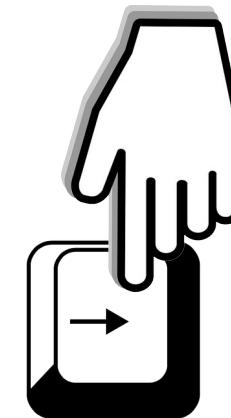
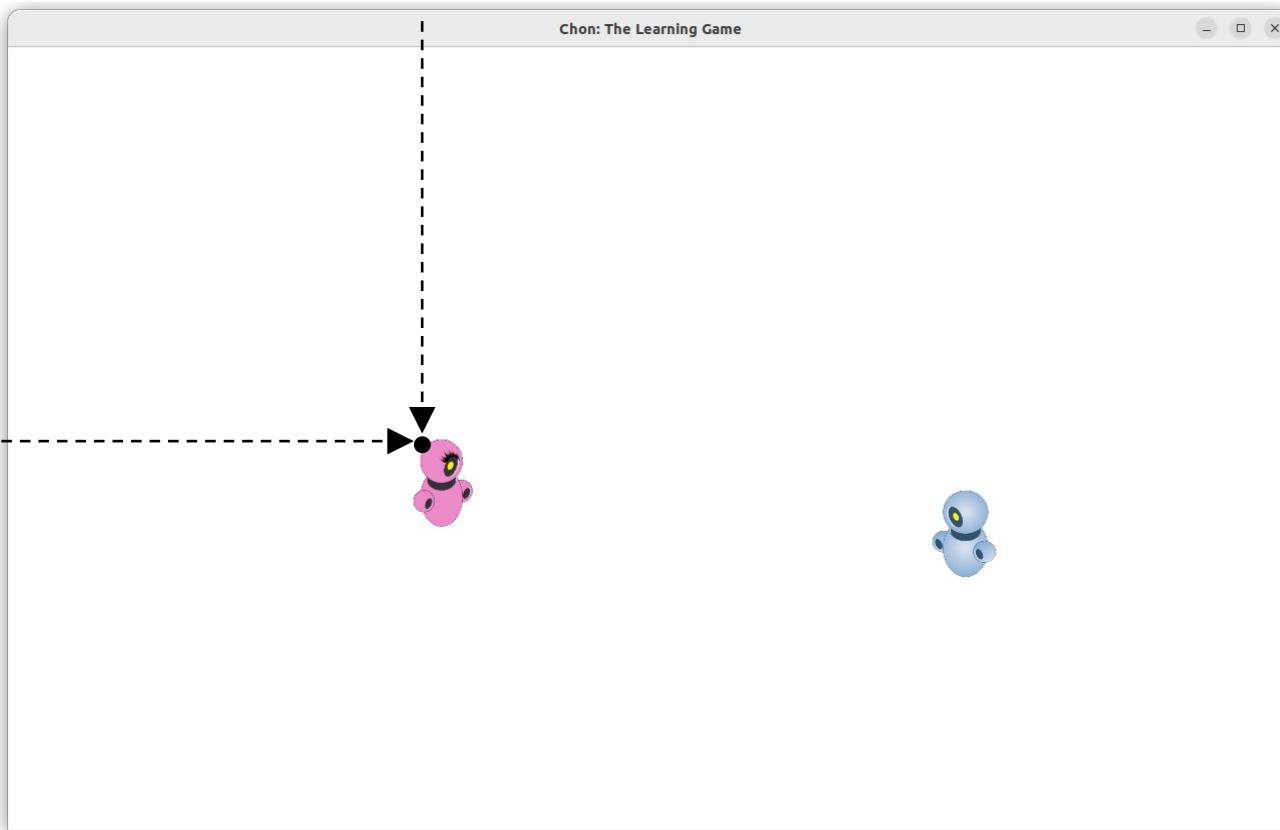
# The RIGHT Logic



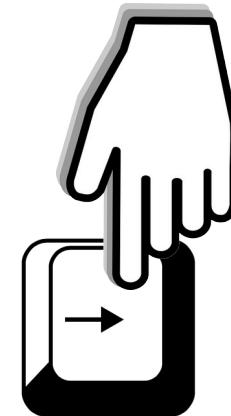
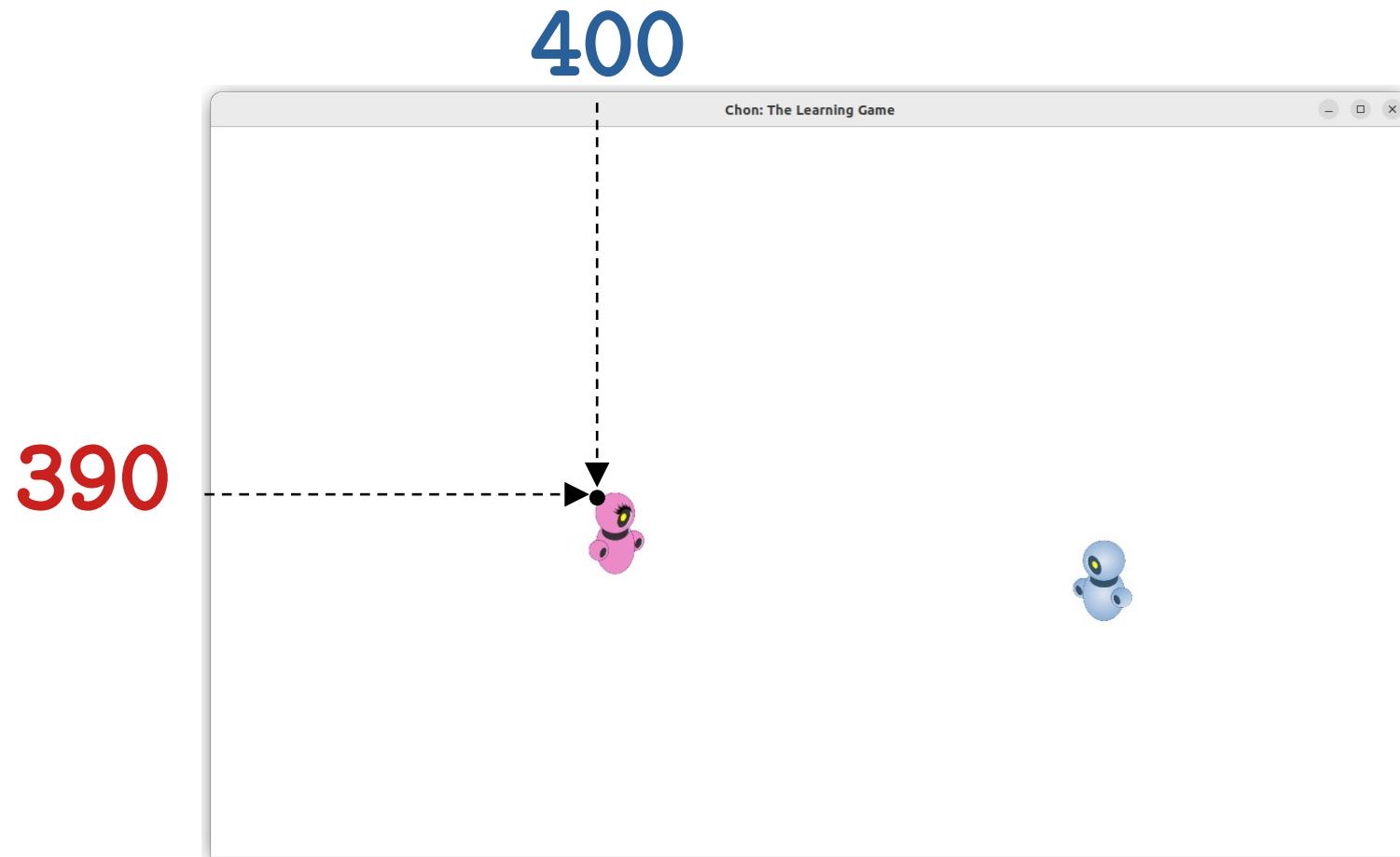
# The RIGHT Logic

400

390



# The RIGHT Logic

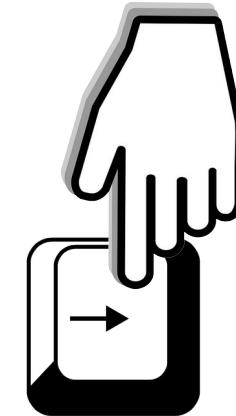
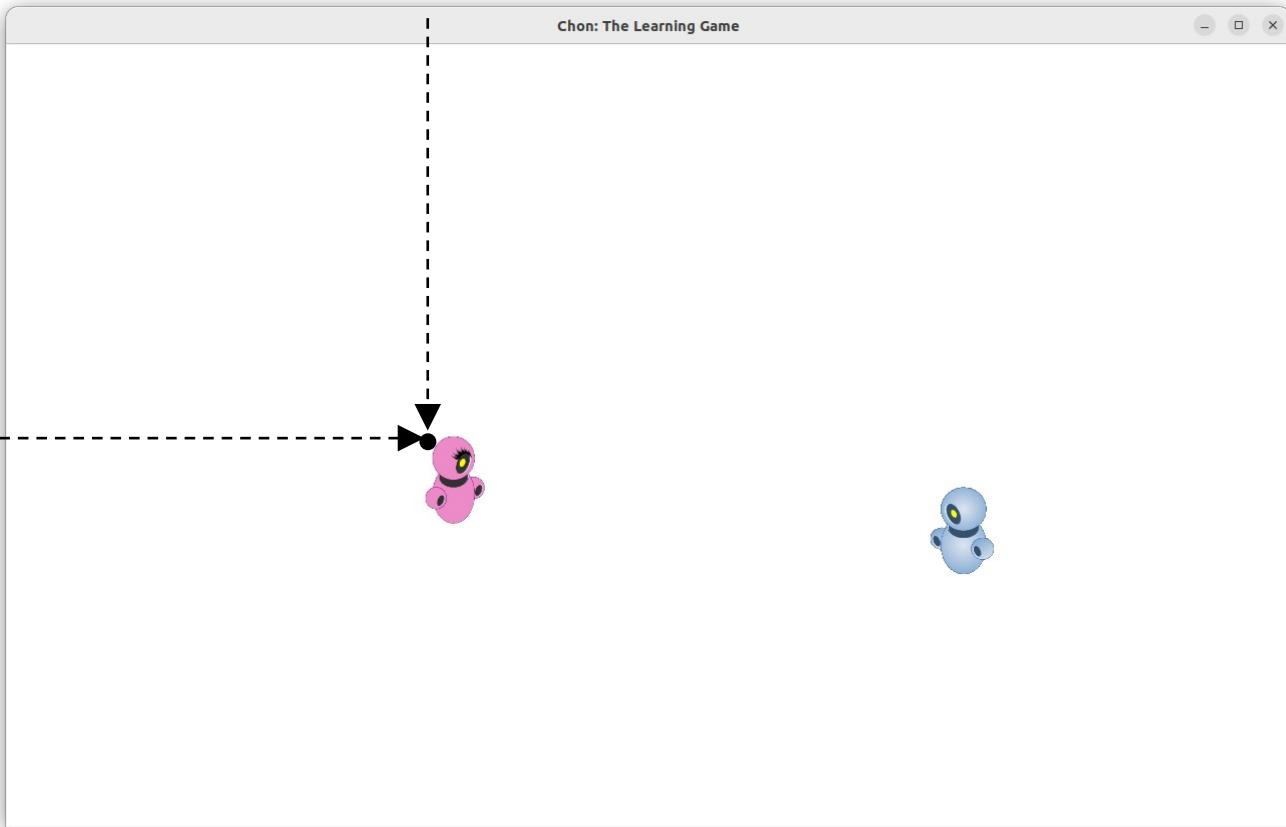


$x + \text{value}$

# The RIGHT Logic

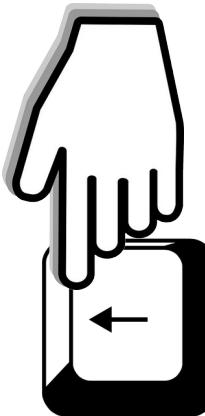
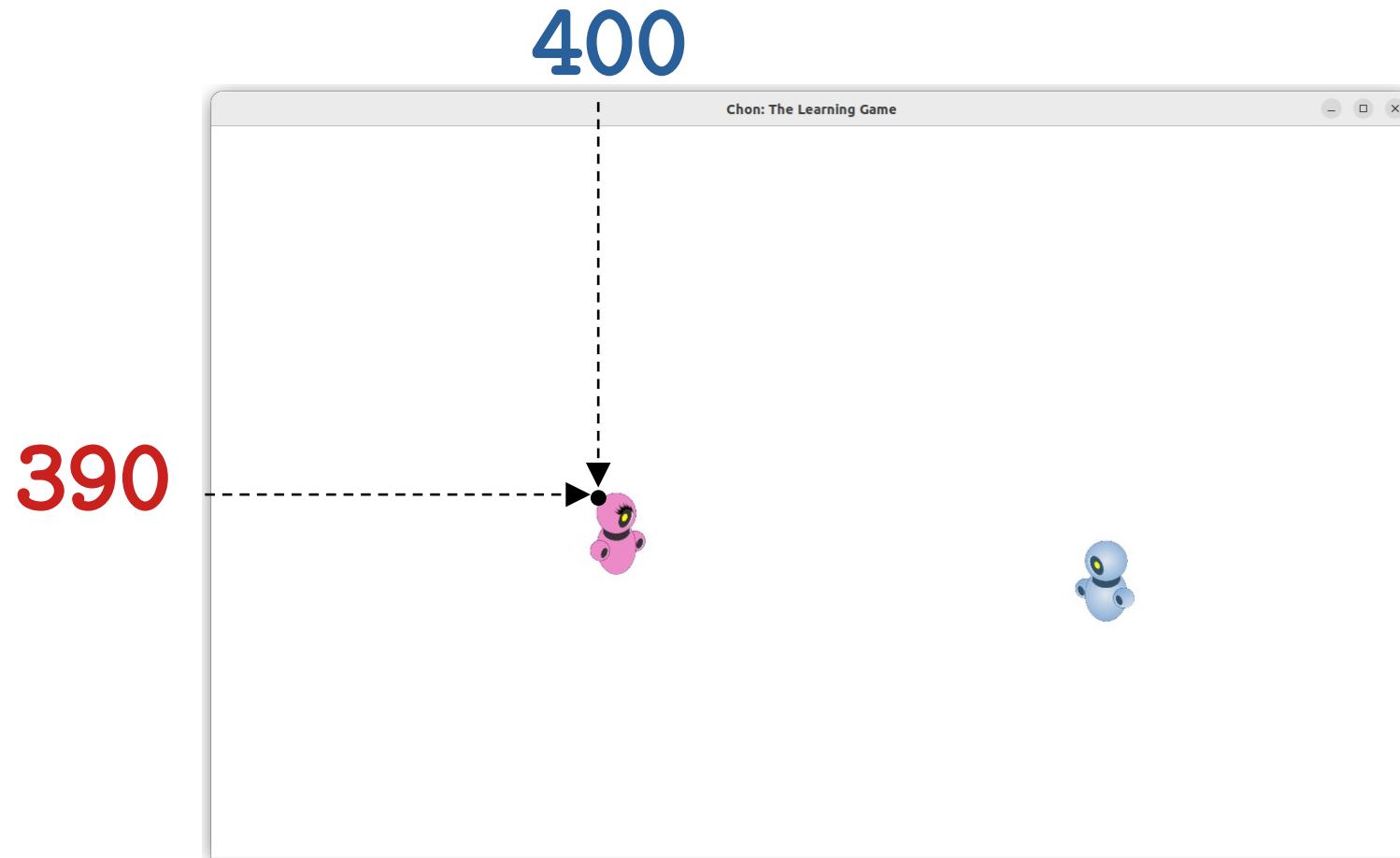
401

390

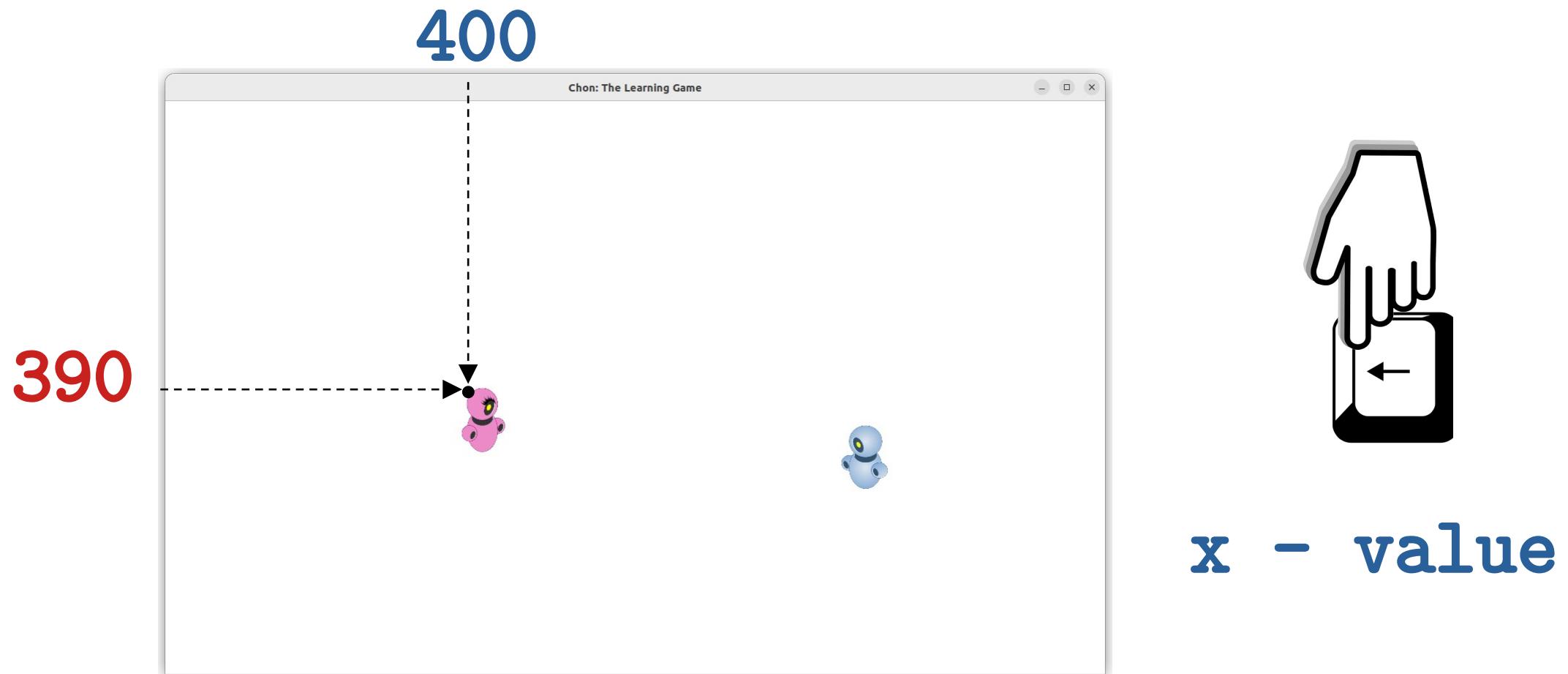


$x + \text{value}$

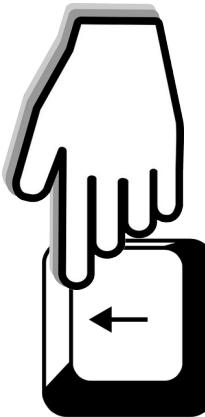
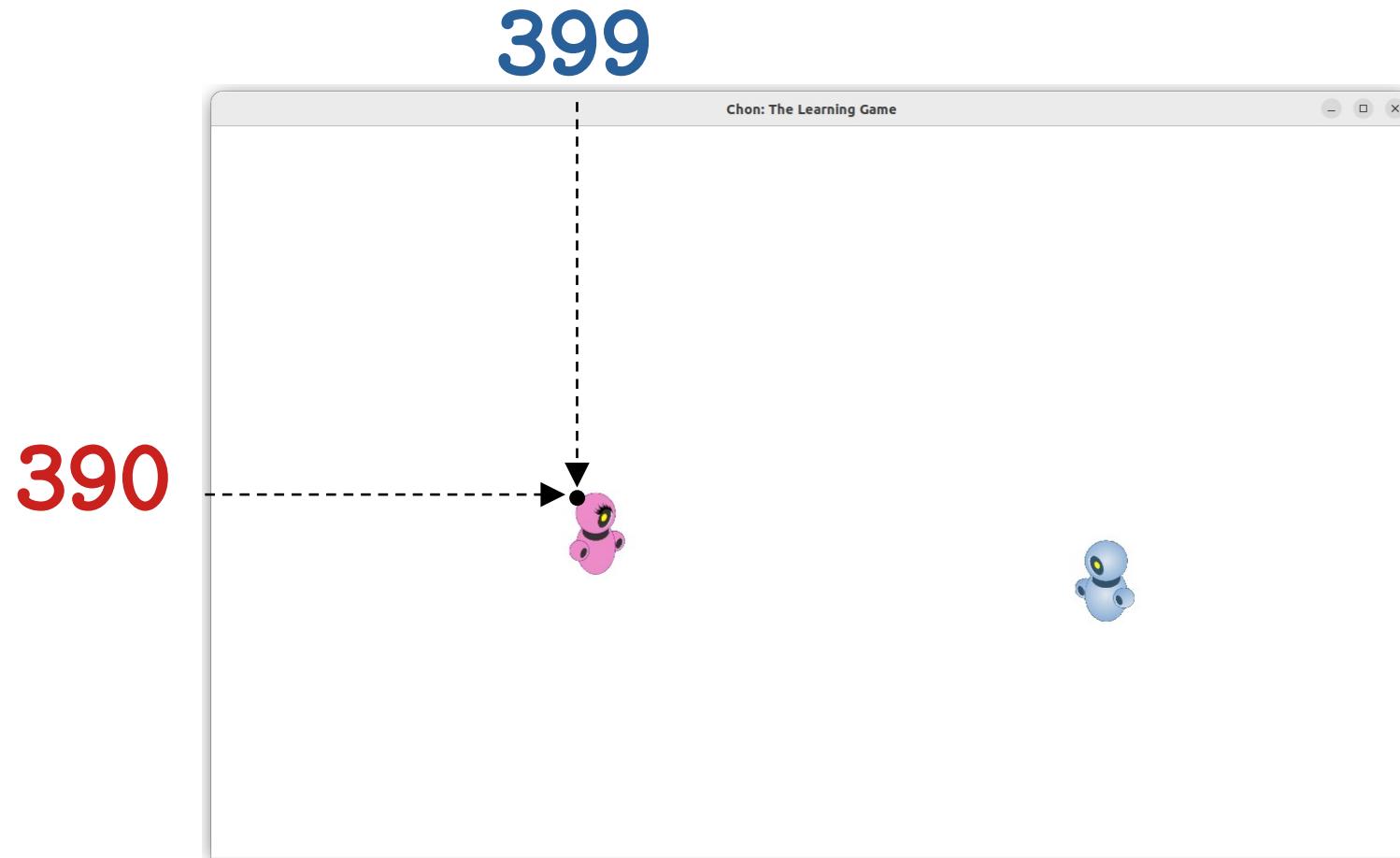
# The LEFT Logic



# The LEFT Logic

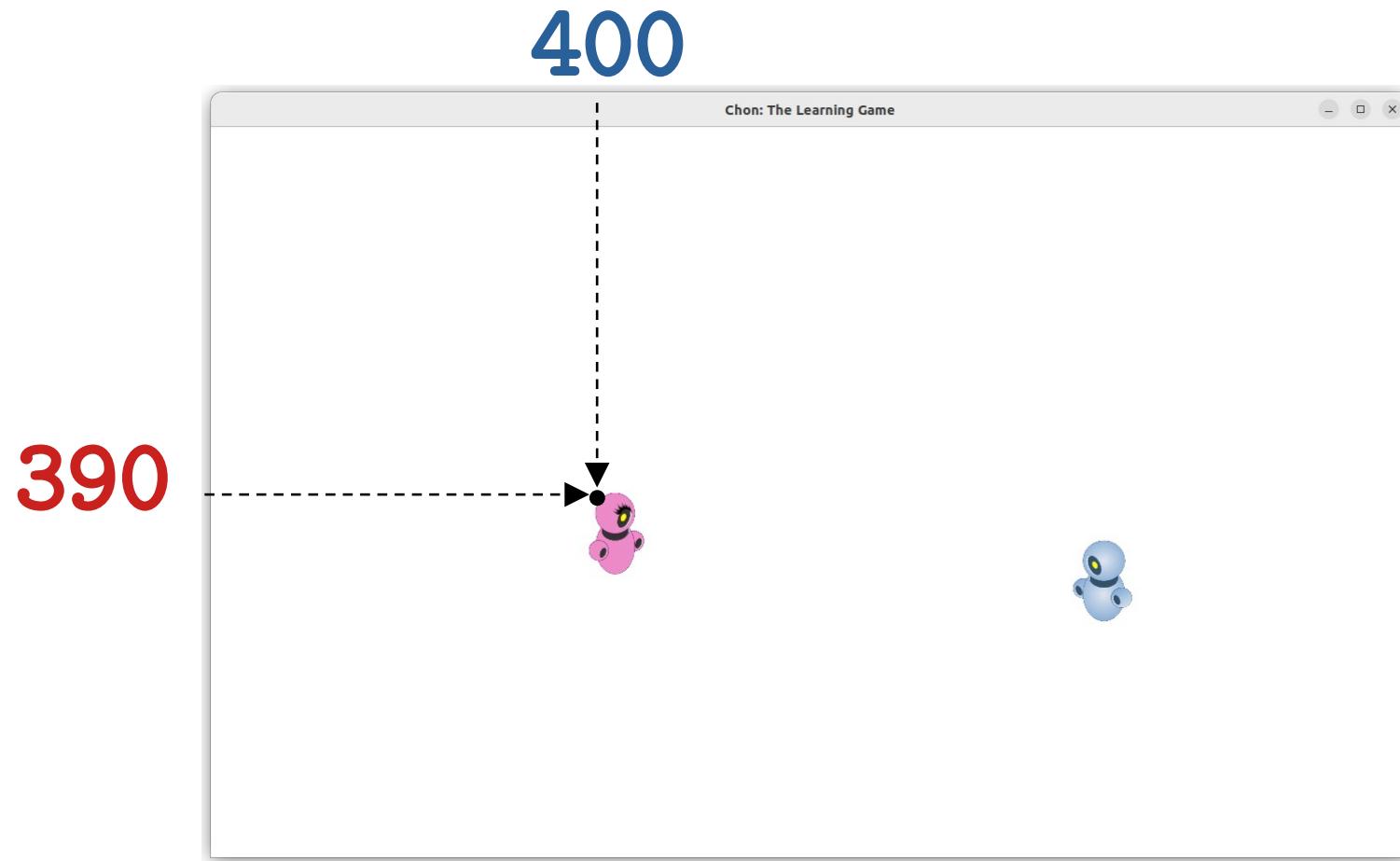


# The LEFT Logic

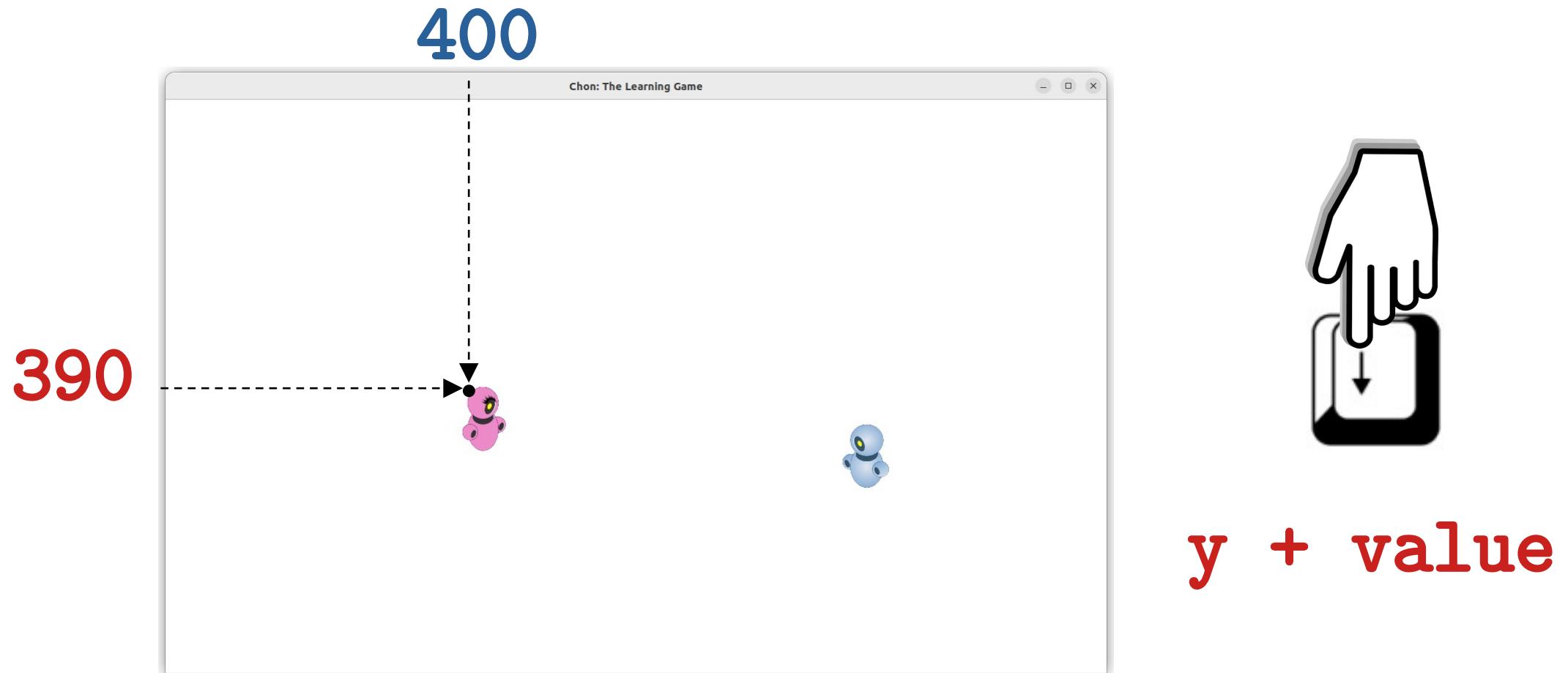


**x - value**

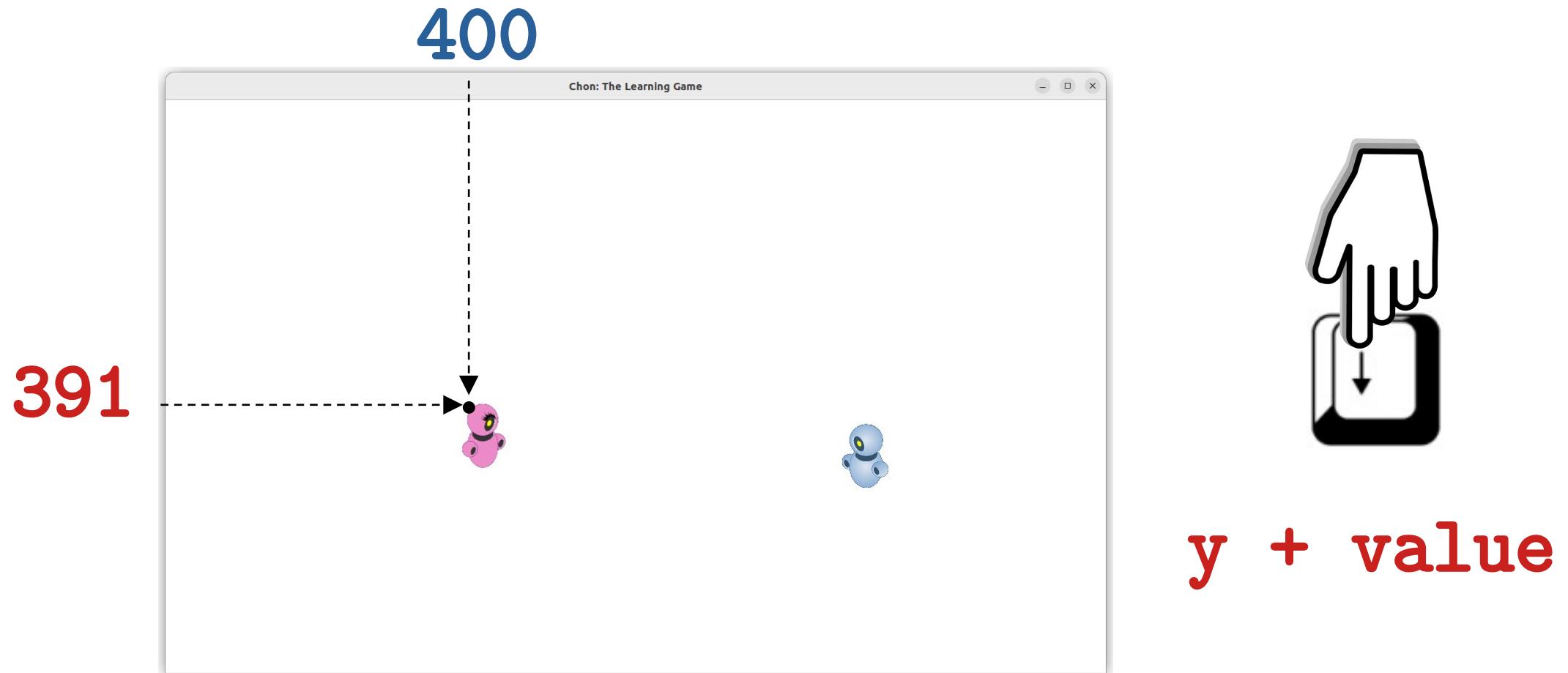
# The DOWN Logic



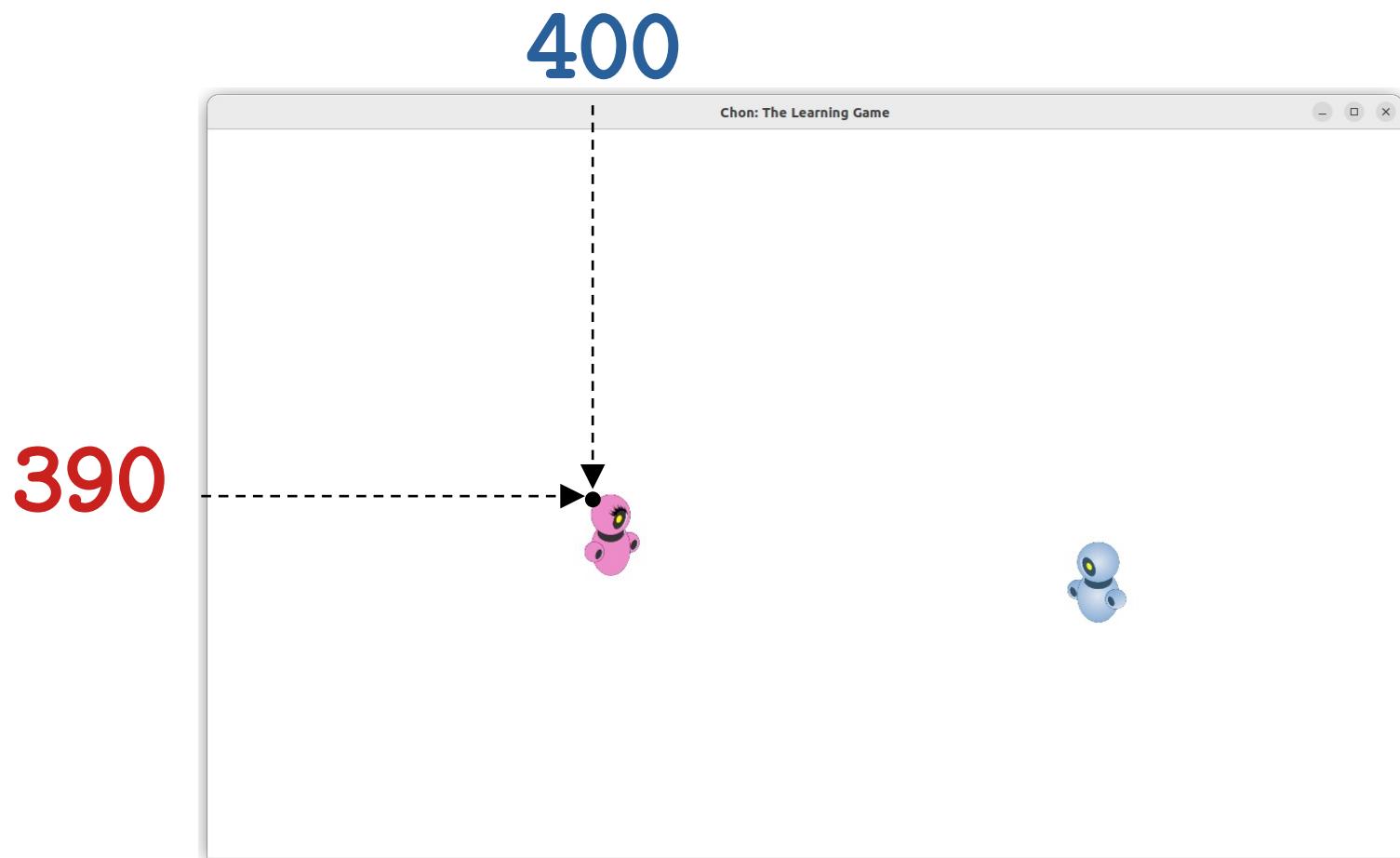
# The DOWN Logic



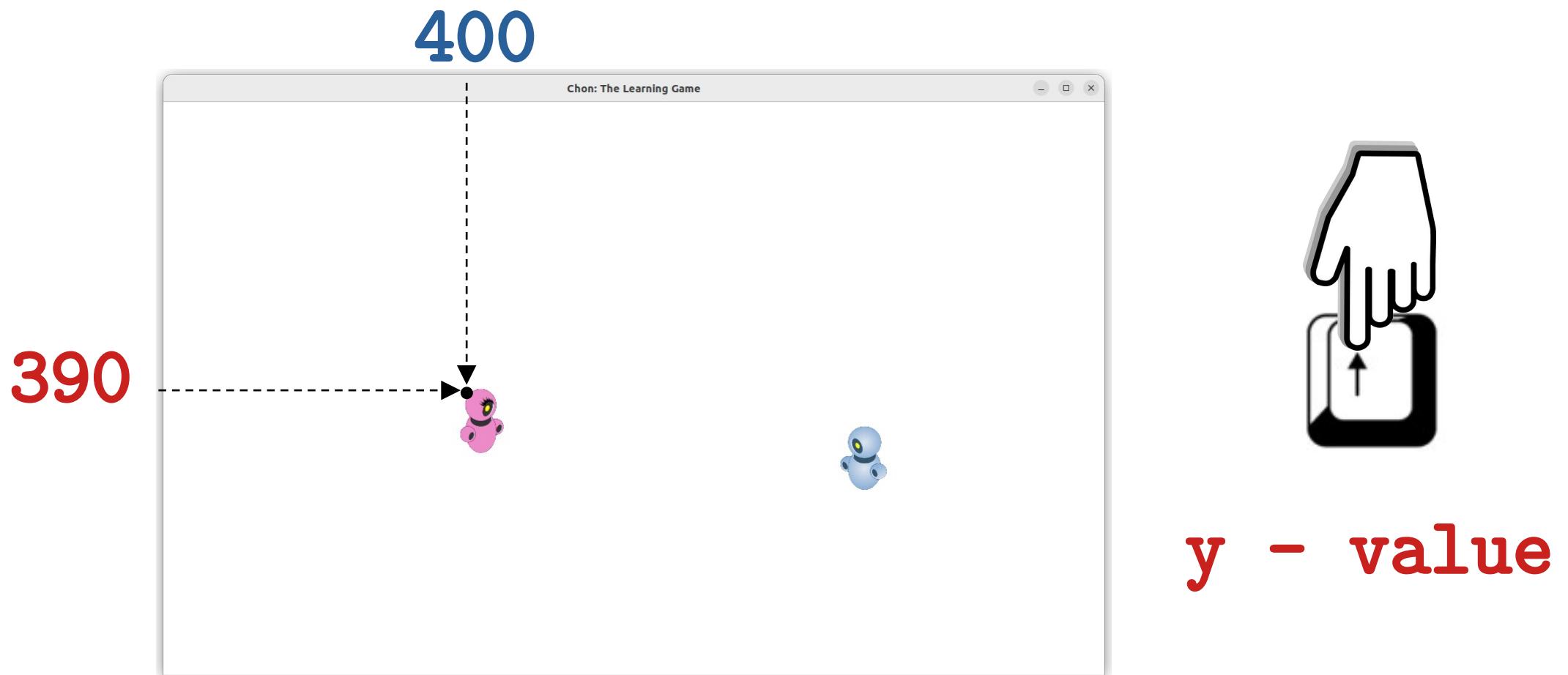
# The DOWN Logic



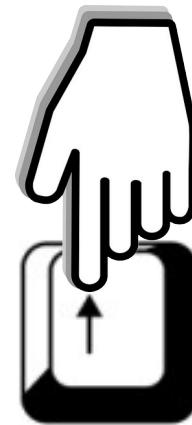
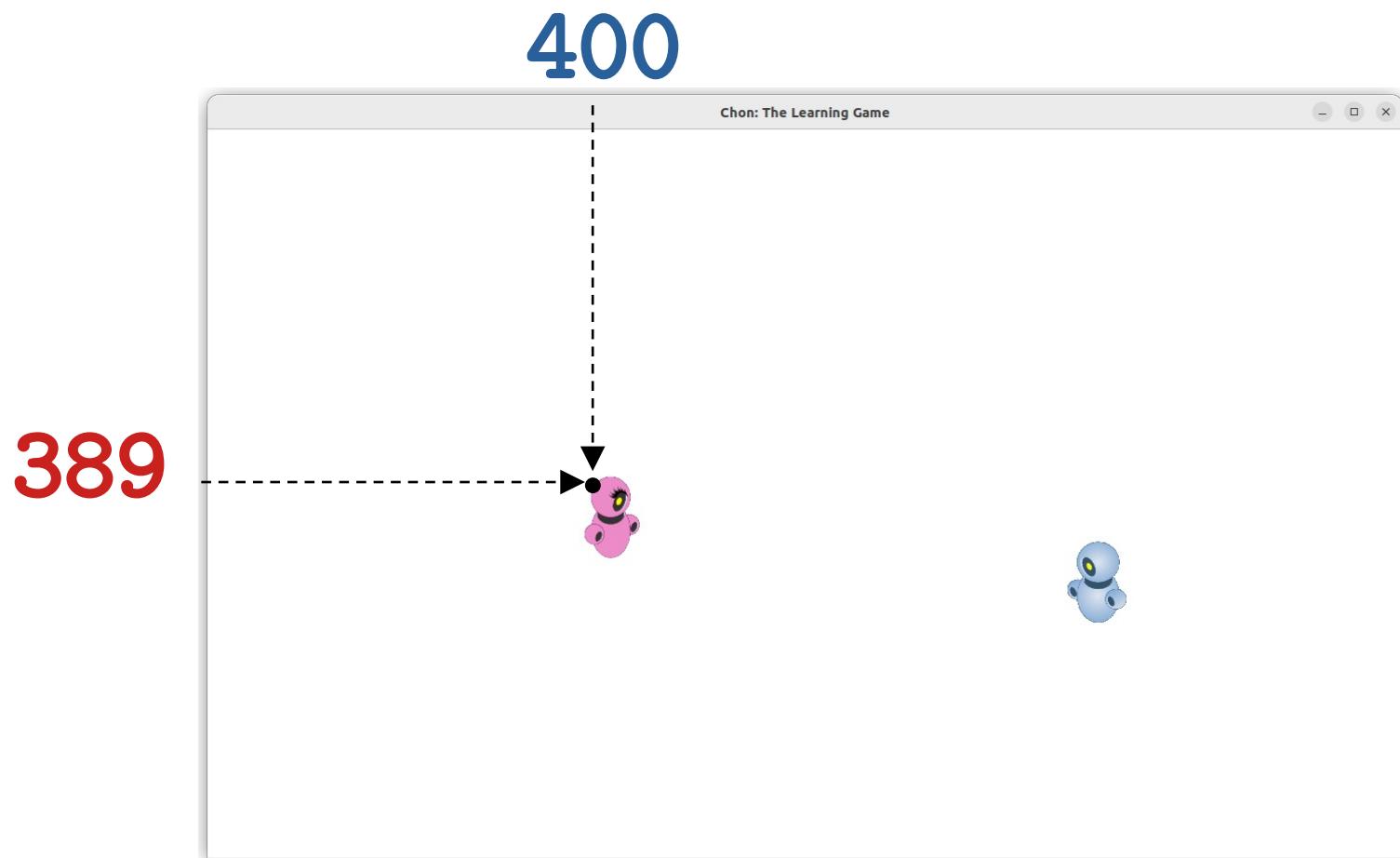
# The UP Logic



# The UP Logic



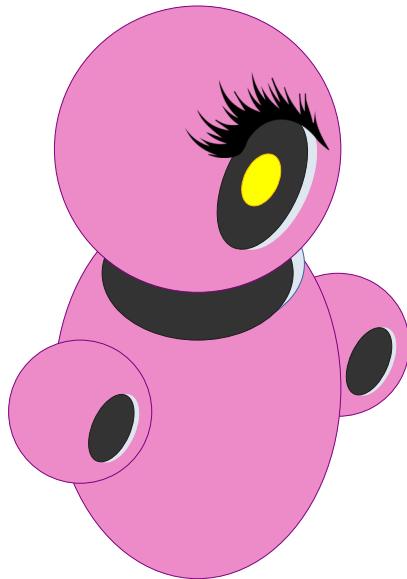
# The UP Logic



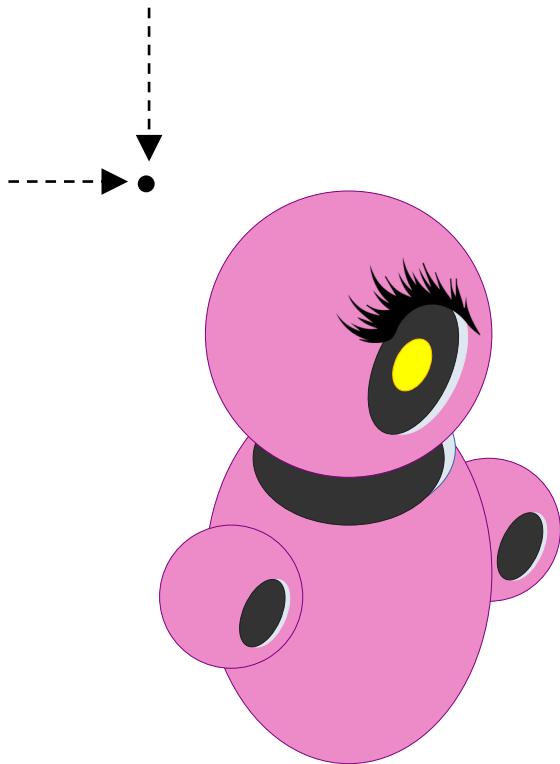
y - value

# Defining Boundaries

Every Image has:



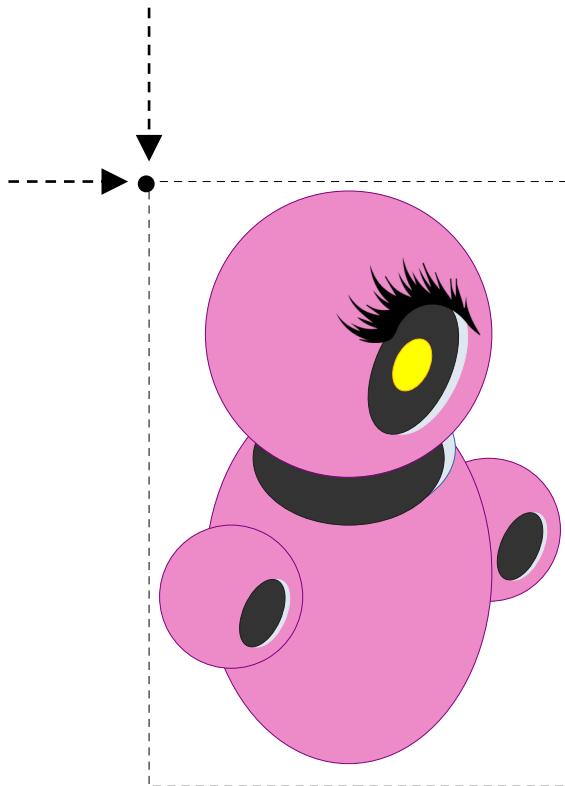
# Defining Boundaries



Every Image has:

- x and y points;

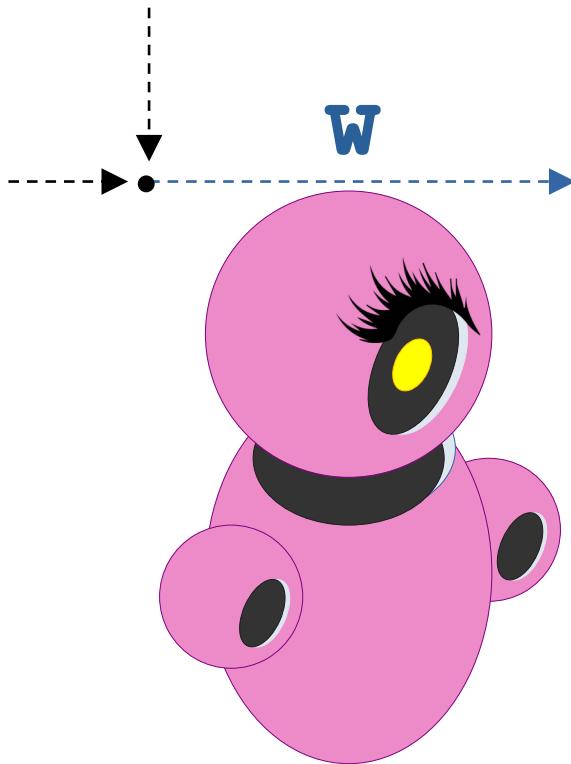
# Defining Boundaries



Every Image has:

- x and y points;

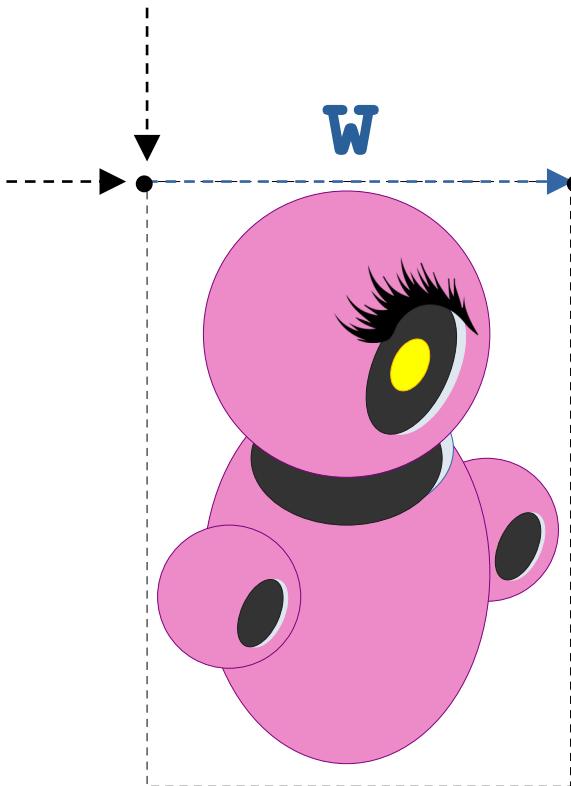
# Defining Boundaries



Every Image has:

- x and y points;
- Width;

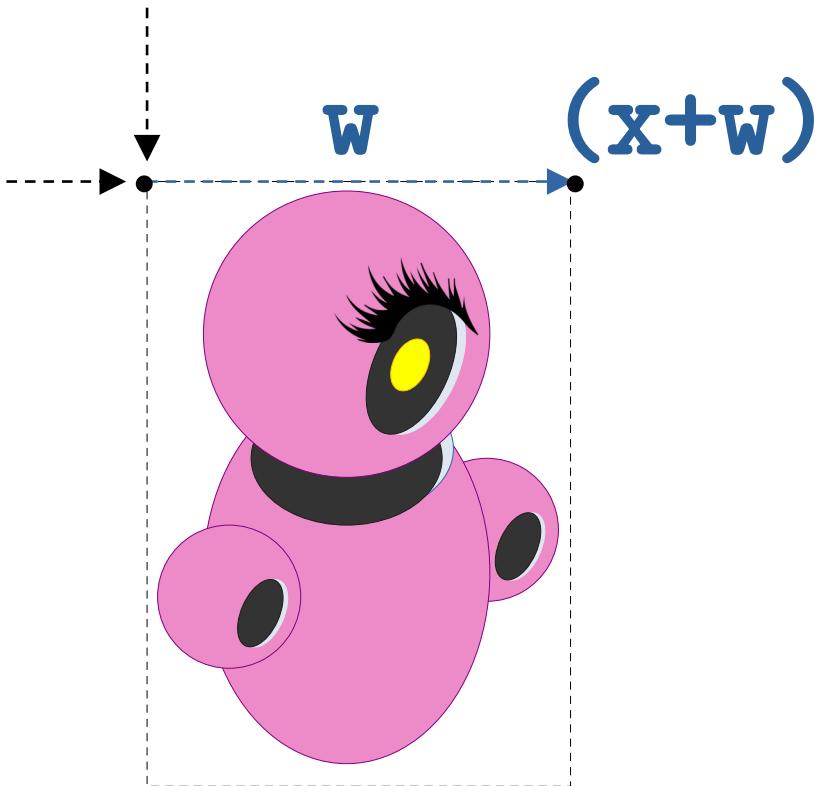
# Defining Boundaries



Every Image has:

- x and y points;
- Width;

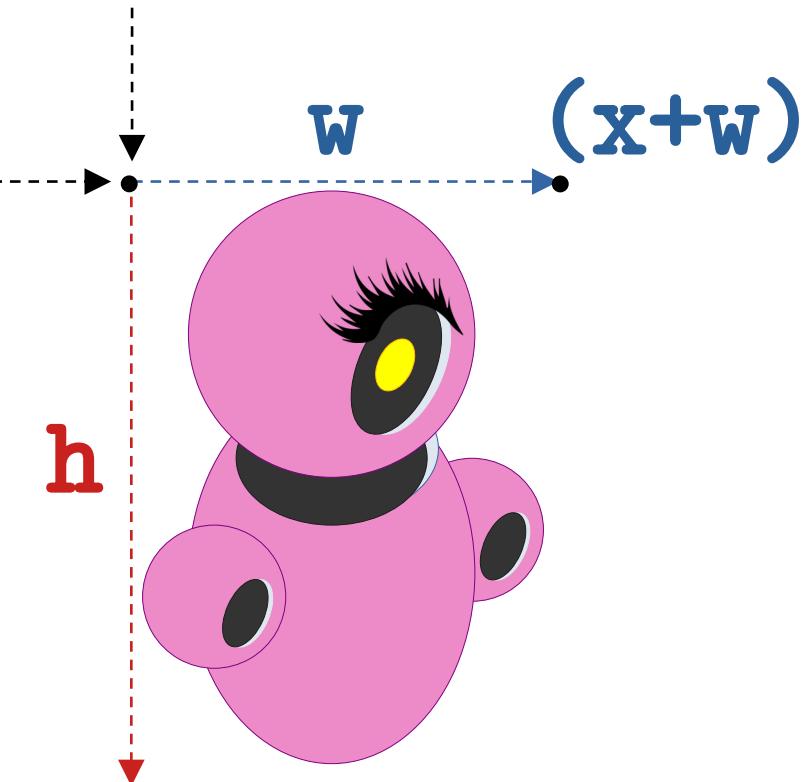
# Defining Boundaries



Every Image has:

- x and y points;
- Width;

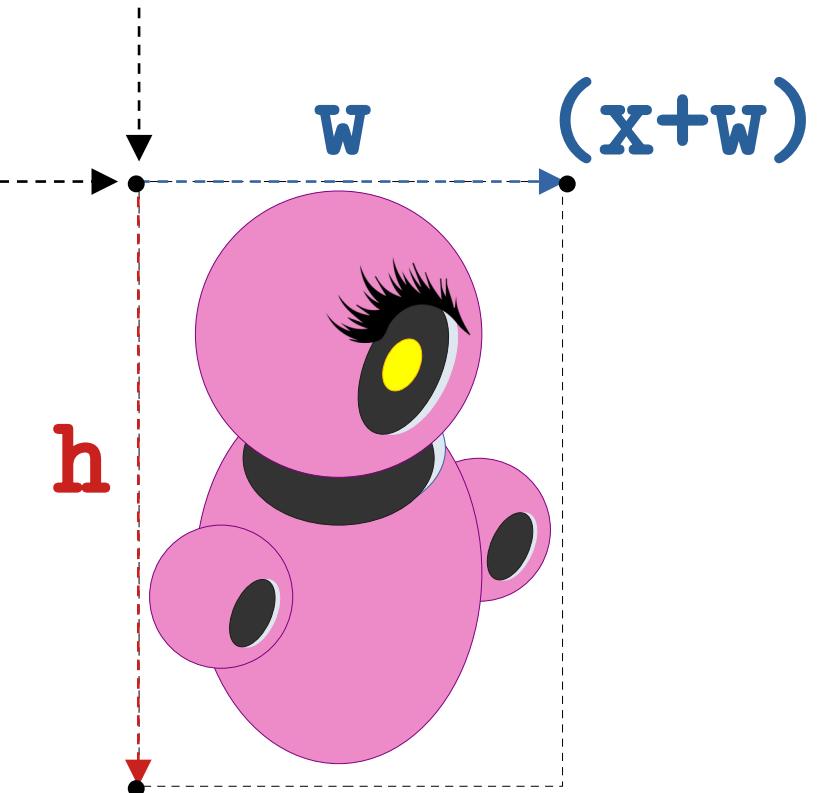
# Defining Boundaries



Every Image has:

- x and y points;
  - Width;
  - Height.

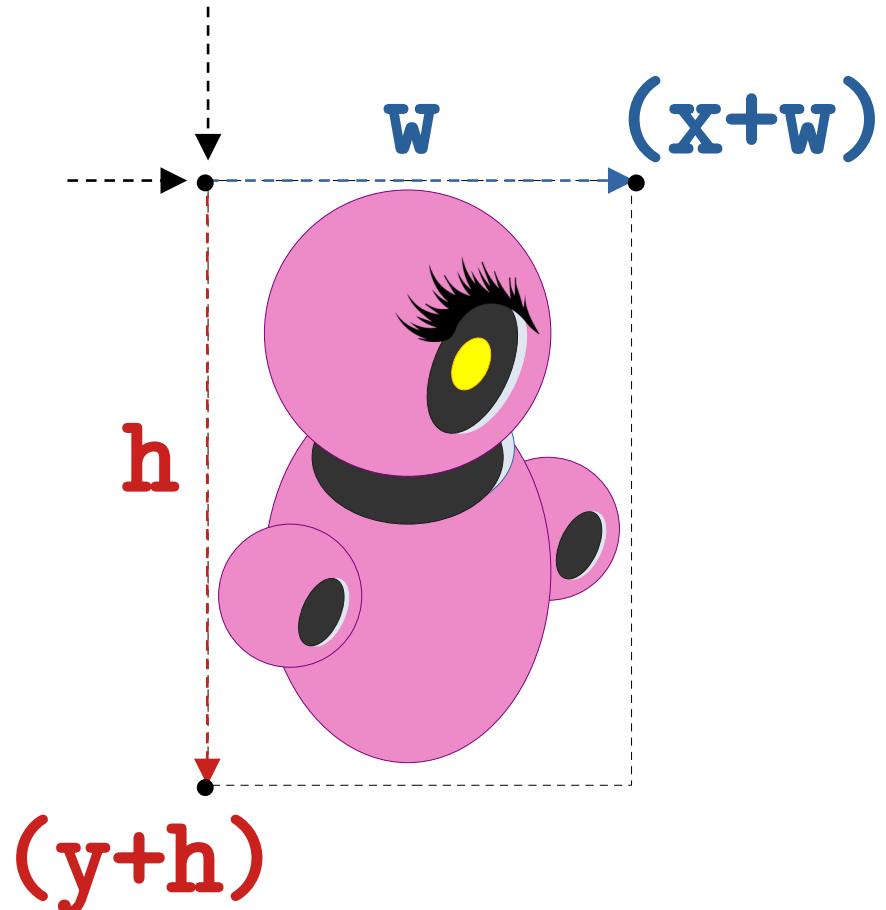
# Defining Boundaries



Every Image has:

- x and y points;
  - Width;
  - Height.

# Defining Boundaries



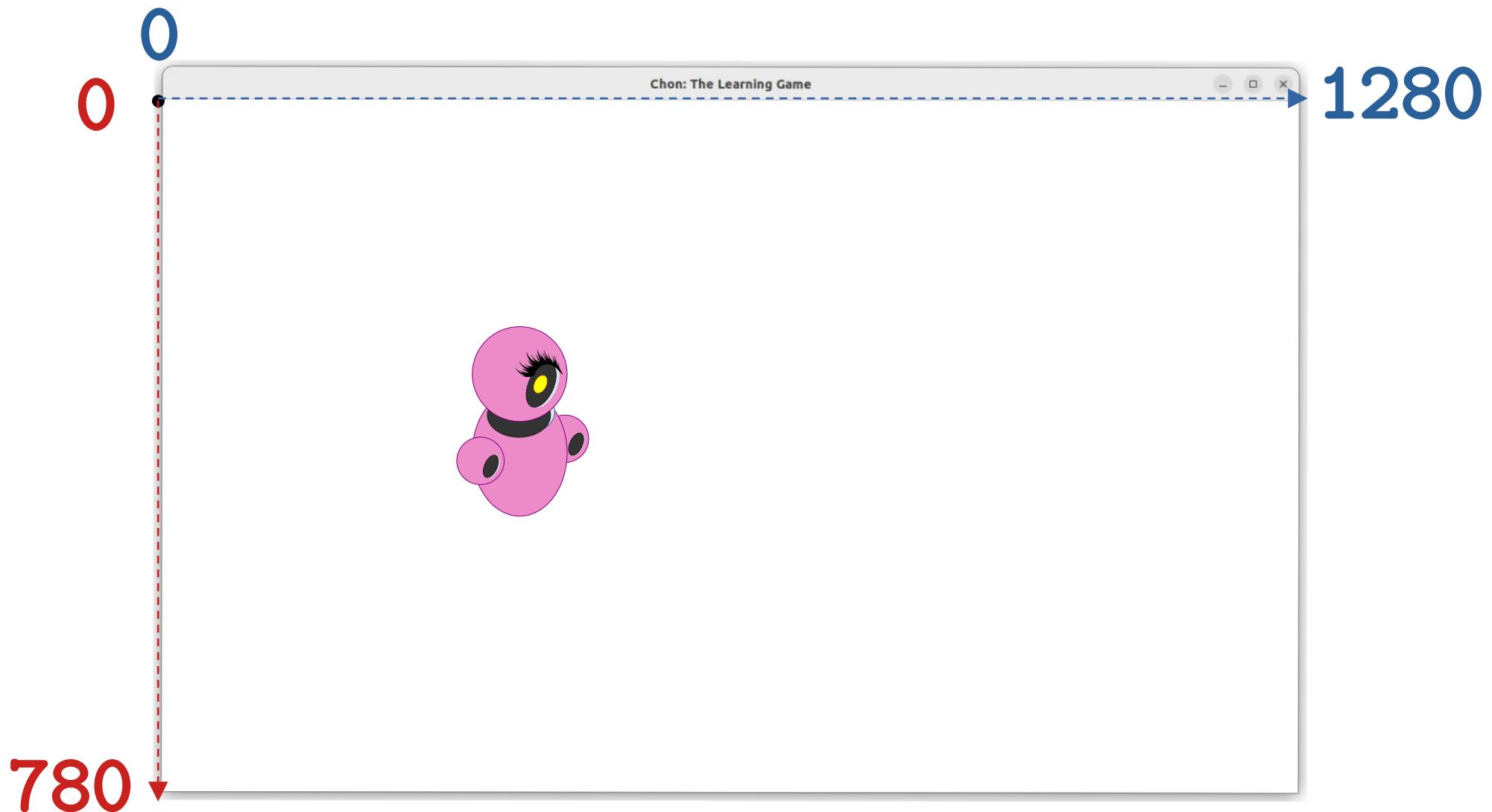
Every Image has:

- x and y points;
  - Width;
  - Height.

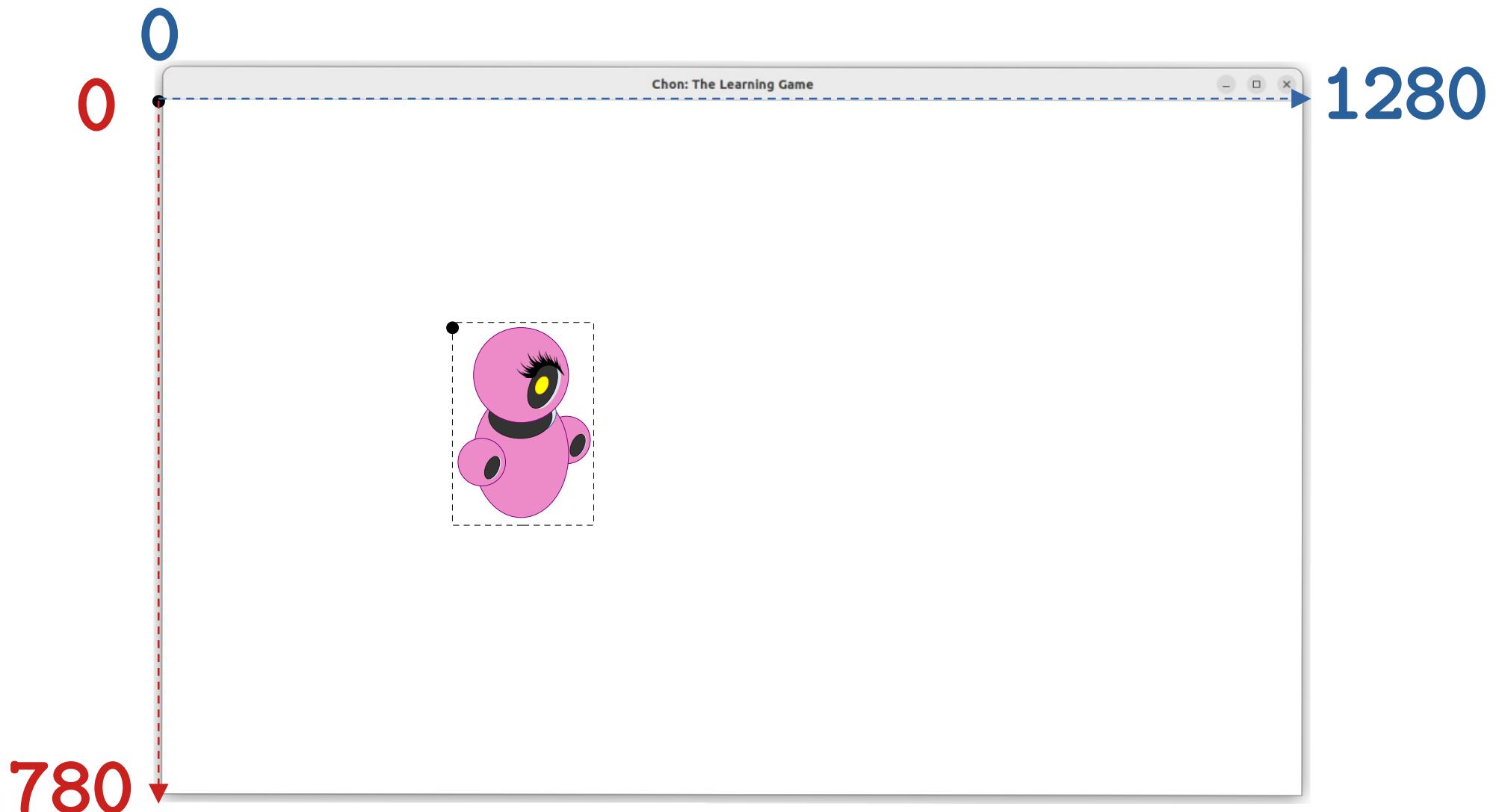
# Defining Boundaries



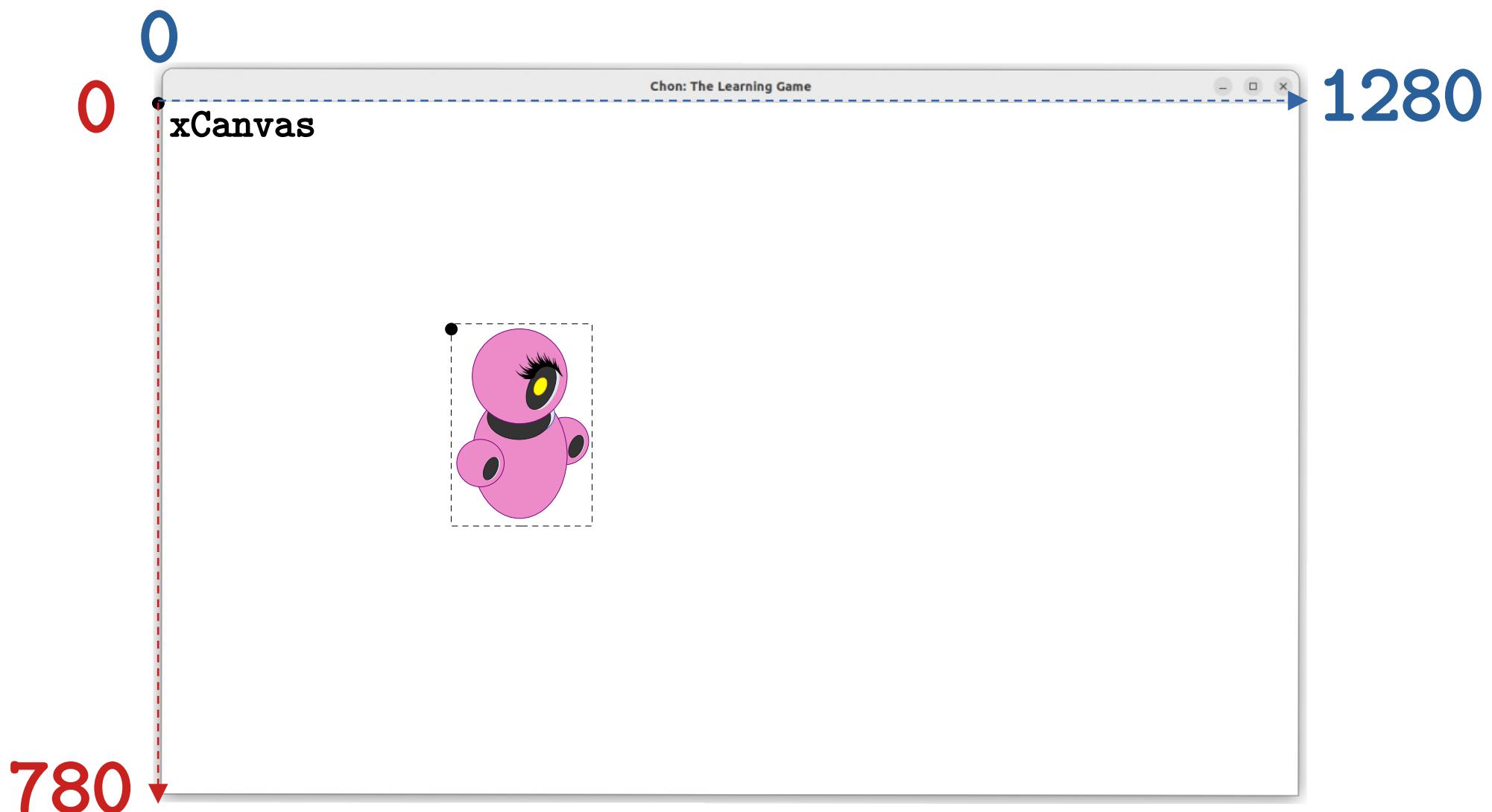
# Defining Boundaries



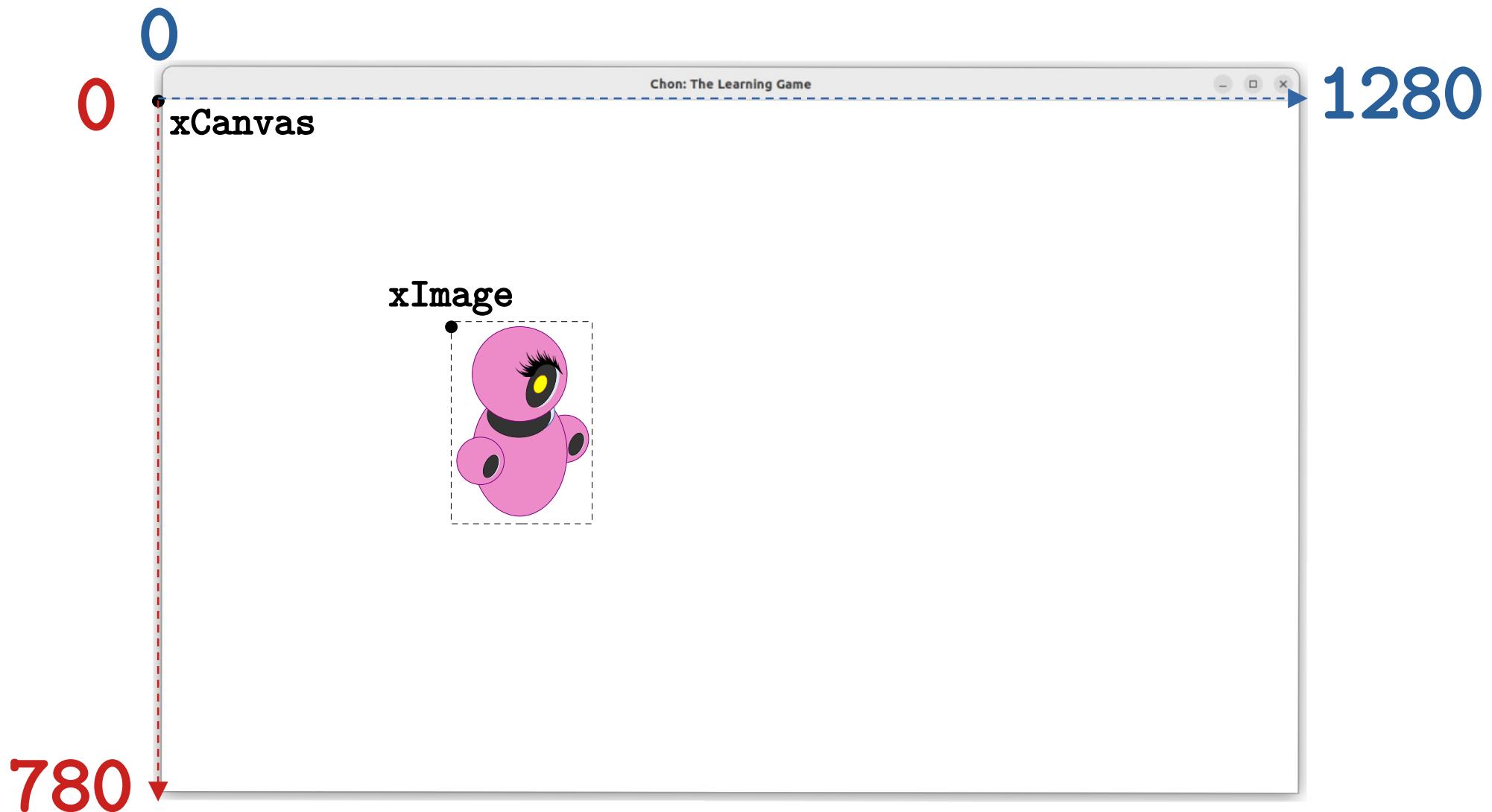
# Defining Boundaries at the LEFT



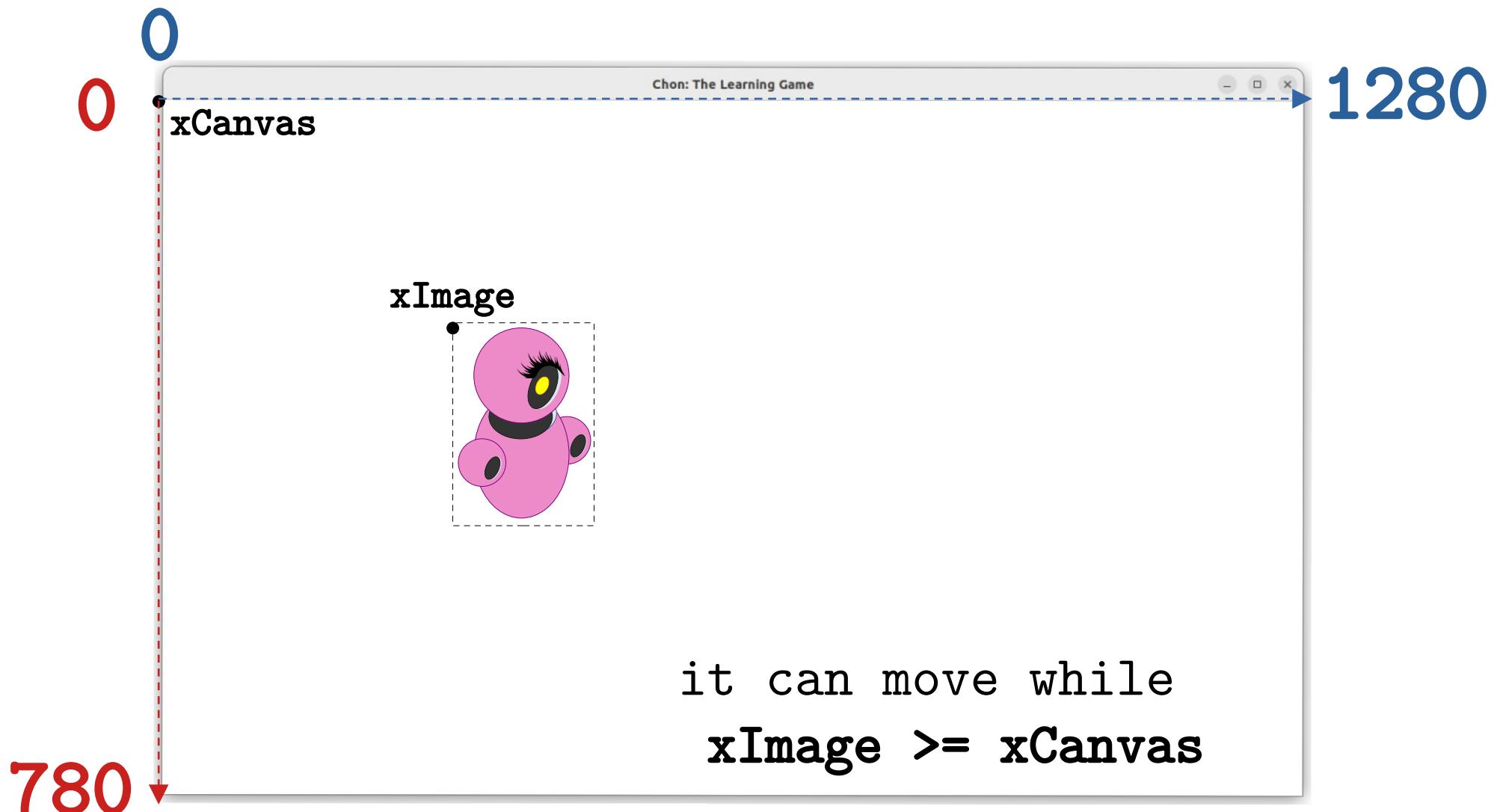
# Defining Boundaries at the LEFT



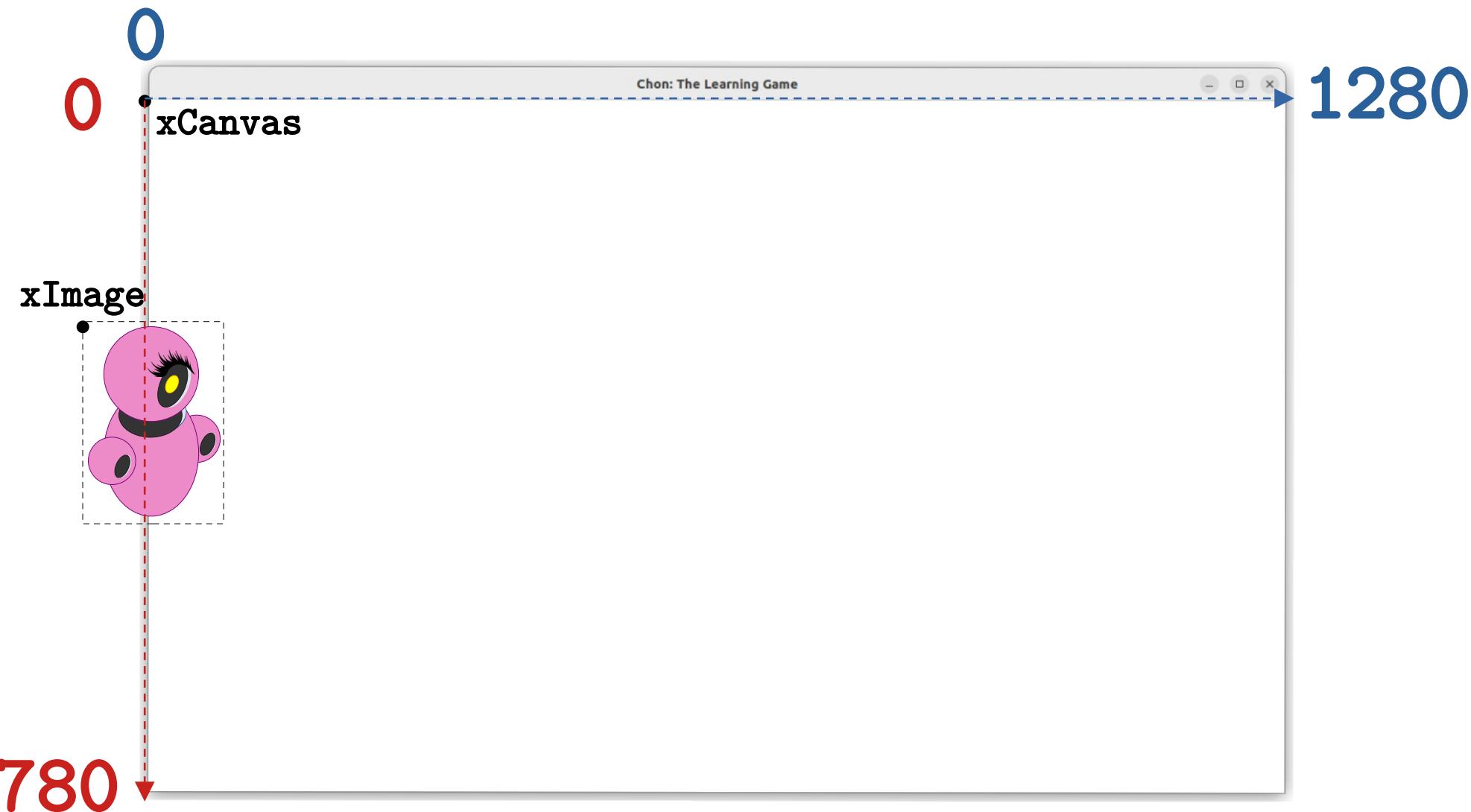
# Defining Boundaries at the LEFT



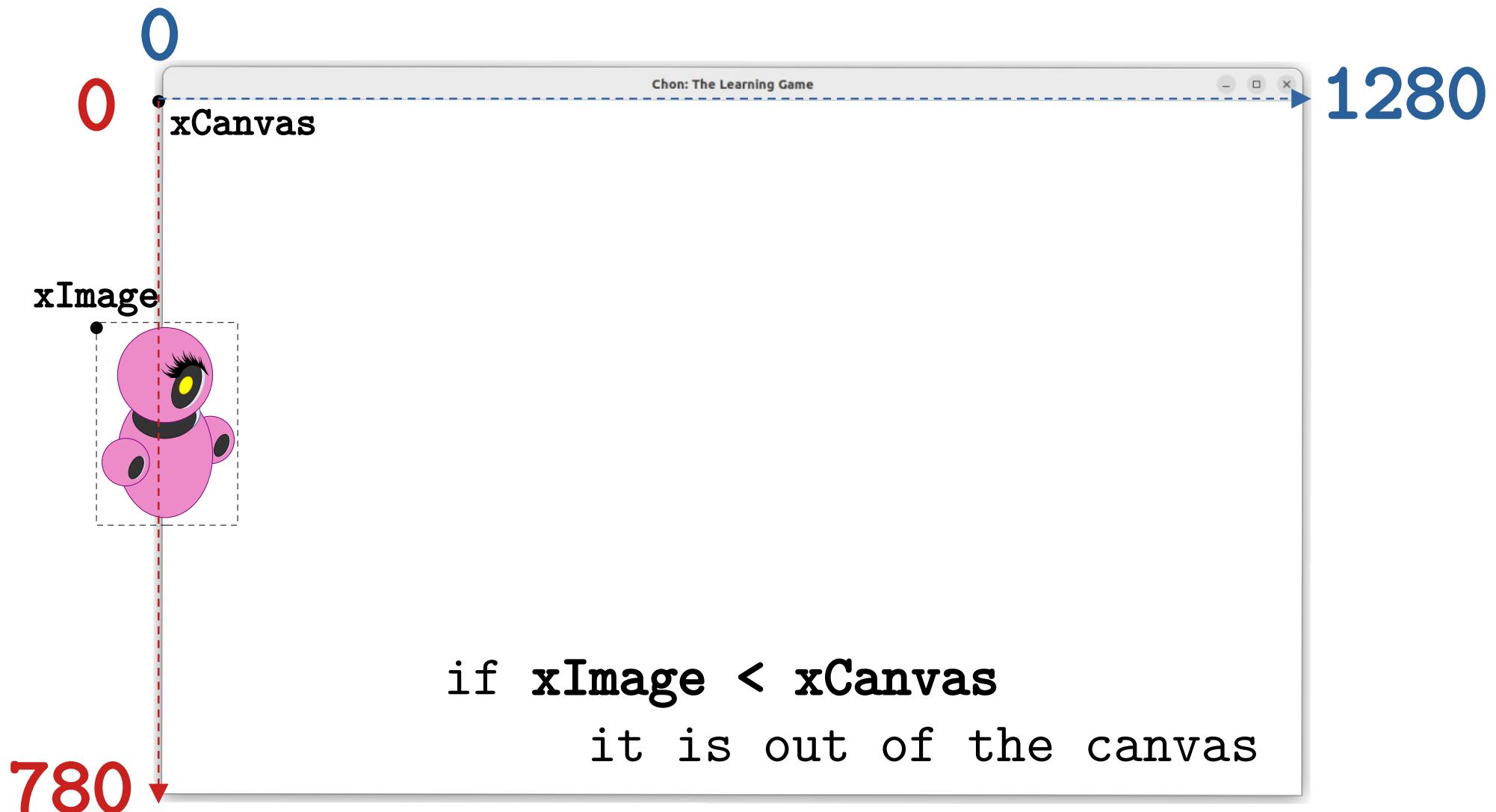
# Defining Boundaries at the LEFT



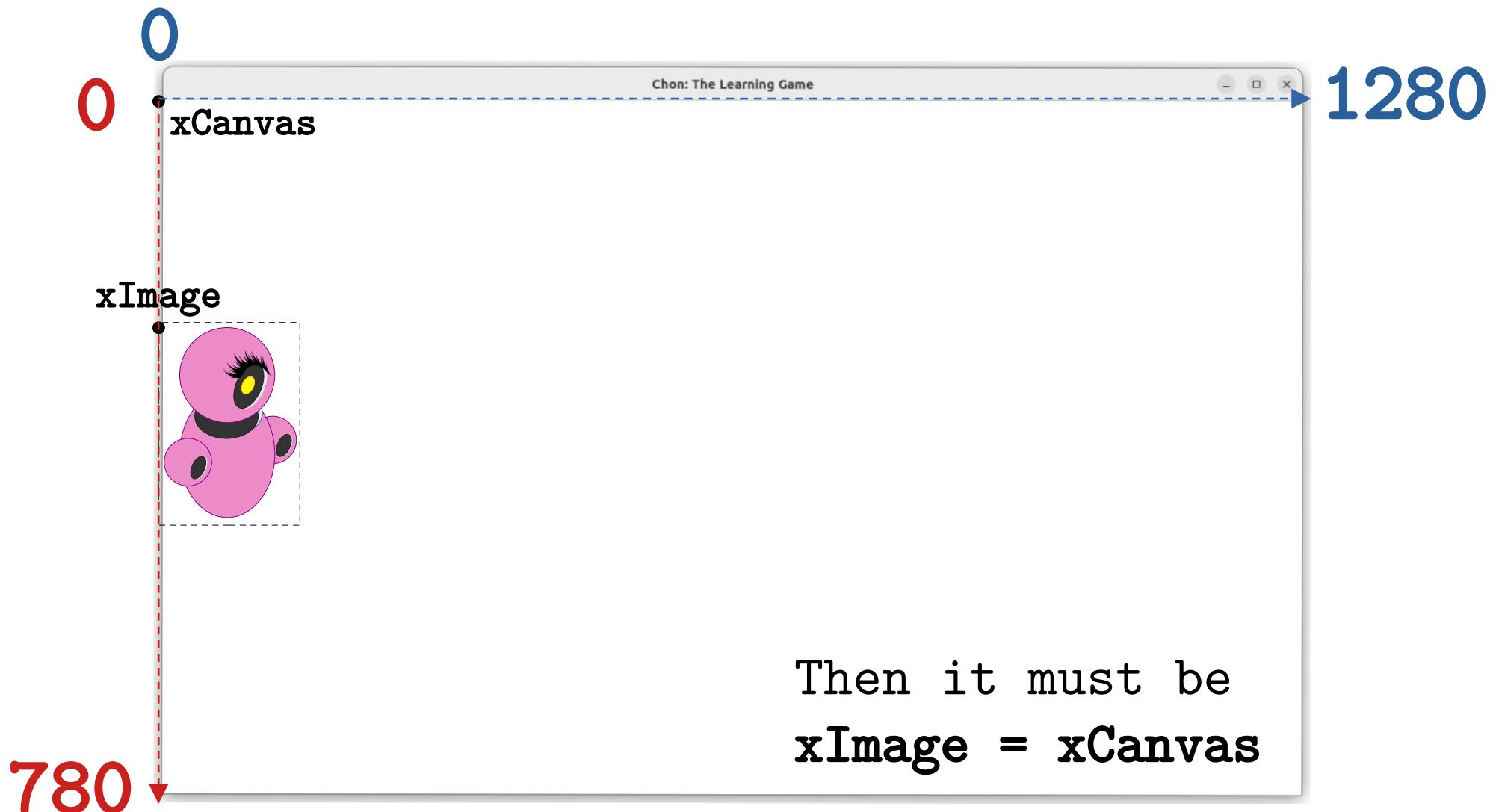
# Defining Boundaries at the LEFT



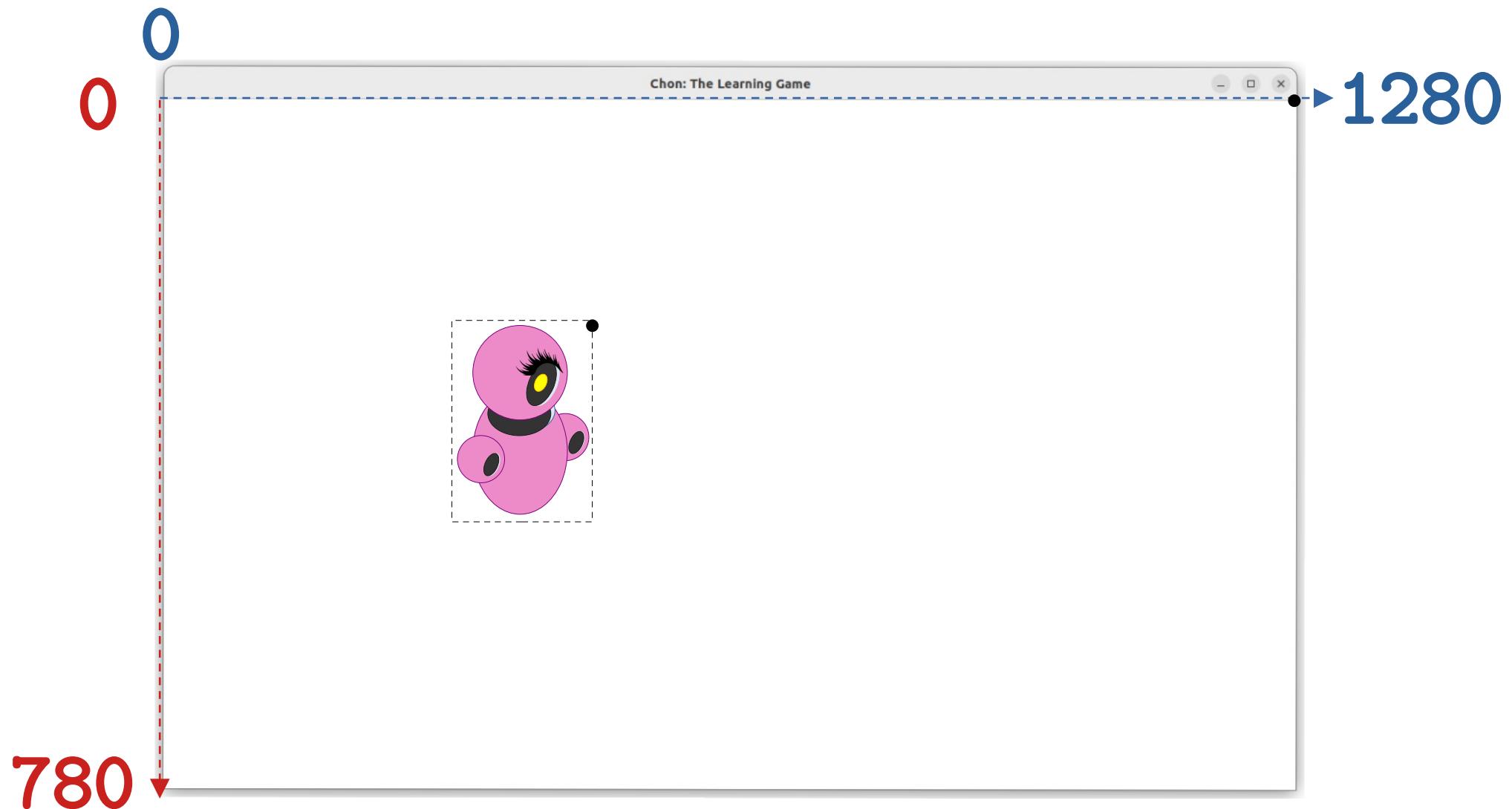
# Defining Boundaries at the LEFT



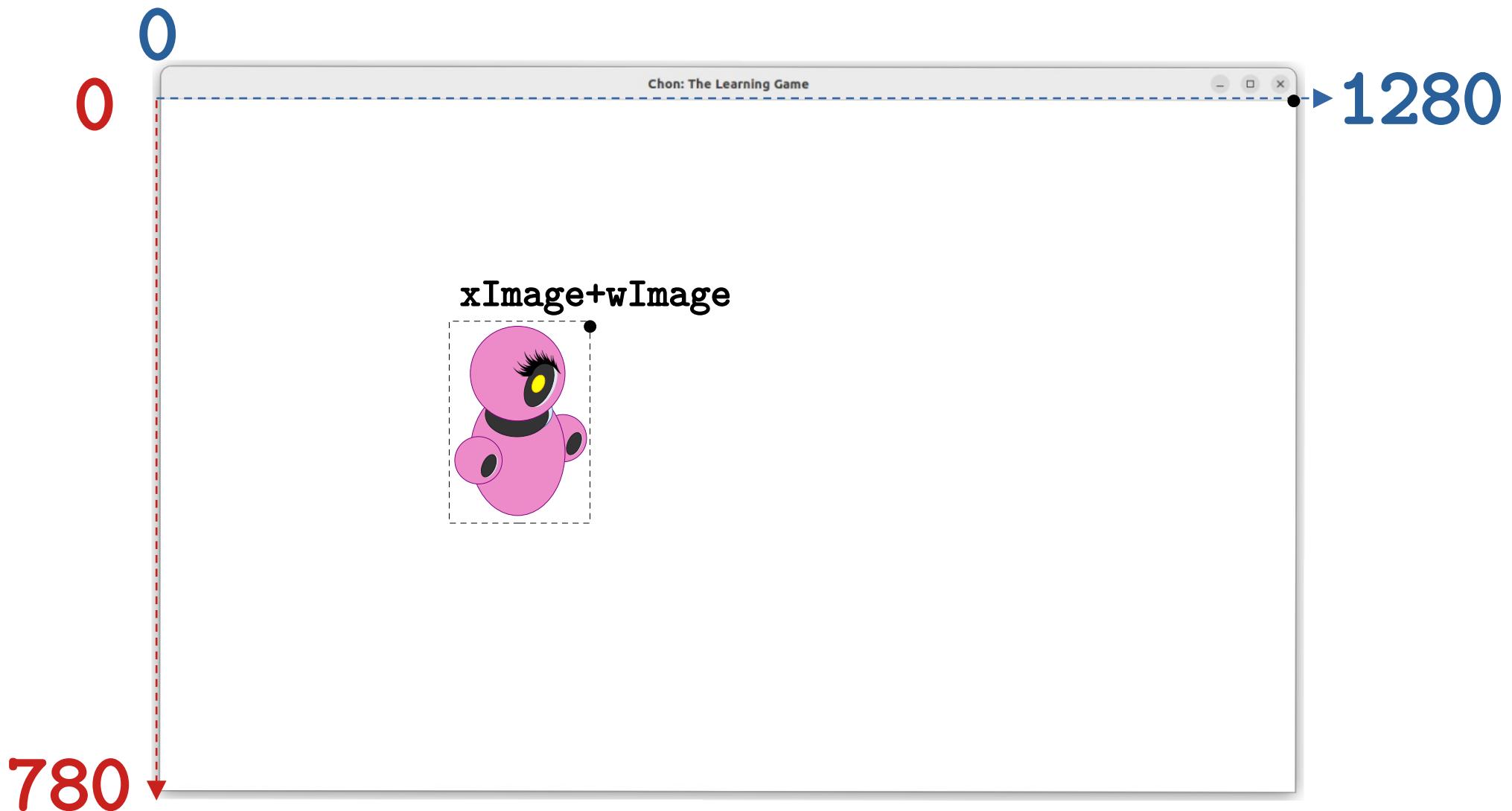
# Defining Boundaries at the LEFT



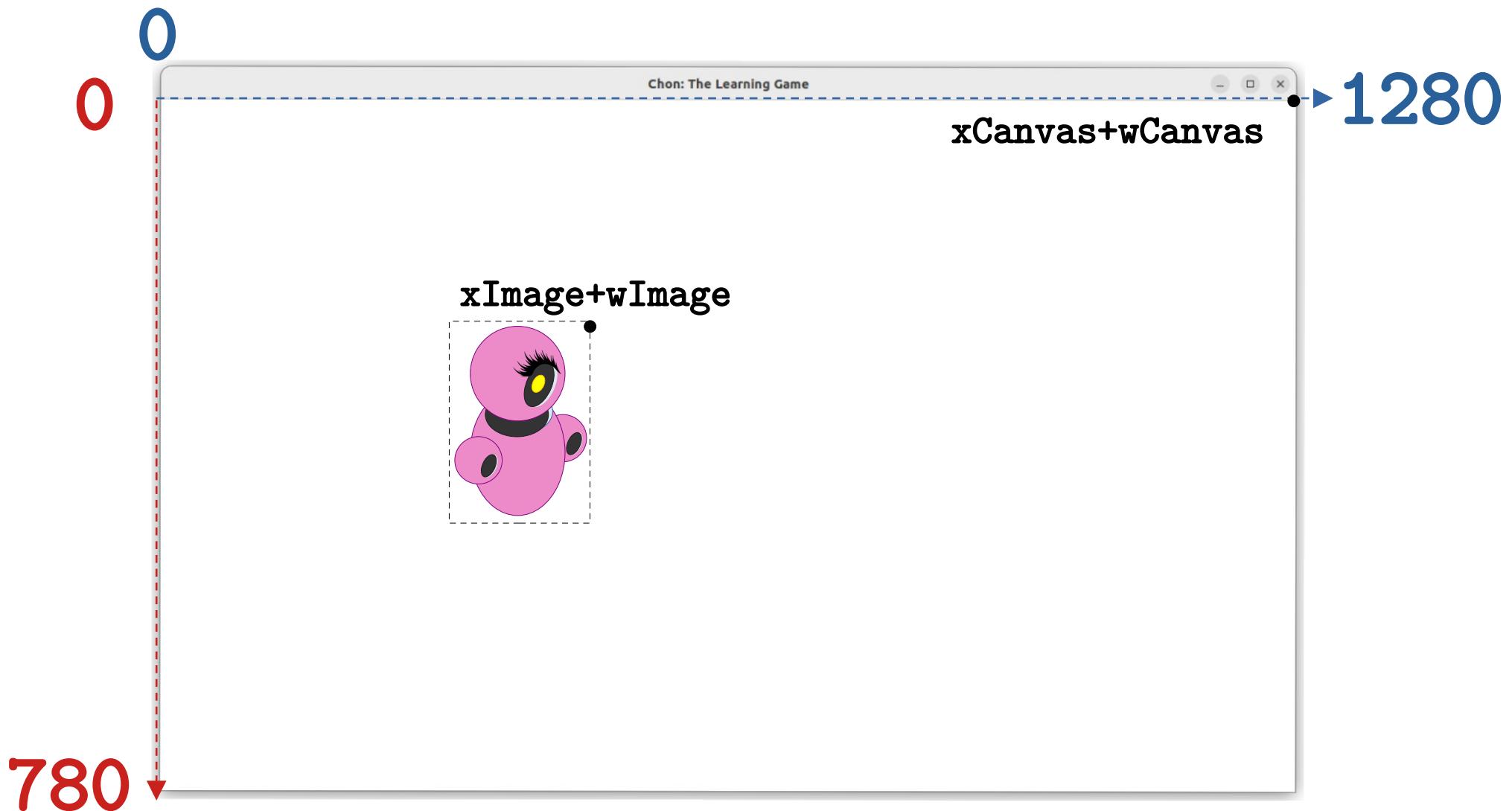
# Defining Boundaries at the RIGHT



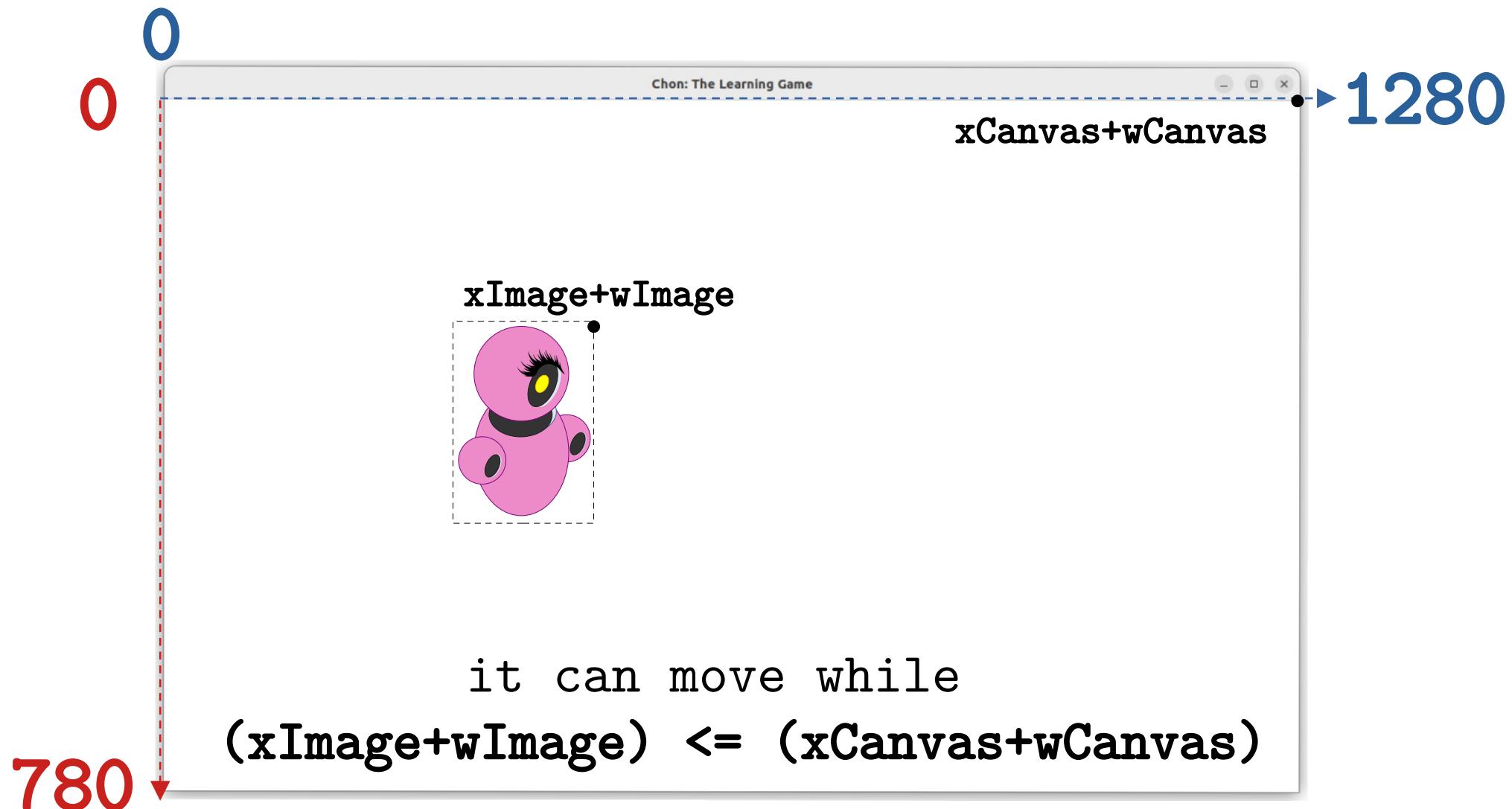
# Defining Boundaries at the RIGHT



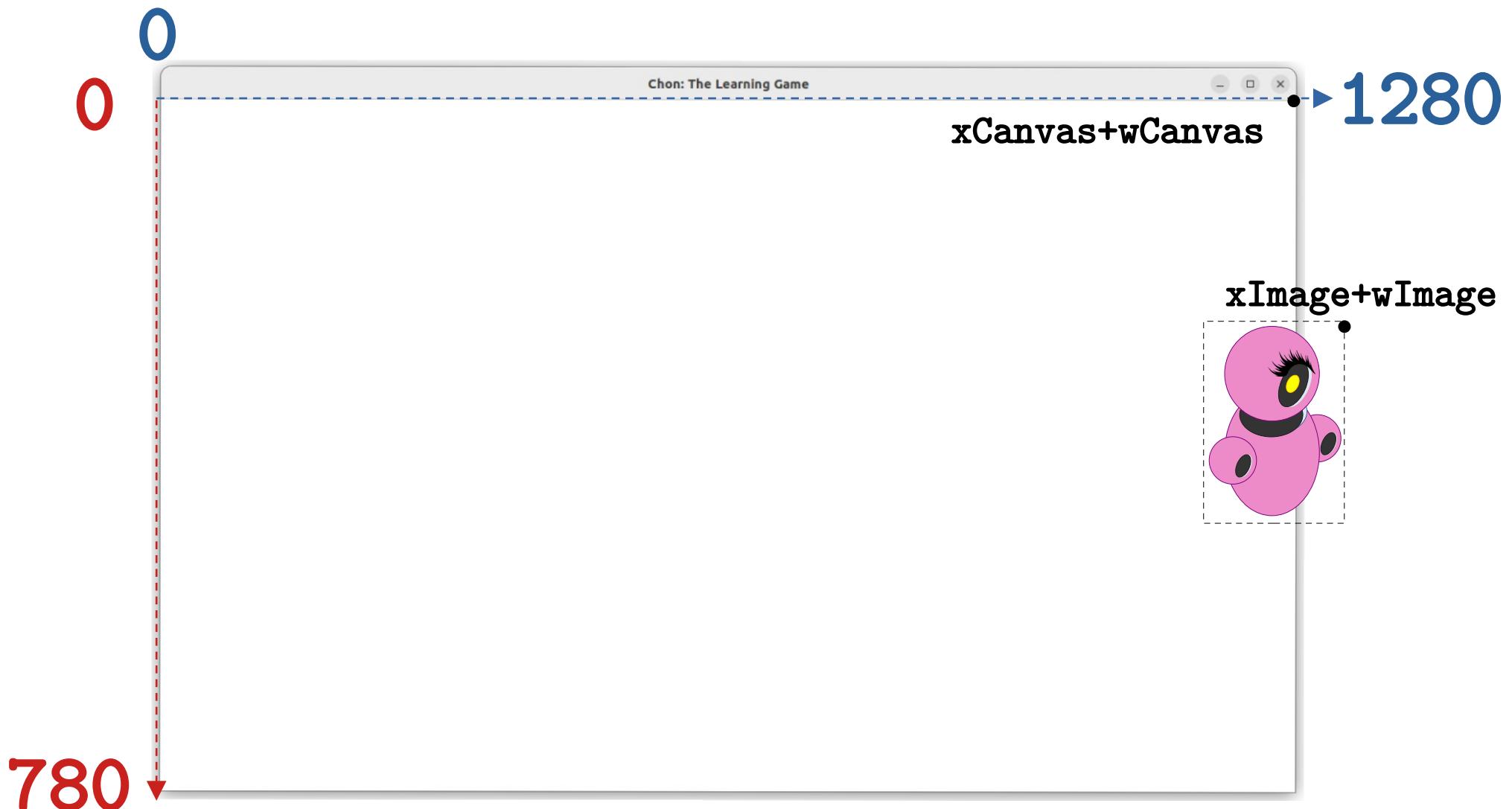
# Defining Boundaries at the RIGHT



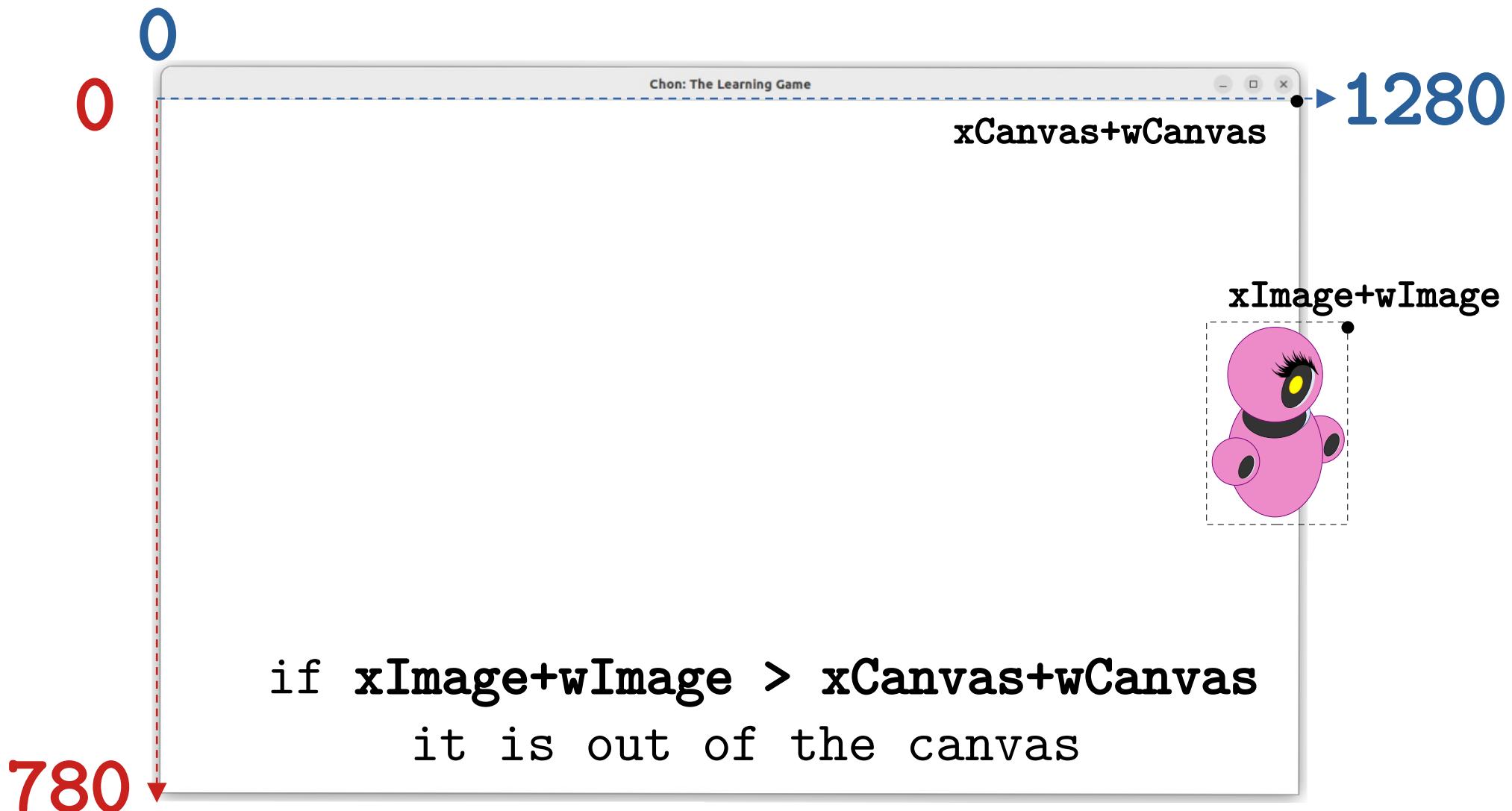
# Defining Boundaries at the RIGHT



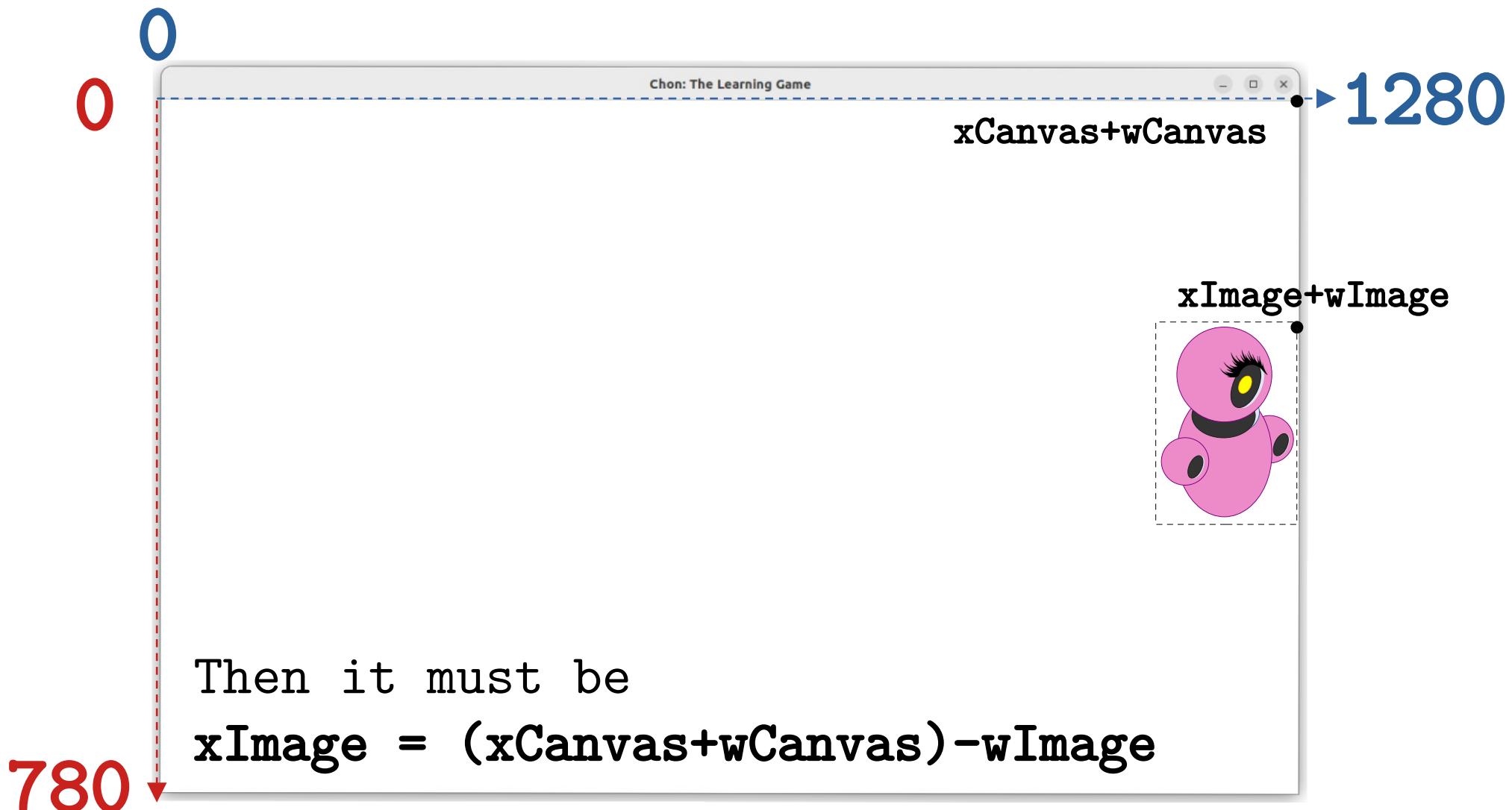
# Defining Boundaries at the RIGHT



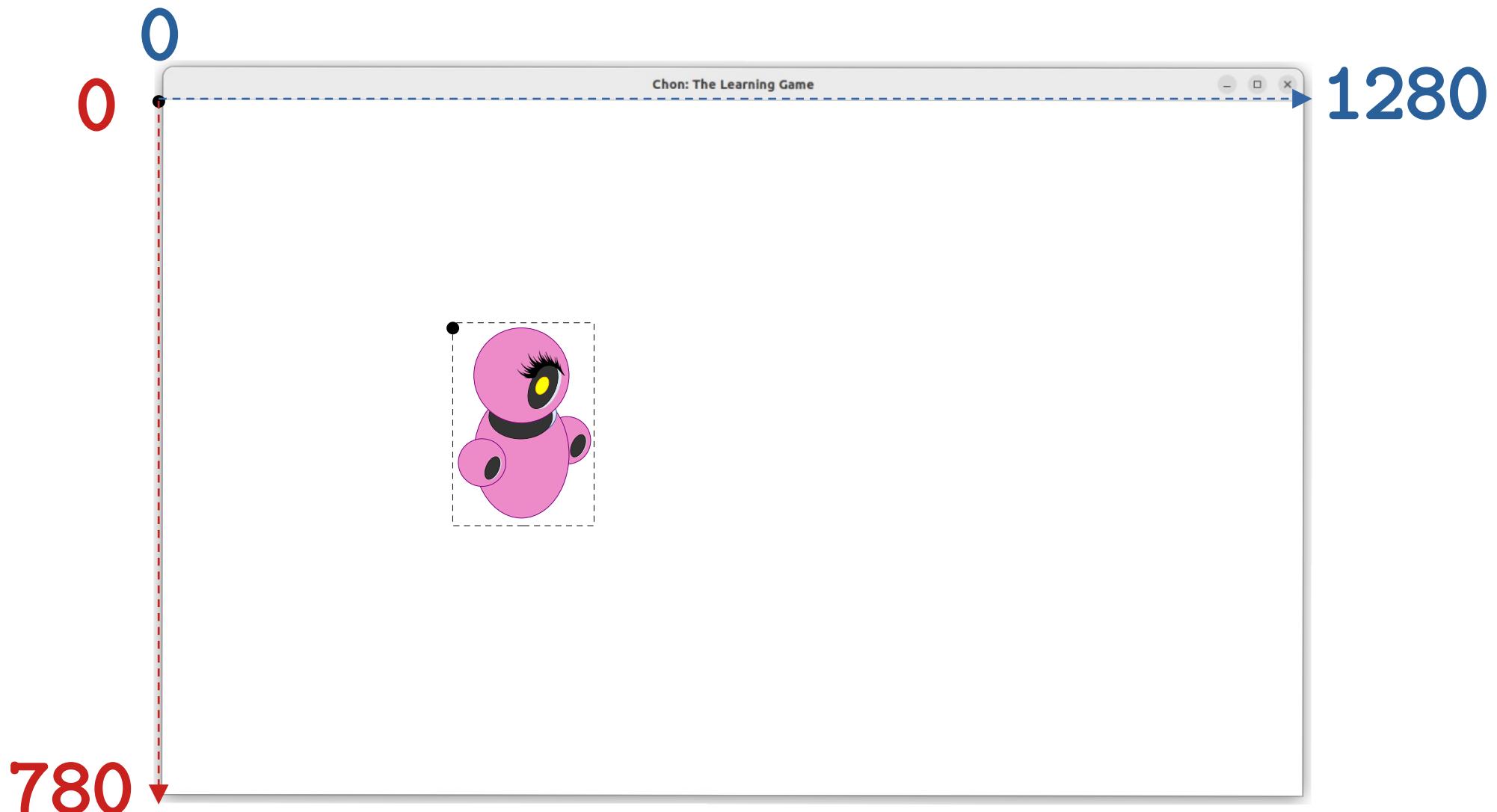
# Defining Boundaries at the RIGHT



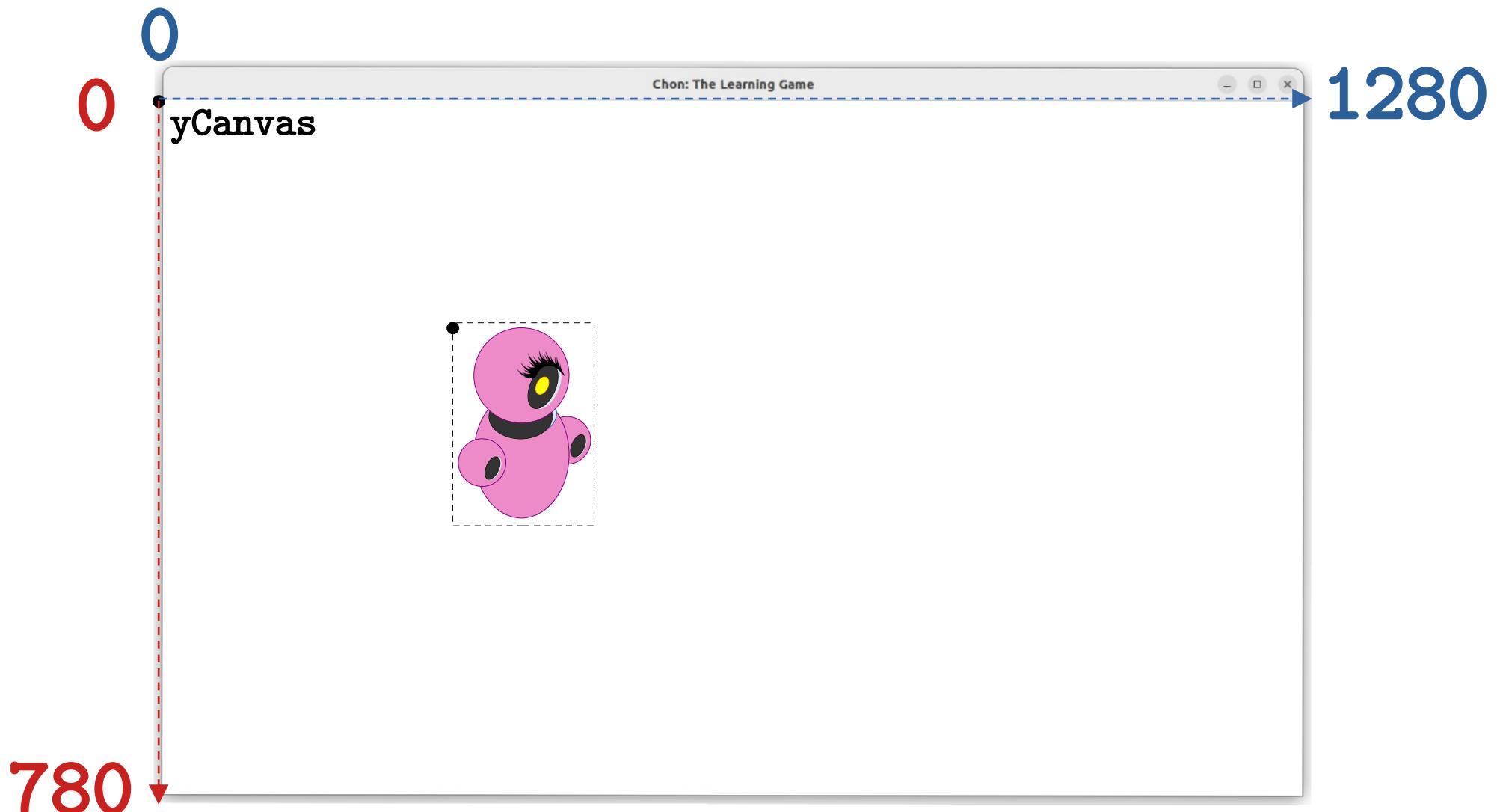
# Defining Boundaries at the RIGHT



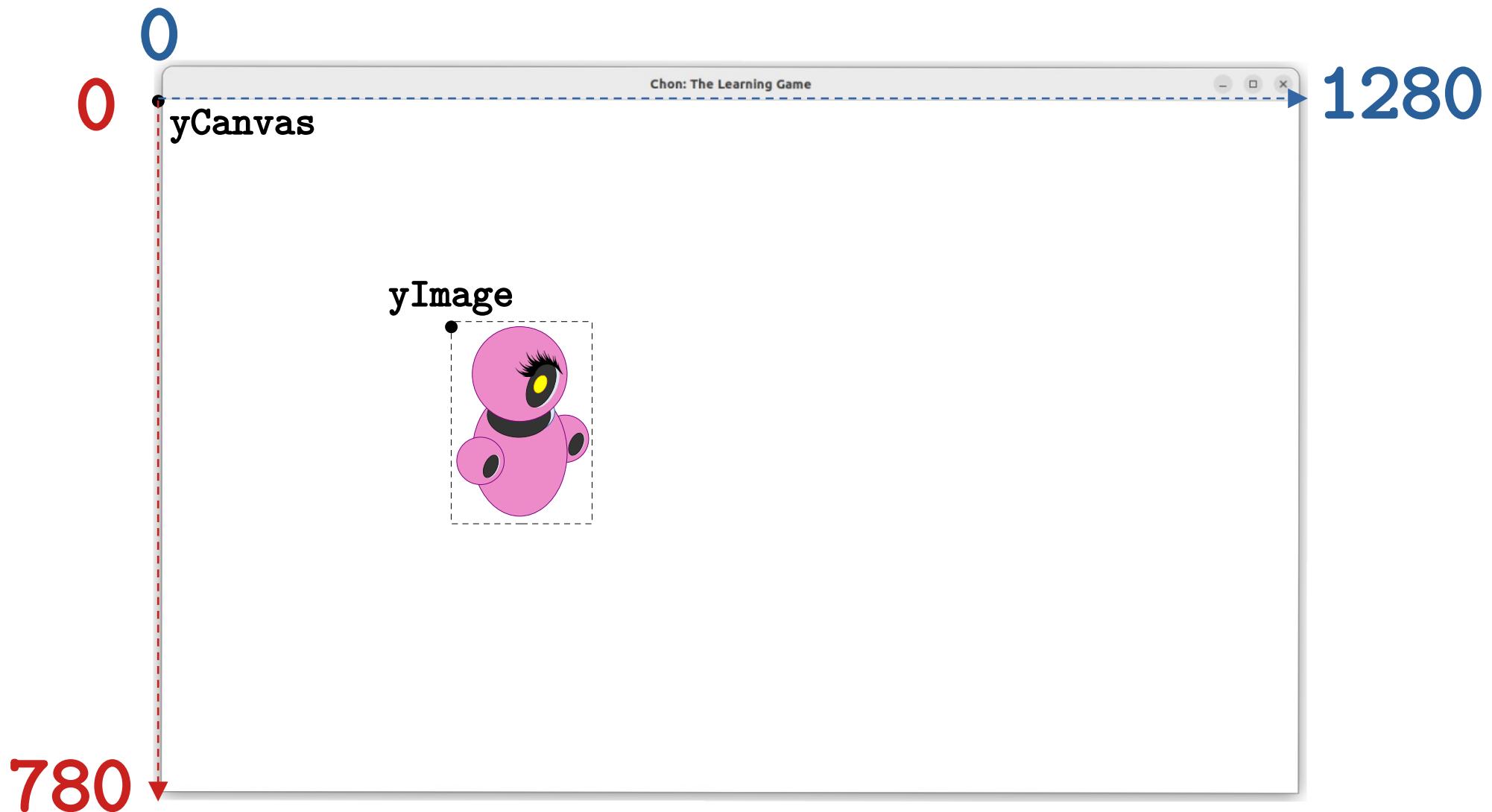
# Defining Boundaries at the TOP



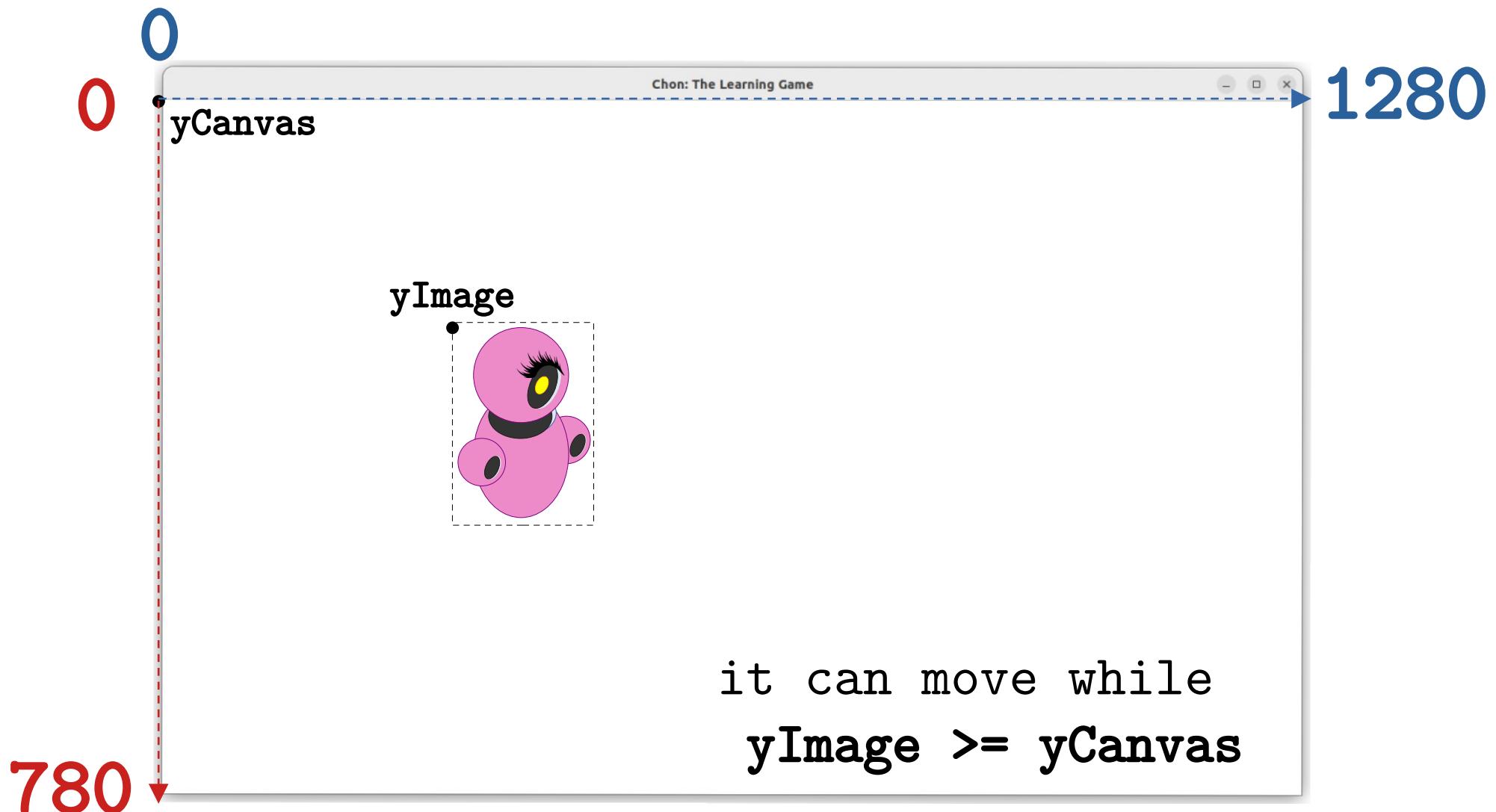
# Defining Boundaries at the TOP



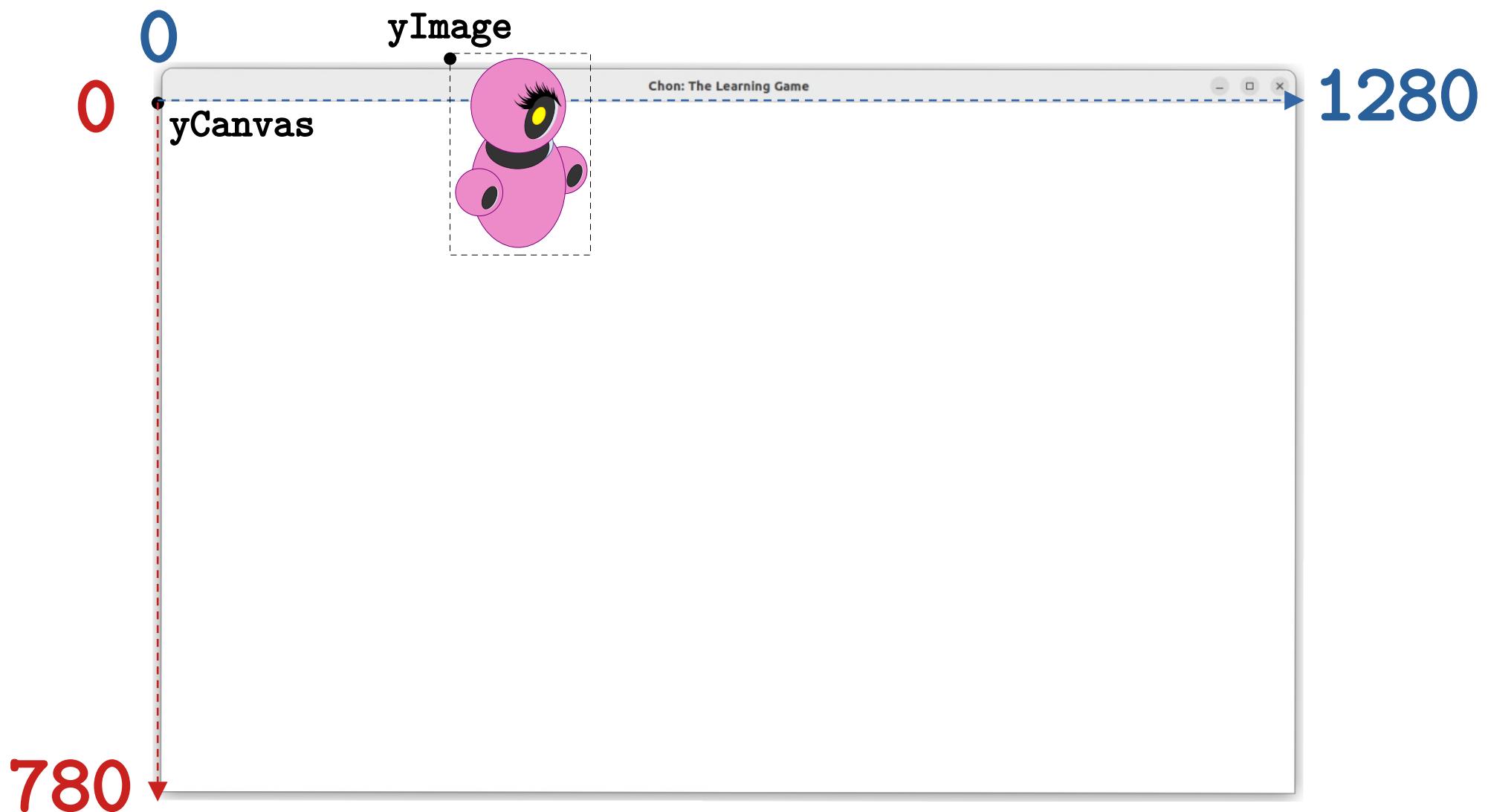
# Defining Boundaries at the TOP



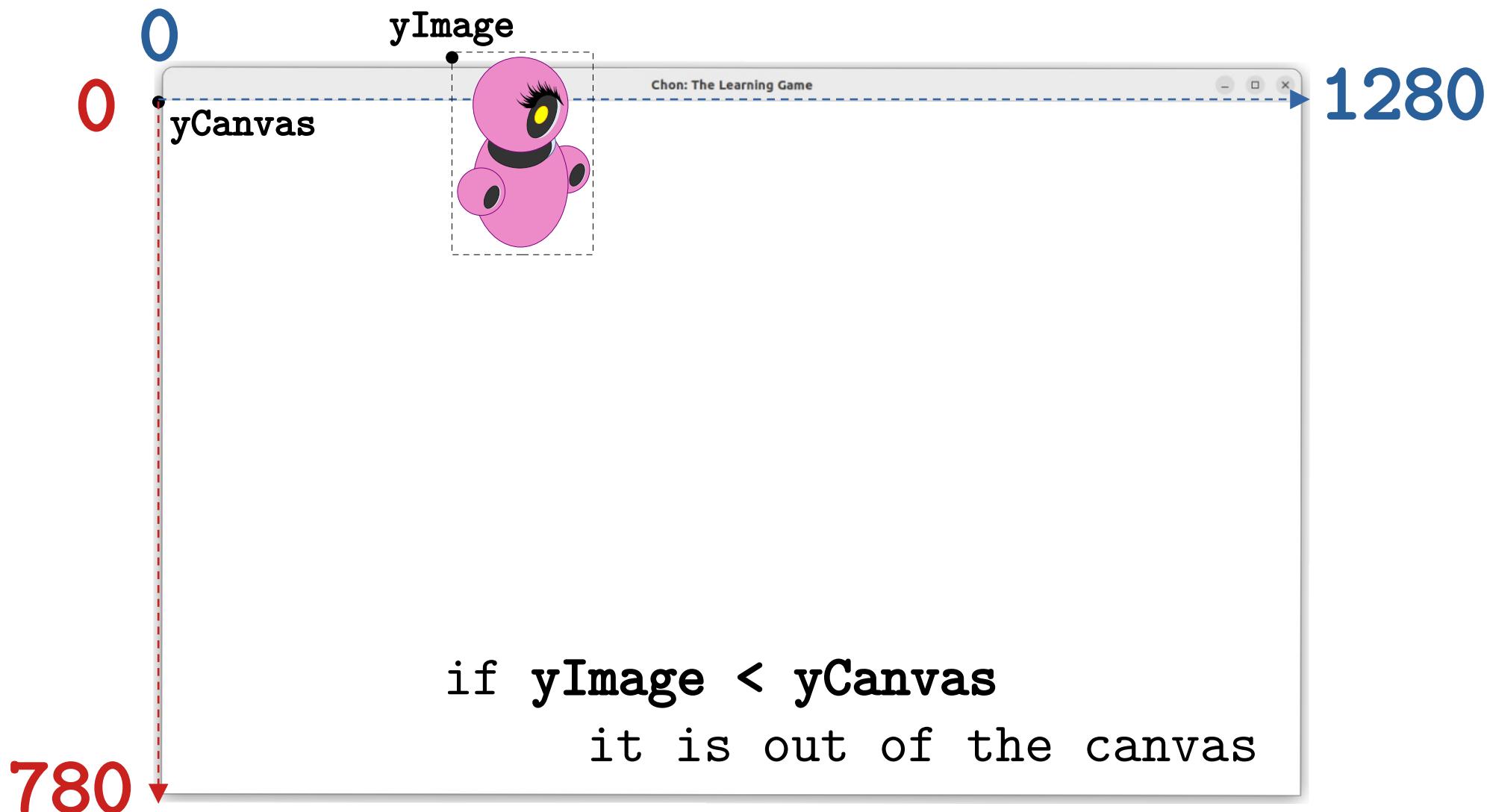
# Defining Boundaries at the TOP



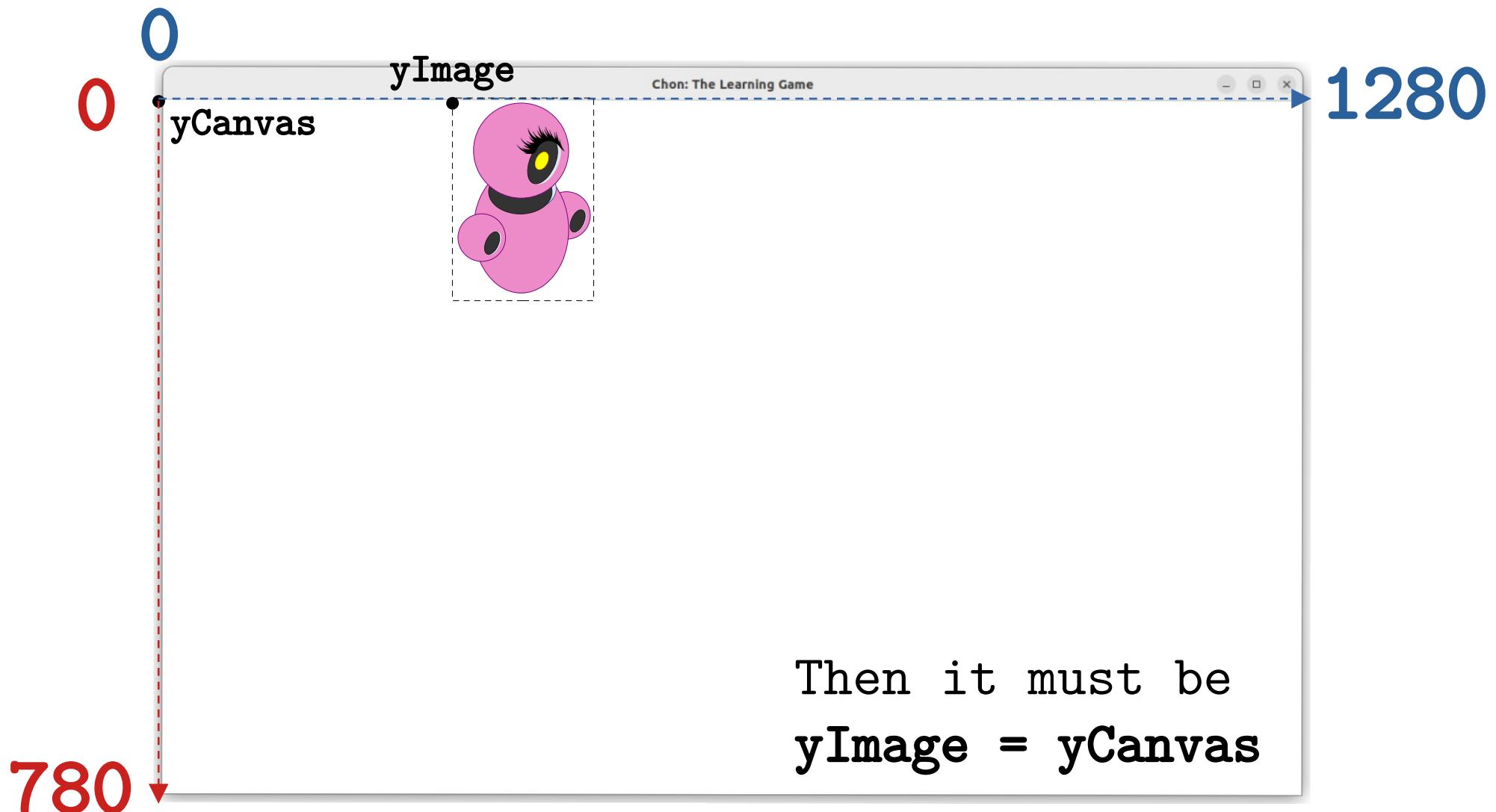
# Defining Boundaries at the TOP



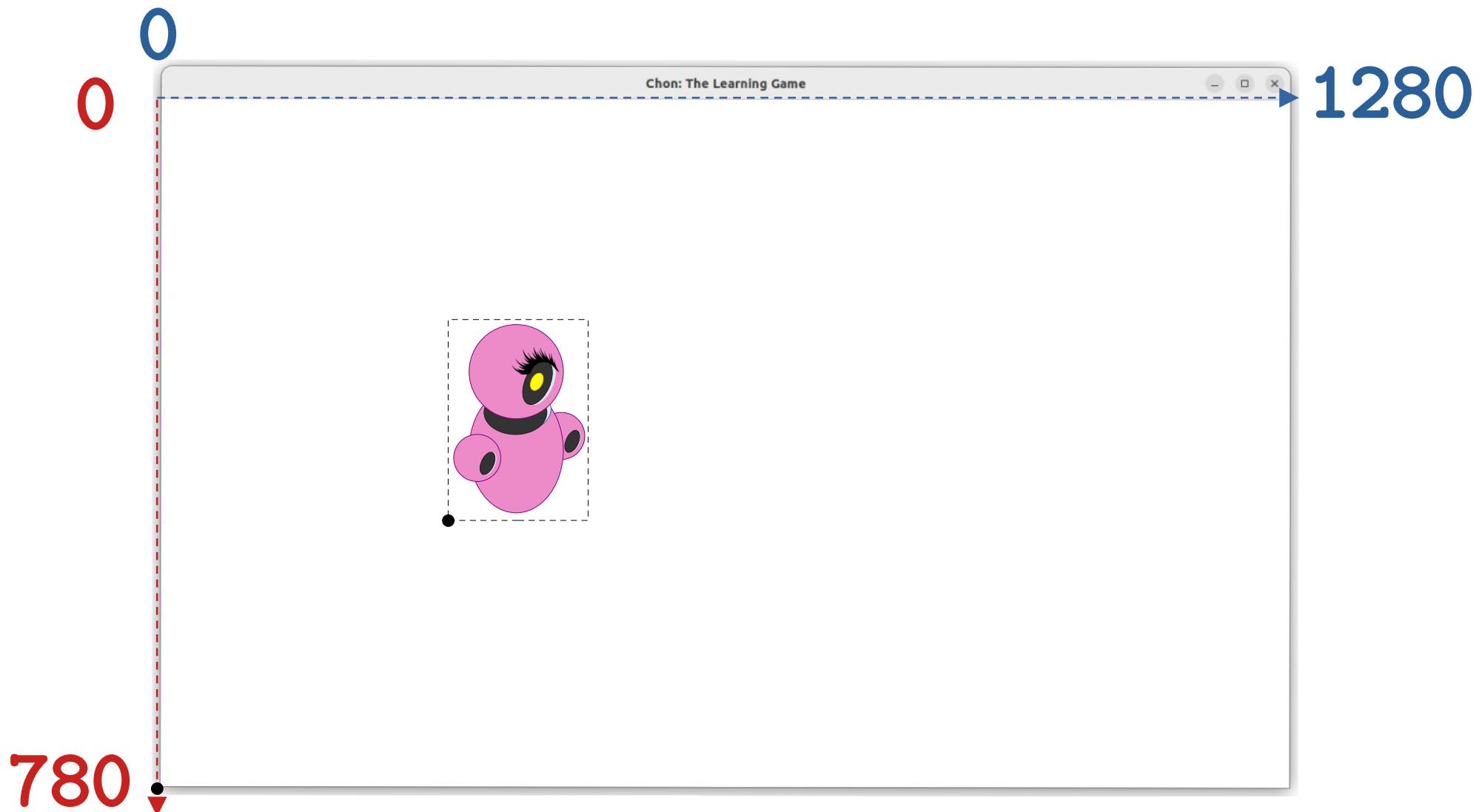
# Defining Boundaries at the TOP



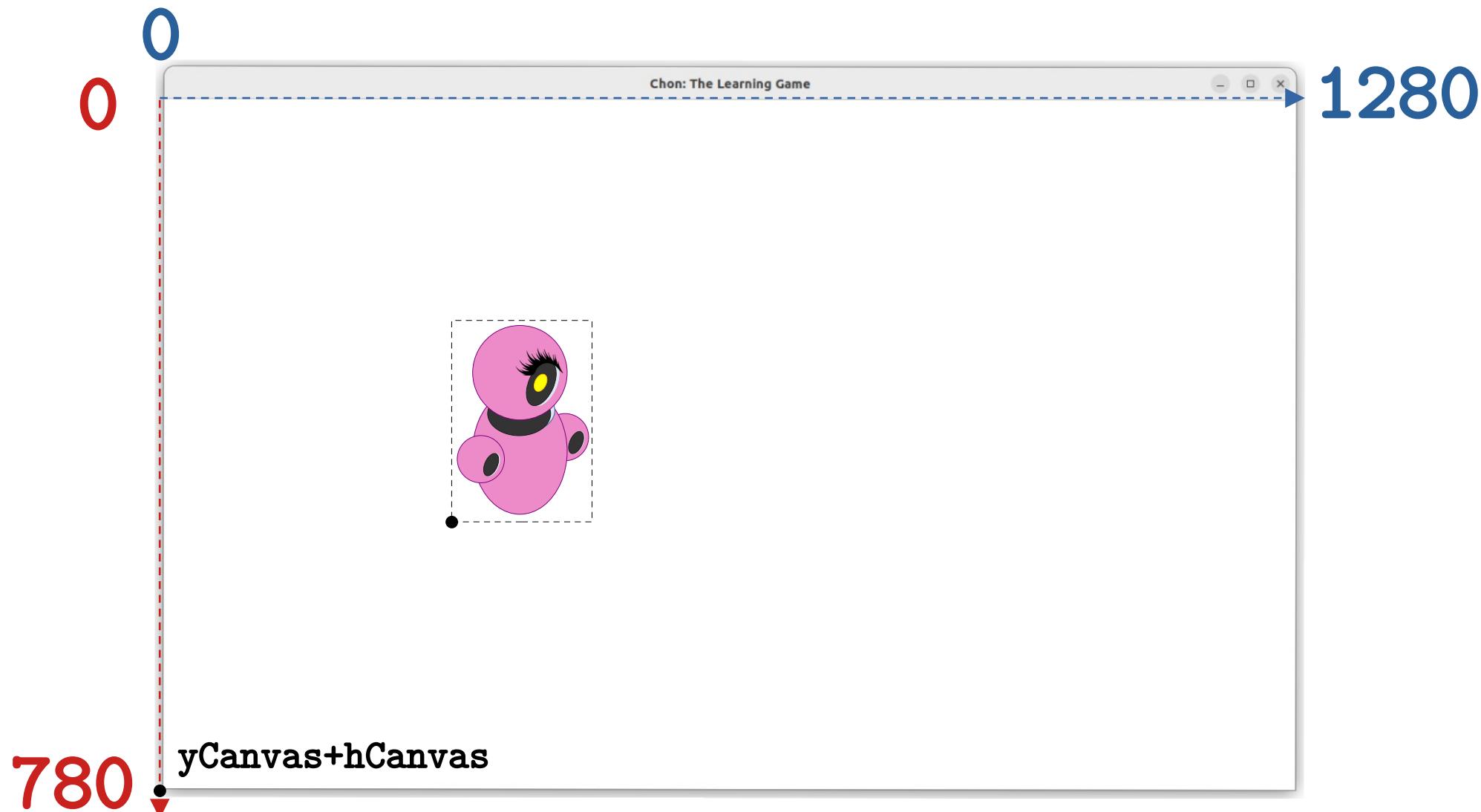
# Defining Boundaries at the TOP



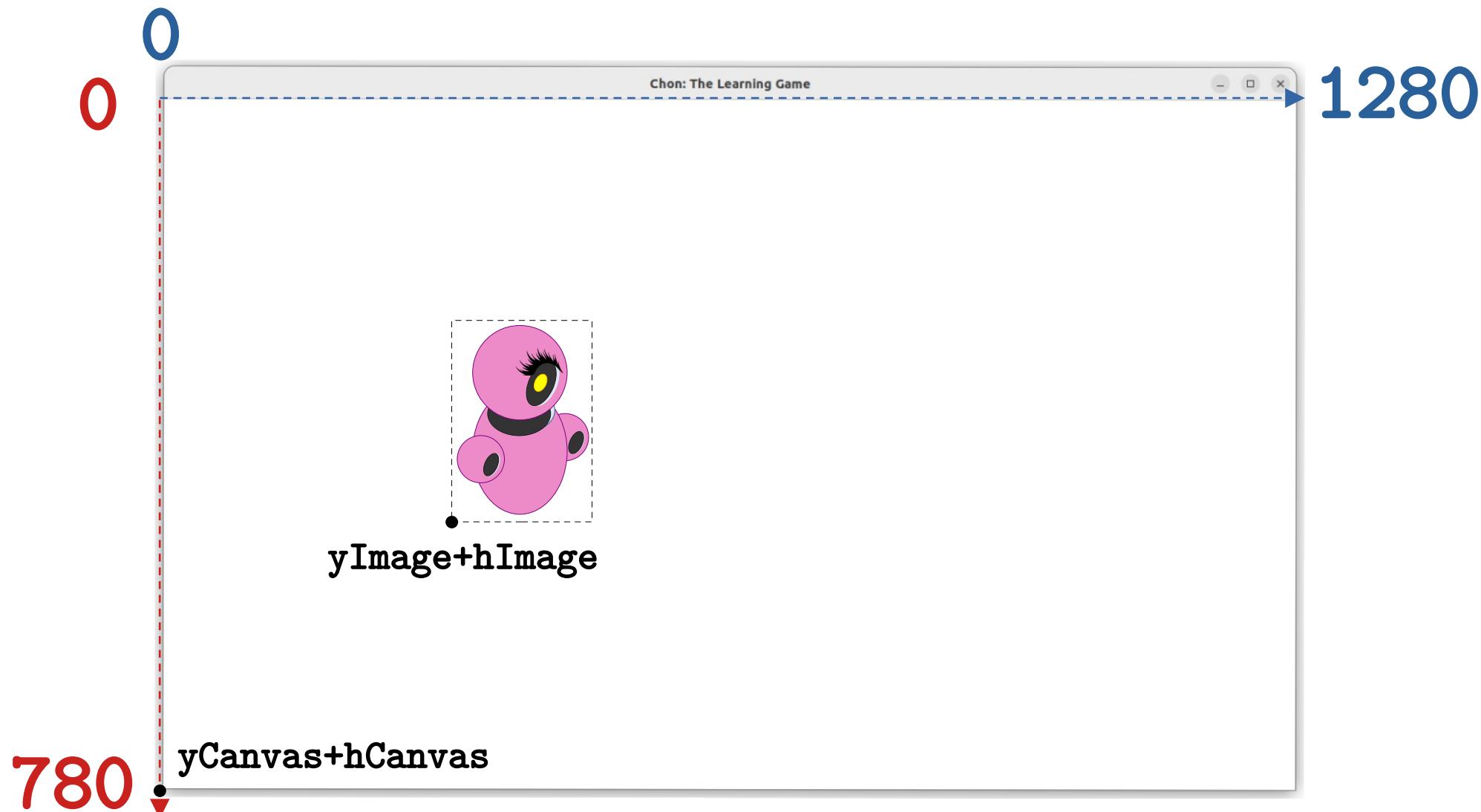
# Defining Boundaries at the BOTTOM



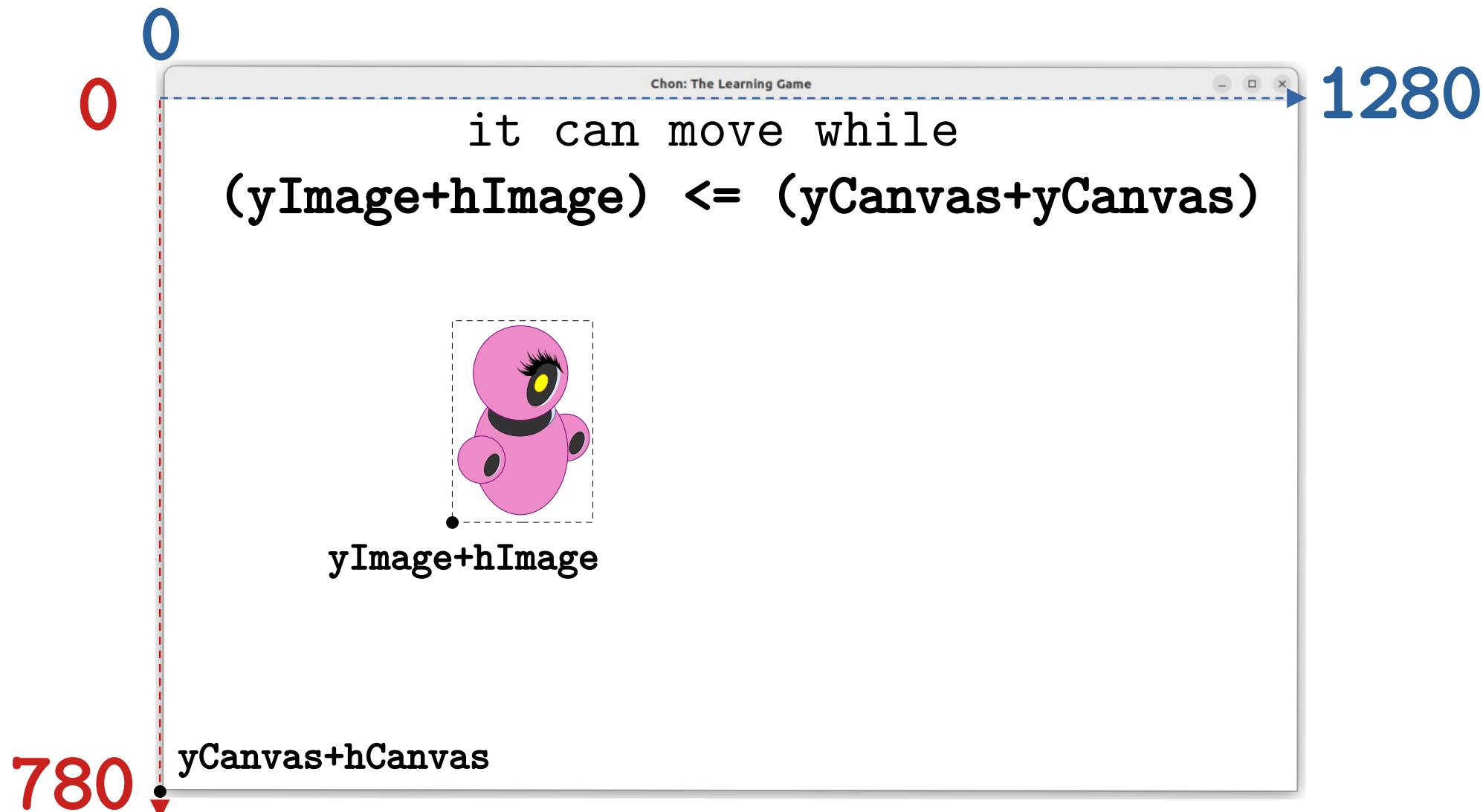
# Defining Boundaries at the BOTTOM



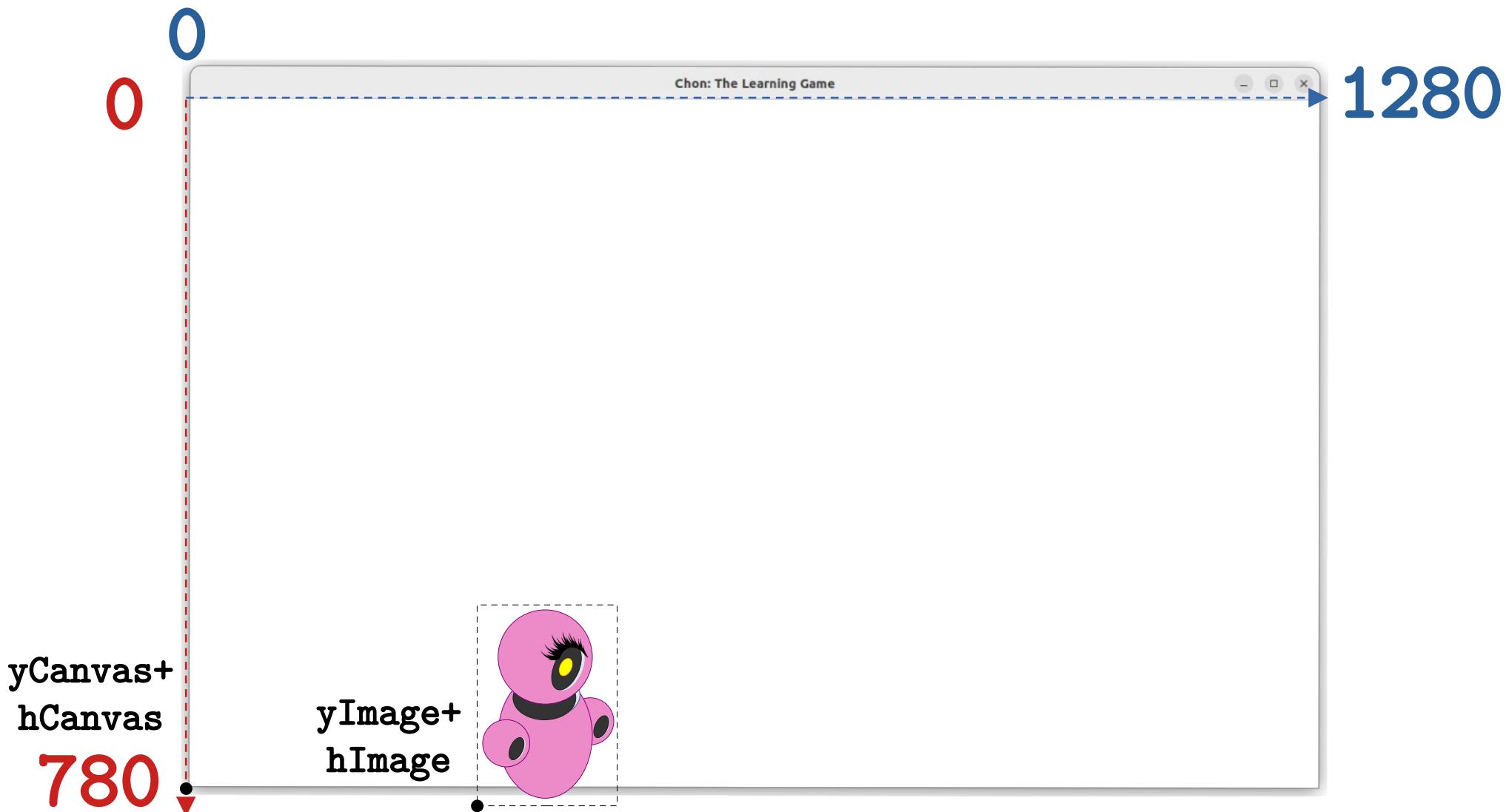
# Defining Boundaries at the BOTTOM



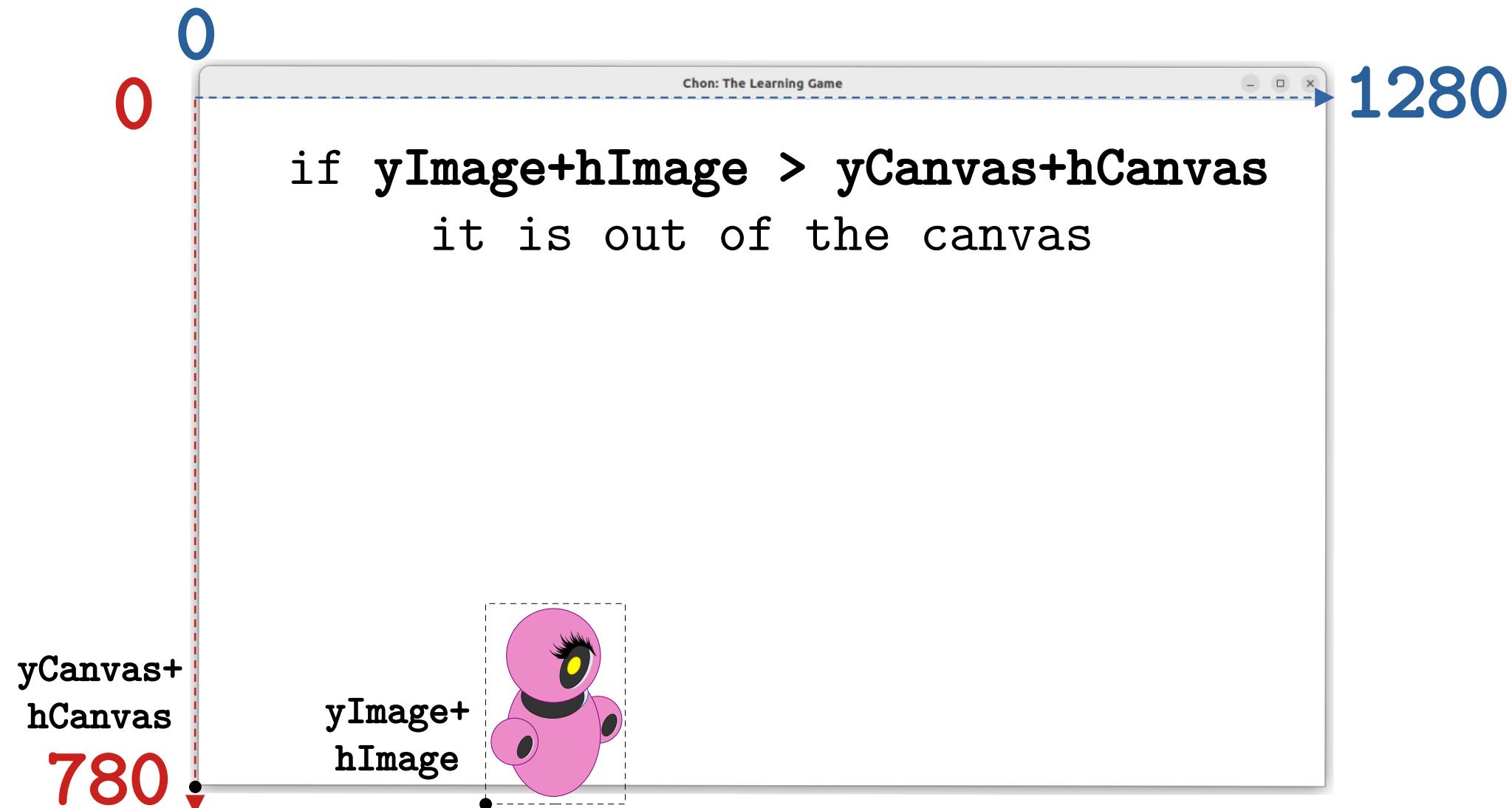
# Defining Boundaries at the BOTTOM



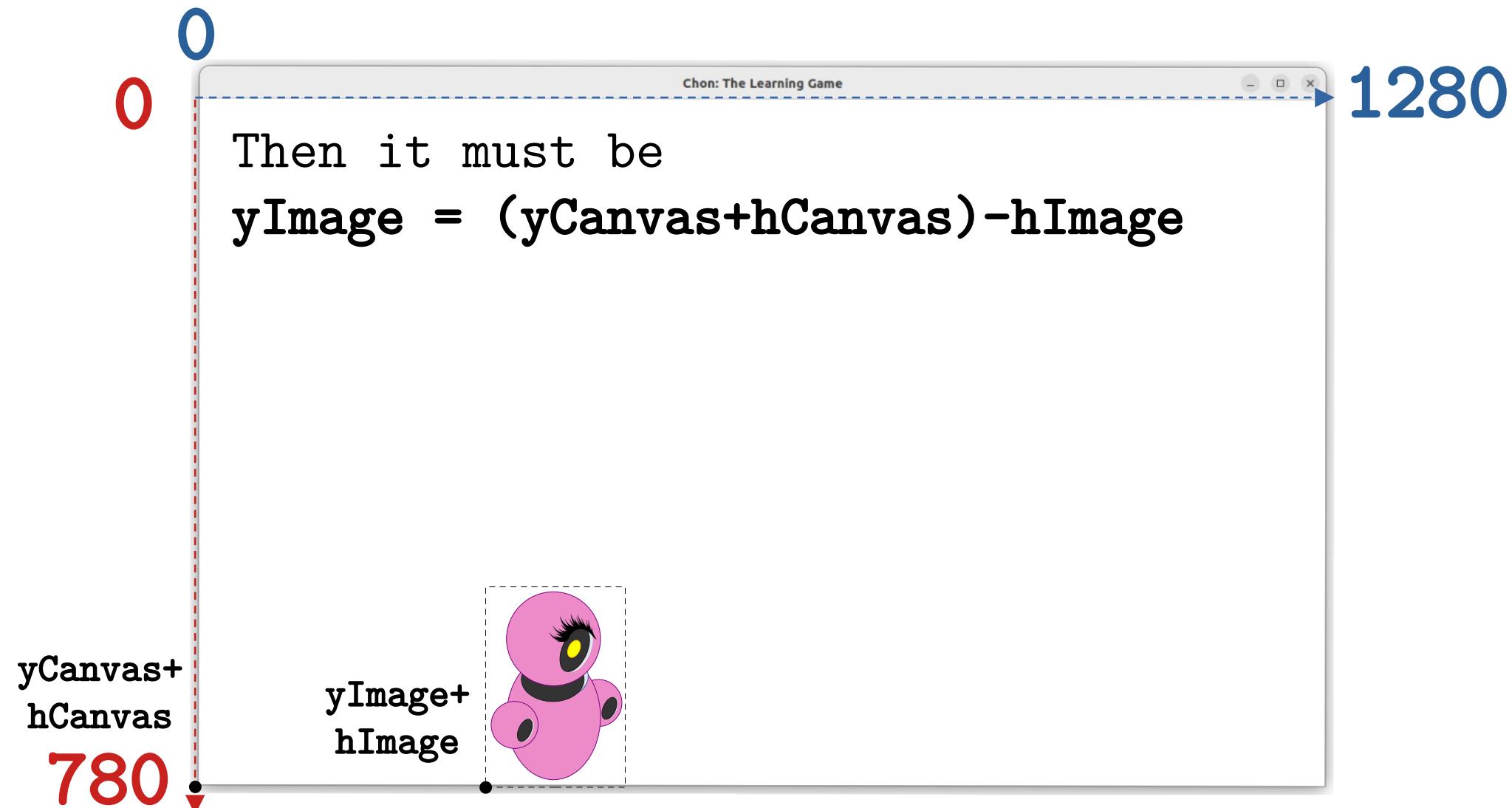
# Defining Boundaries at the BOTTOM



# Defining Boundaries at the BOTTOM



# Defining Boundaries at the BOTTOM



# Printing a Status Panel



# Printing a Status Panel



# Image Overlapping

The order in which you place the **Image** matters.

# Image Overlapping

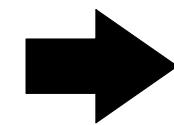
The order in which you place the **Image** matters.

```
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);  
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);
```

# Image Overlapping

The order in which you place the **Image** matters.

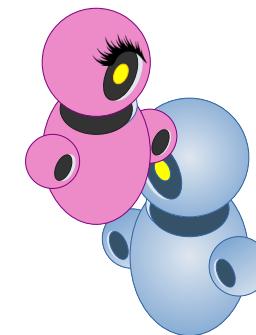
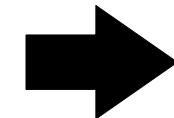
```
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);  
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);
```



# Image Overlapping

The order in which you place the **Image** matters.

```
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);  
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);
```



# Image Overlapping

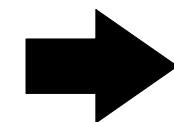
The order in which you place the **Image** matters.

```
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);
```

# Image Overlapping

The order in which you place the **Image** matters.

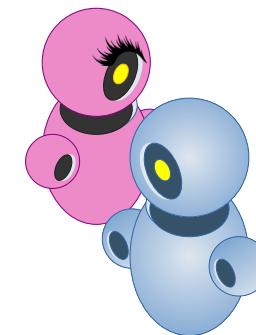
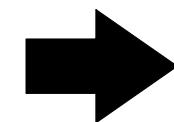
```
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);  
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);
```



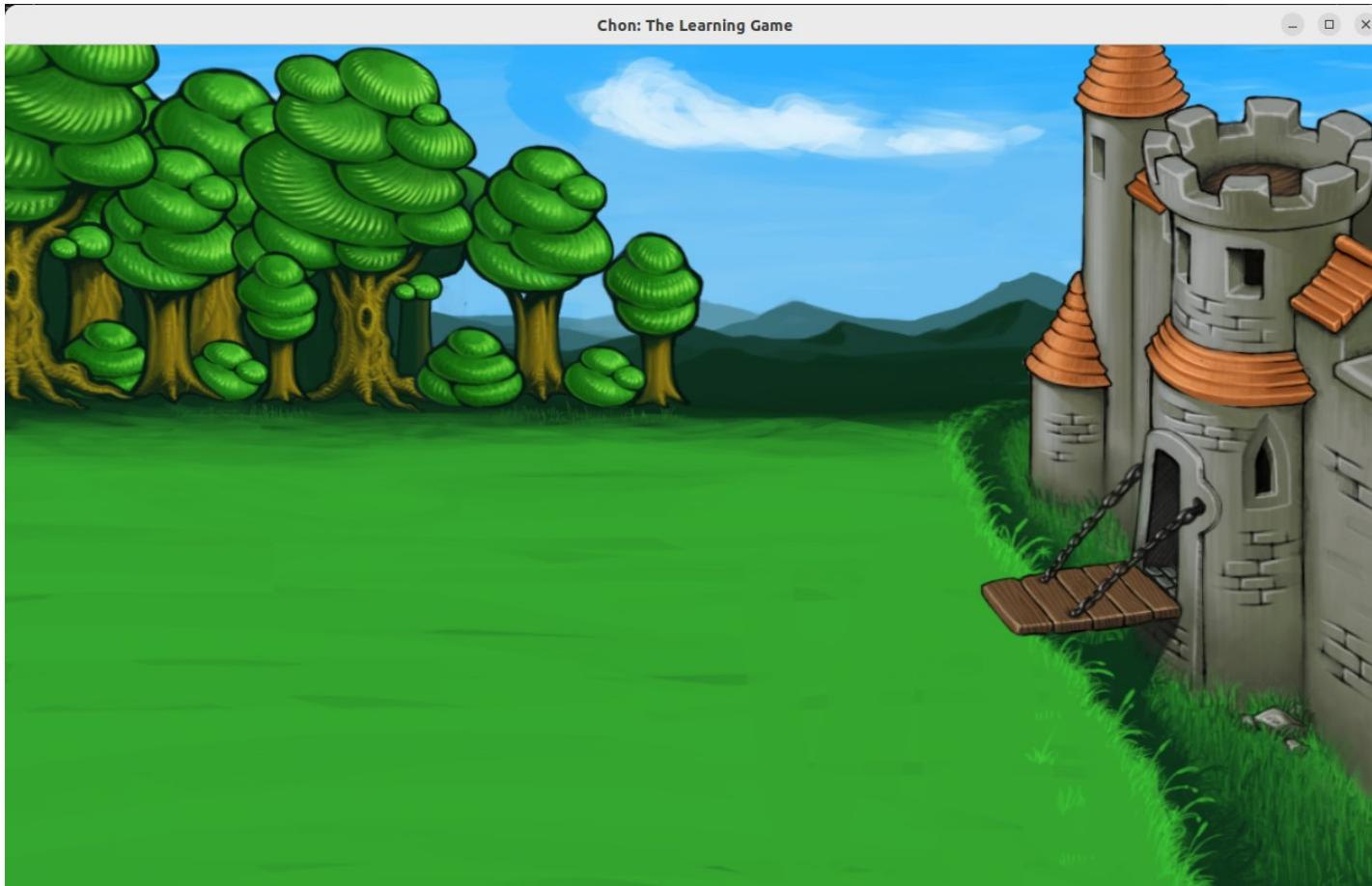
# Image Overlapping

The order in which you place the **Image** matters.

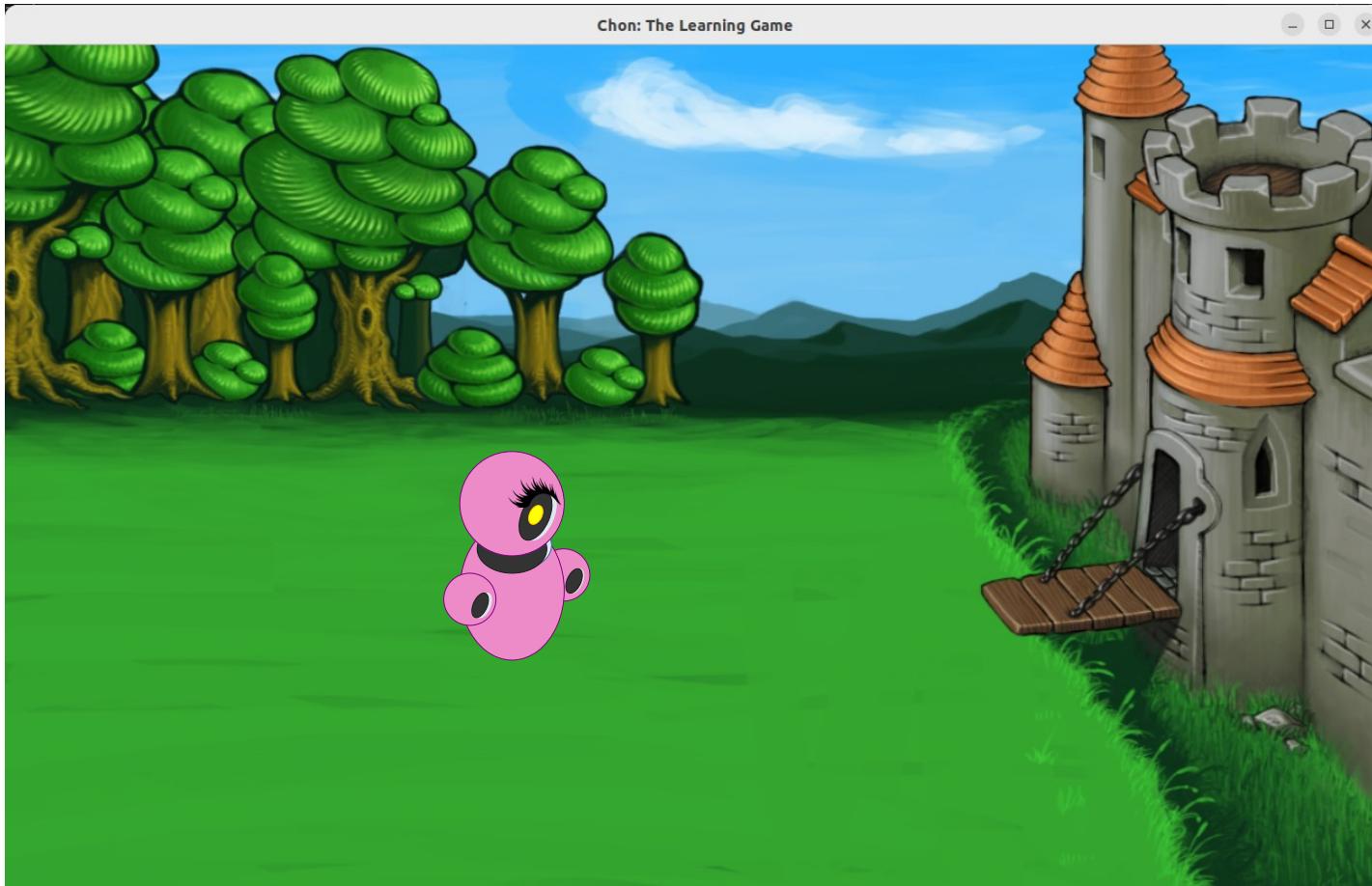
```
gc.drawImage(chonBota, x:400, y:390, wImageBot, hImageBot);  
gc.drawImage(chonBot, x:920, y:440, wImageBot, hImageBot);
```



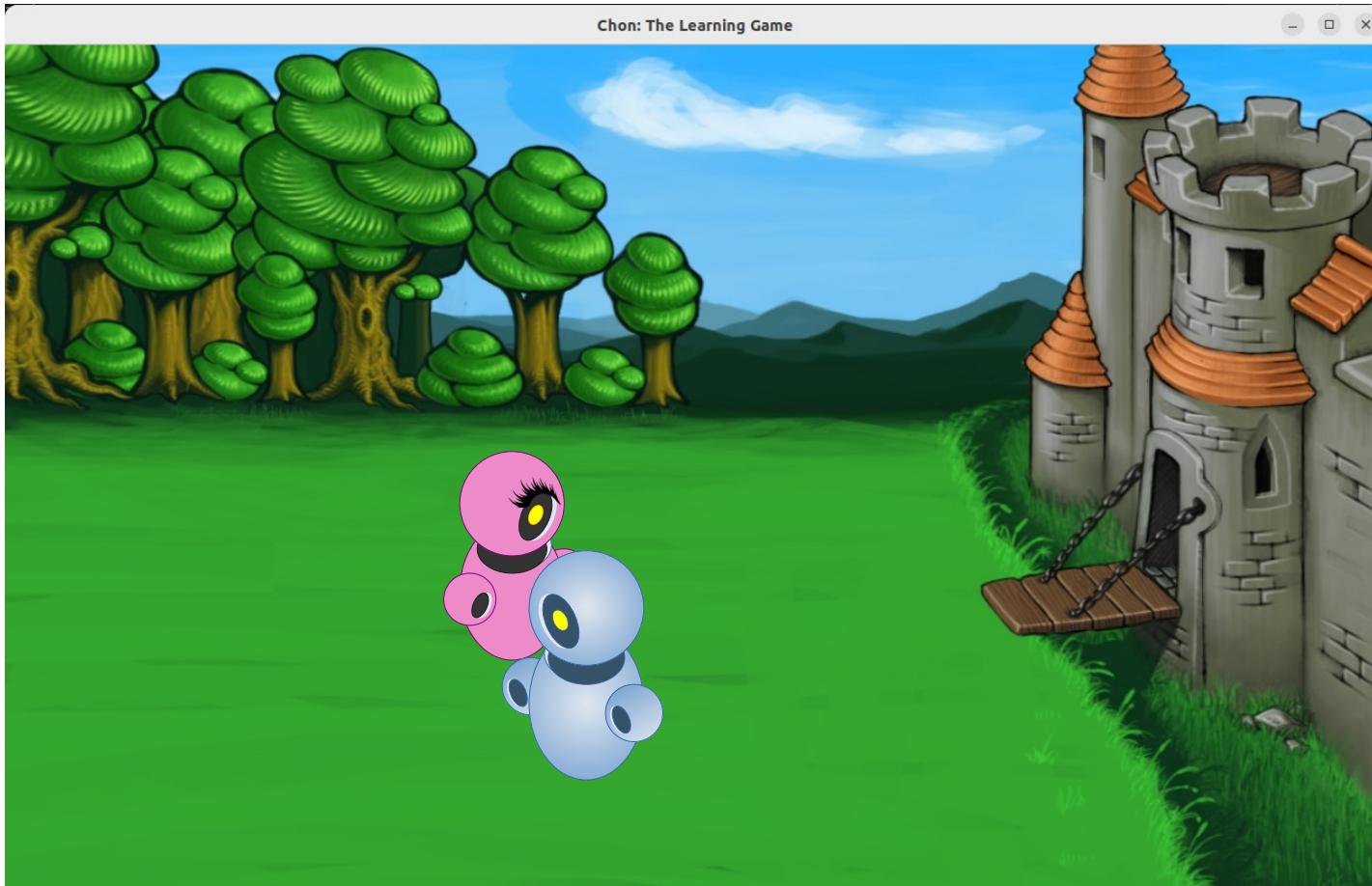
# Image Overlapping



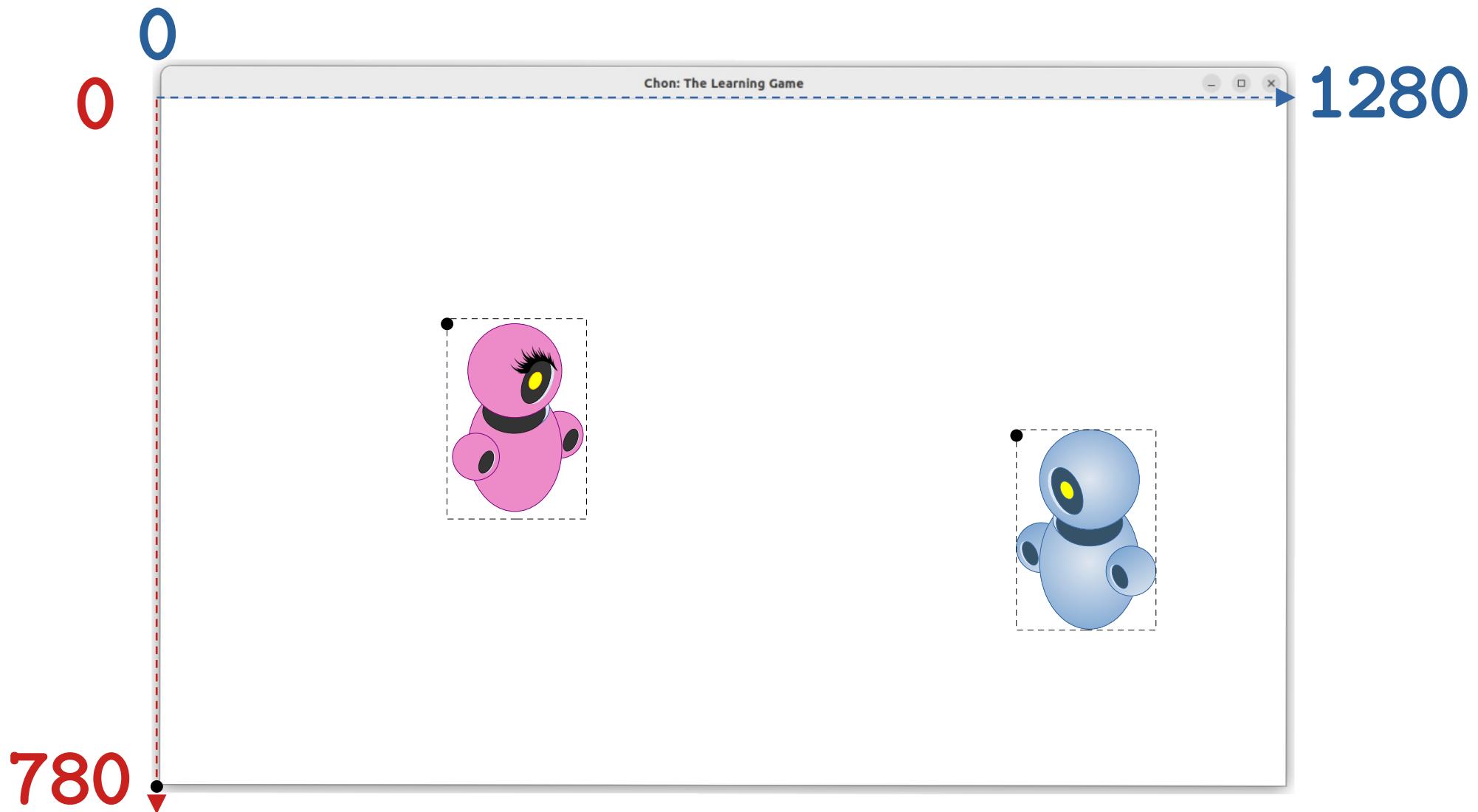
# Image Overlapping



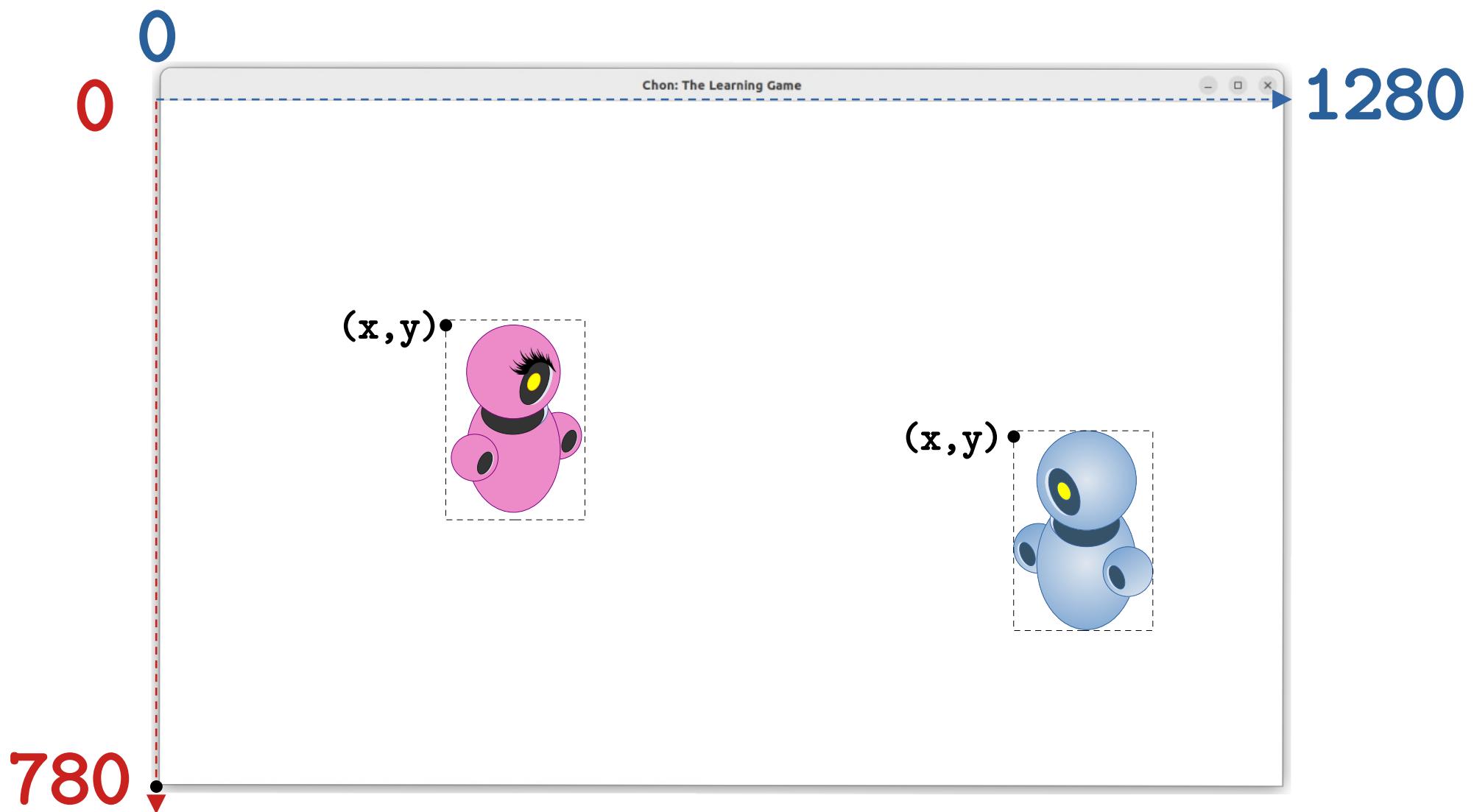
# Image Overlapping



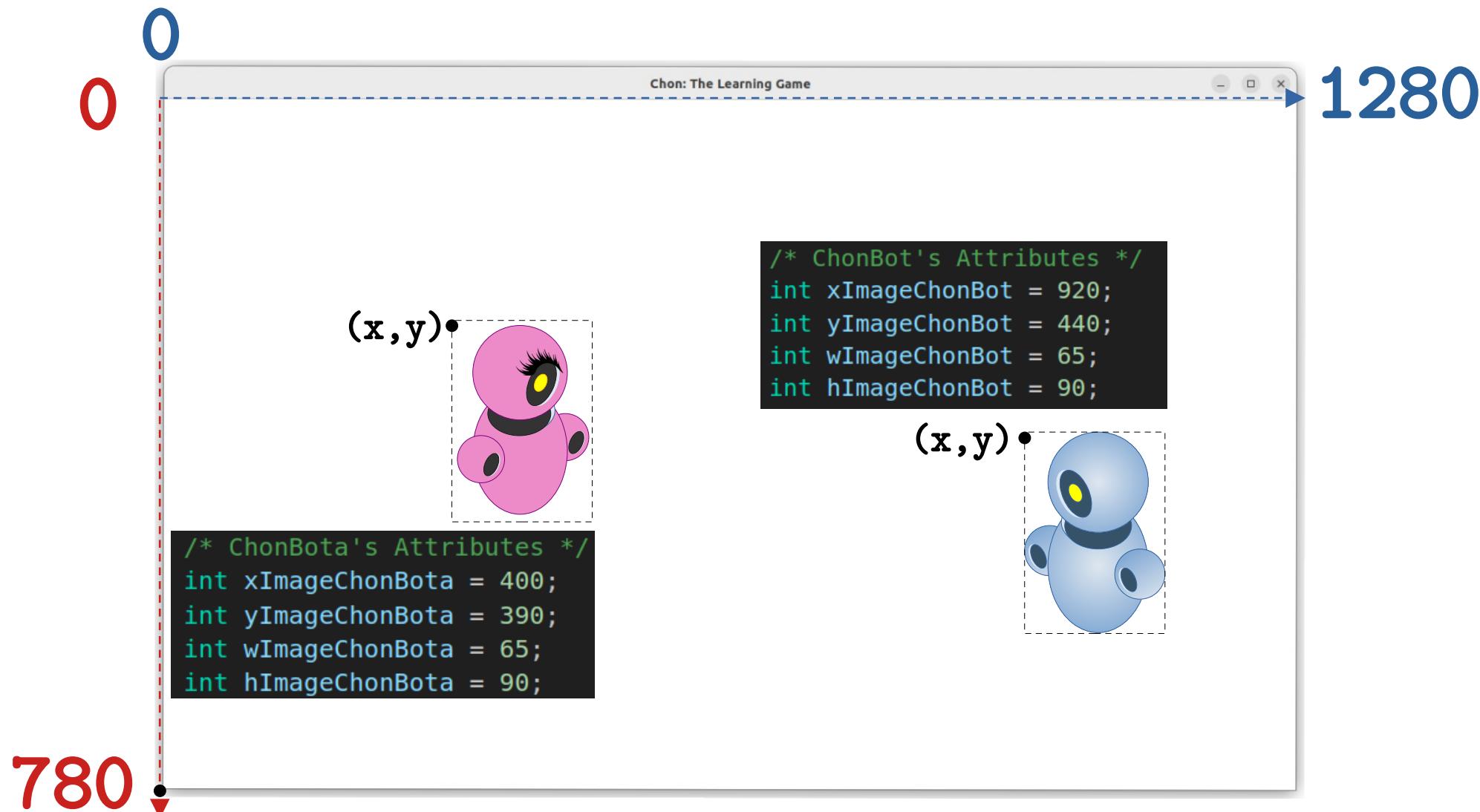
# Moving Another Object



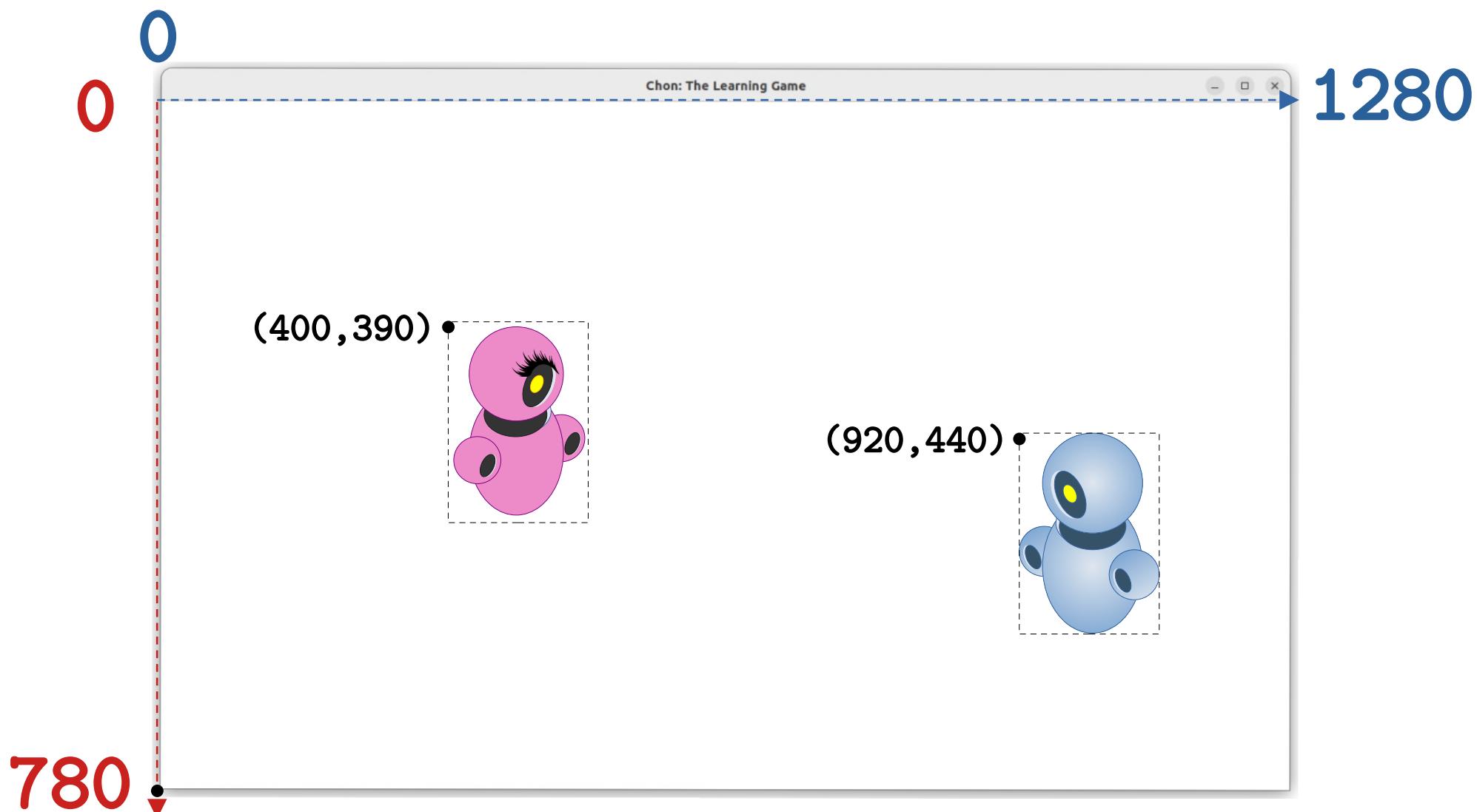
# Moving Another Object



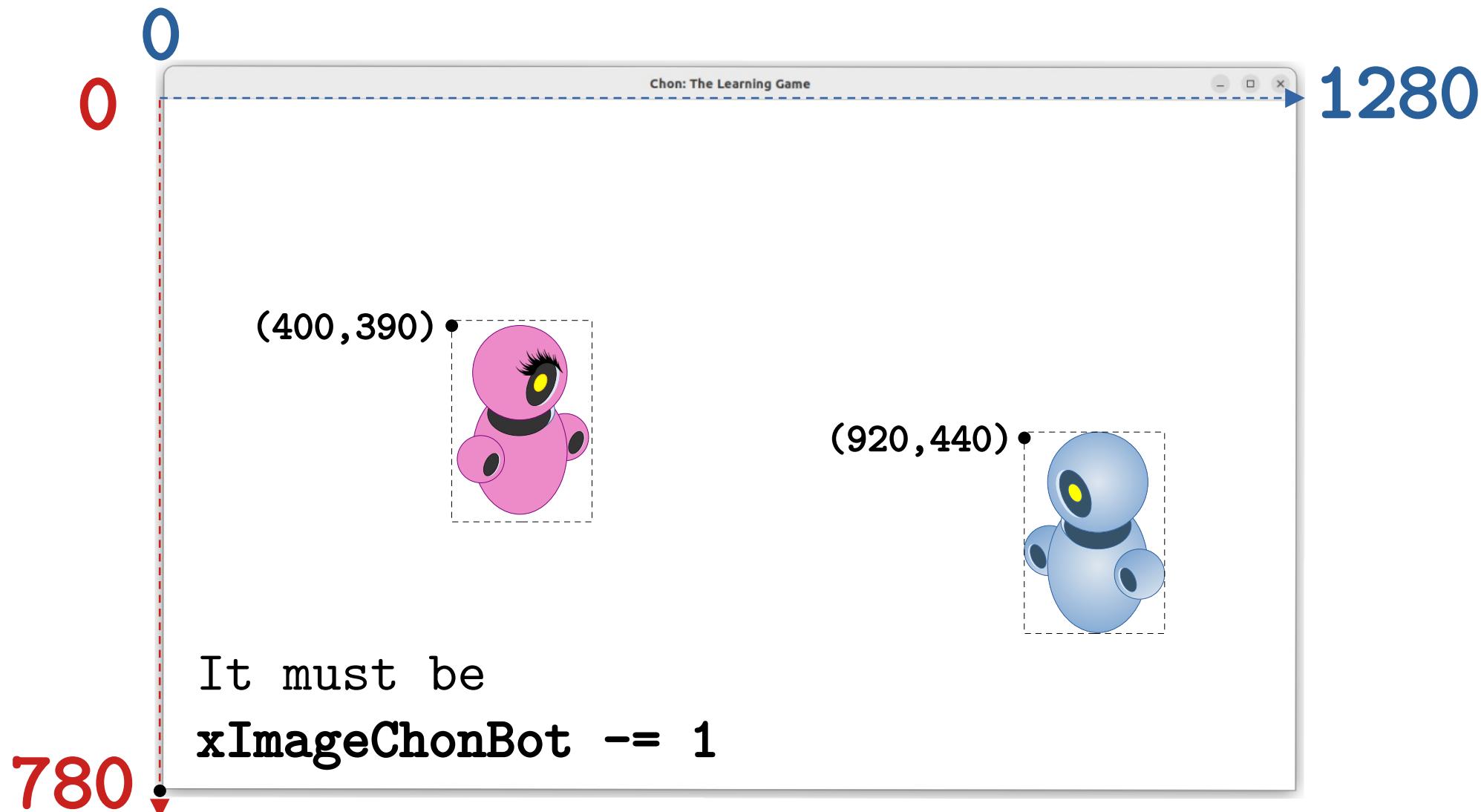
# Moving Another Object



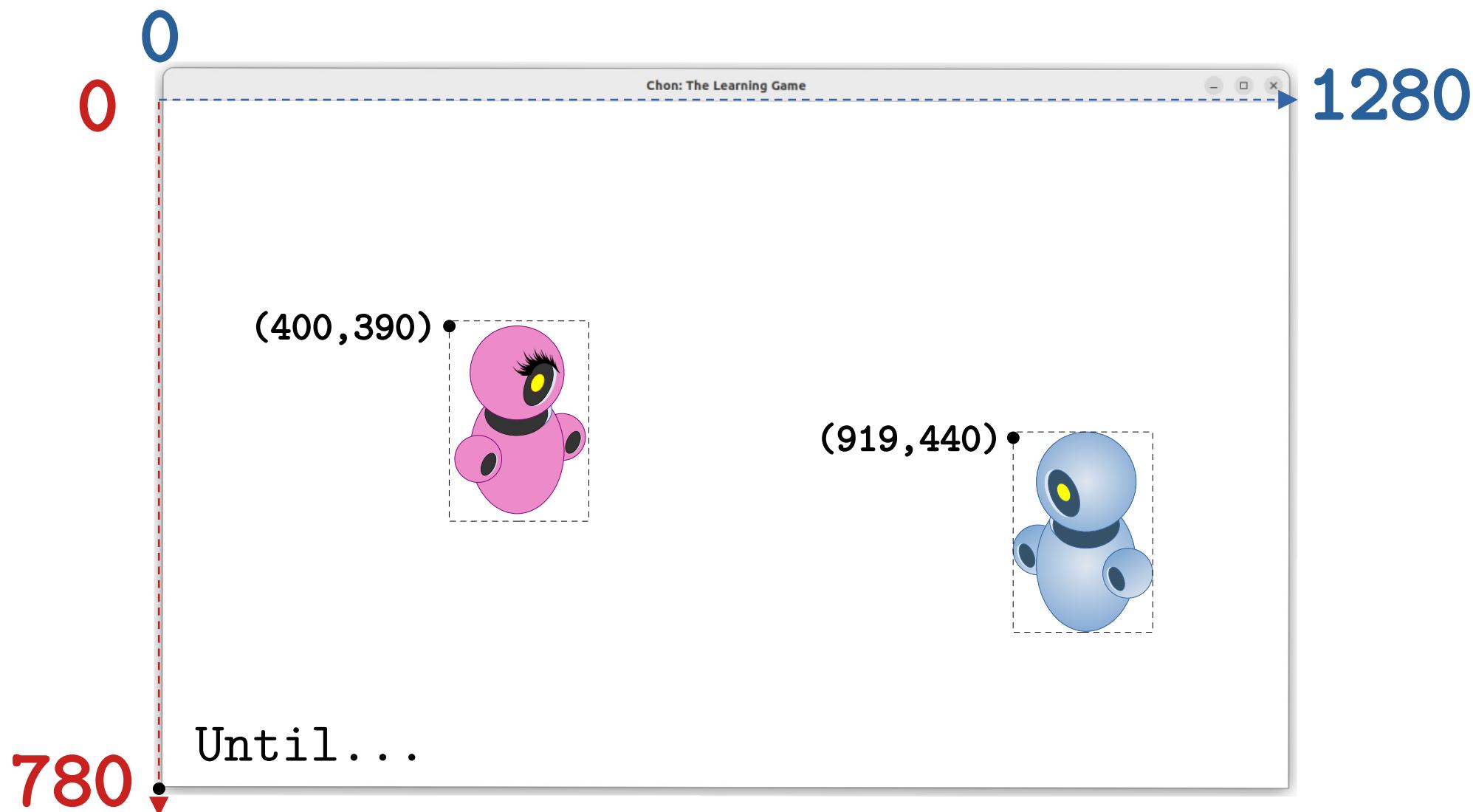
# Moving Another Object to the LEFT



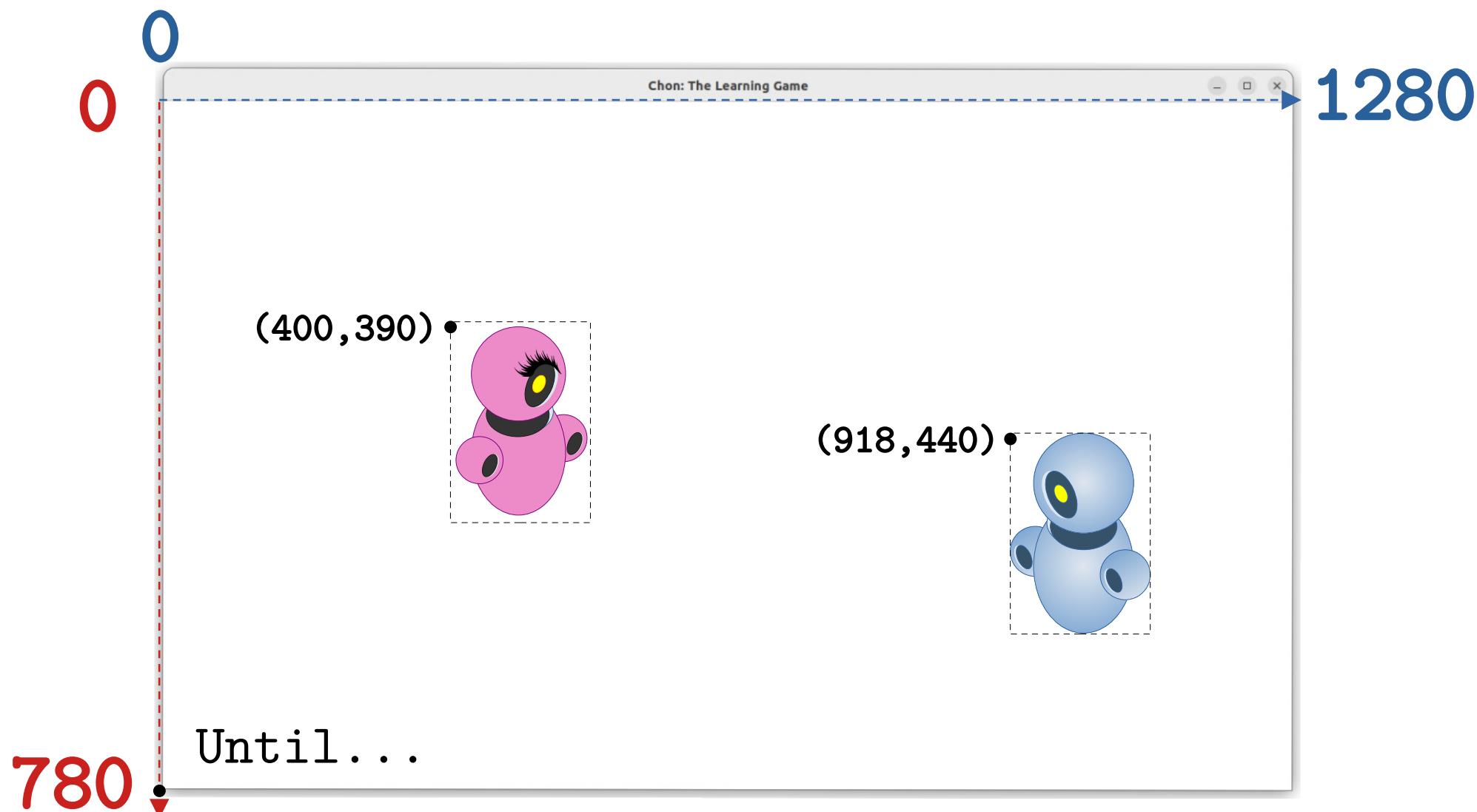
# Moving Another Object to the LEFT



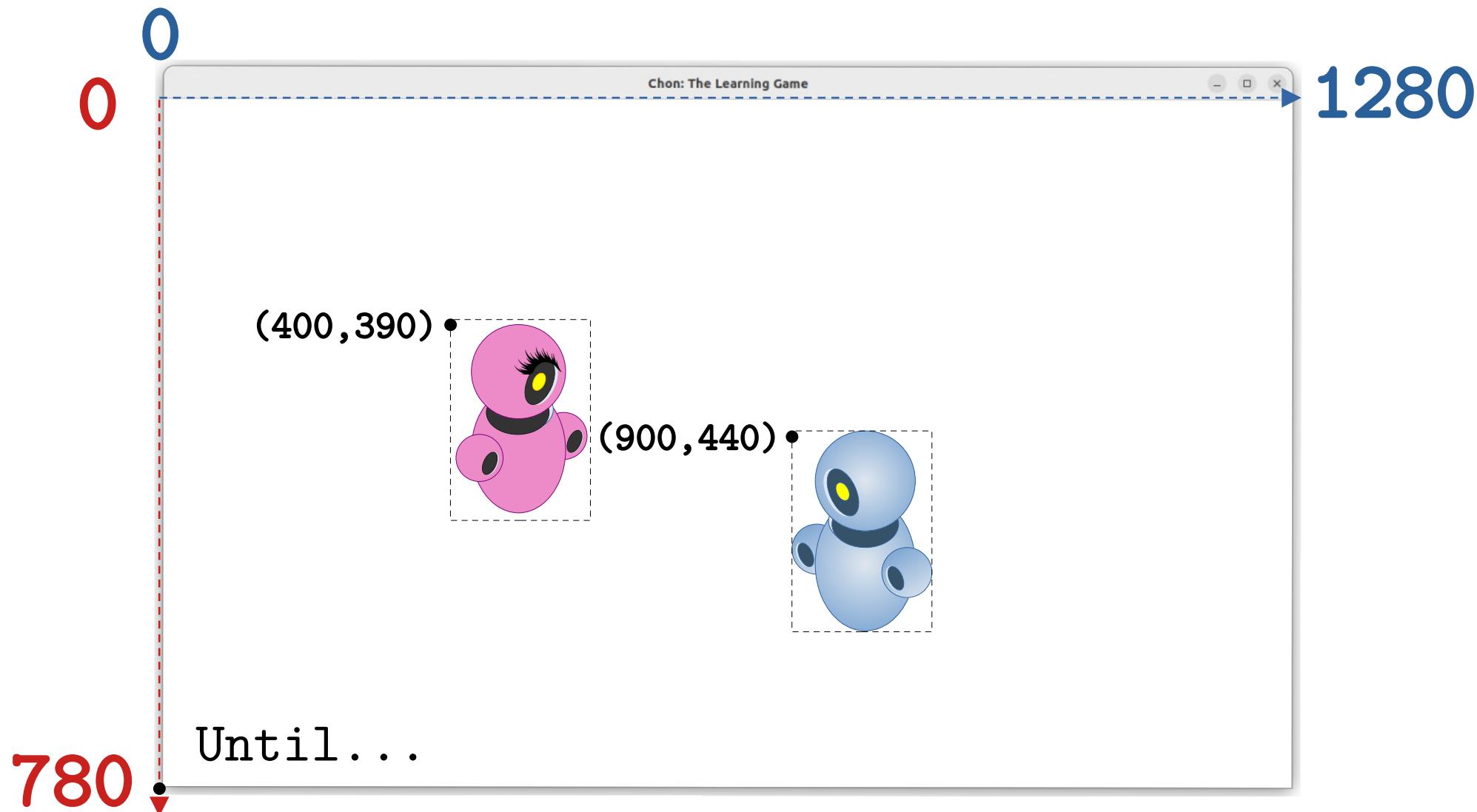
# Moving Another Object to the LEFT



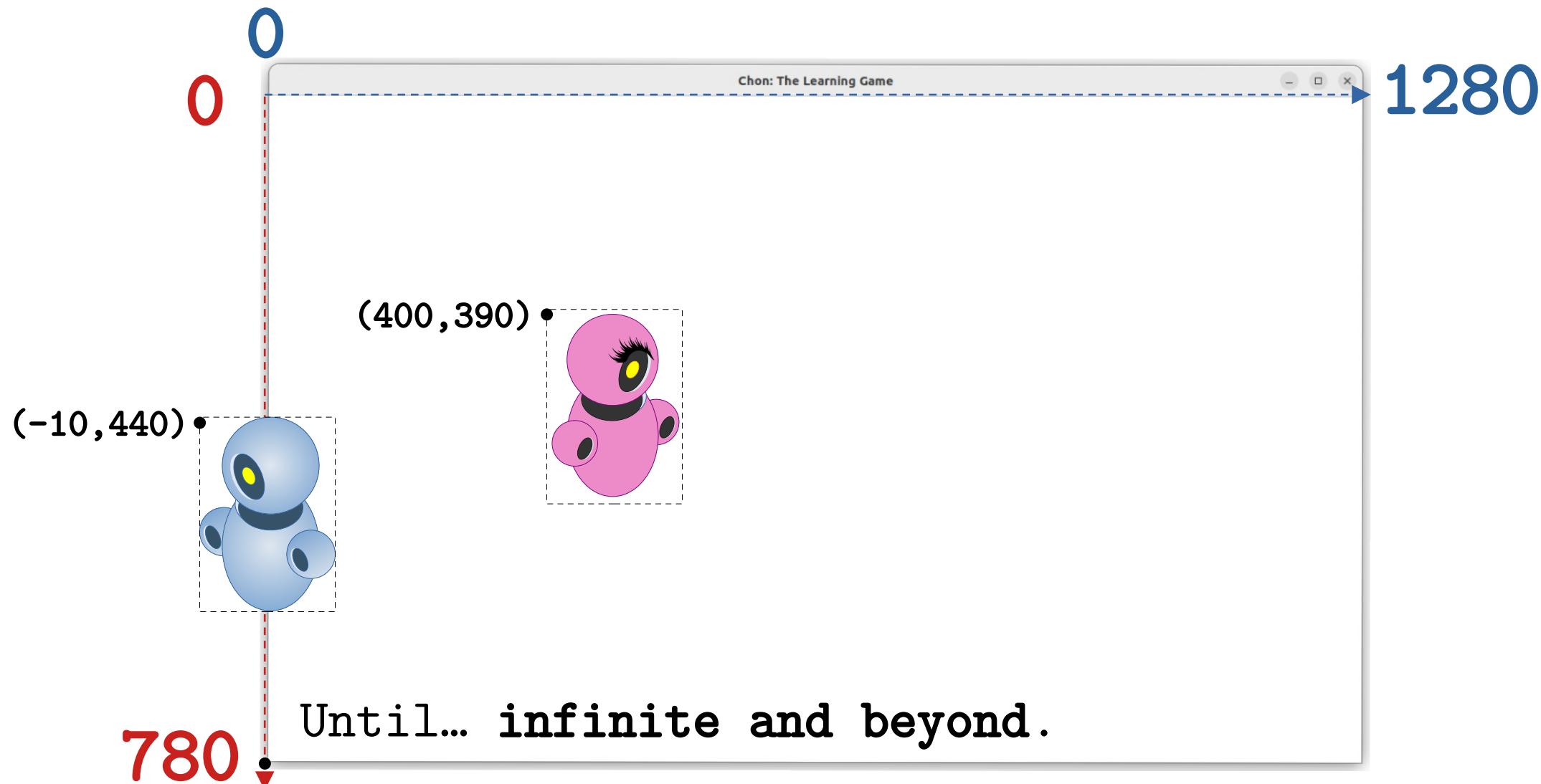
# Moving Another Object to the LEFT



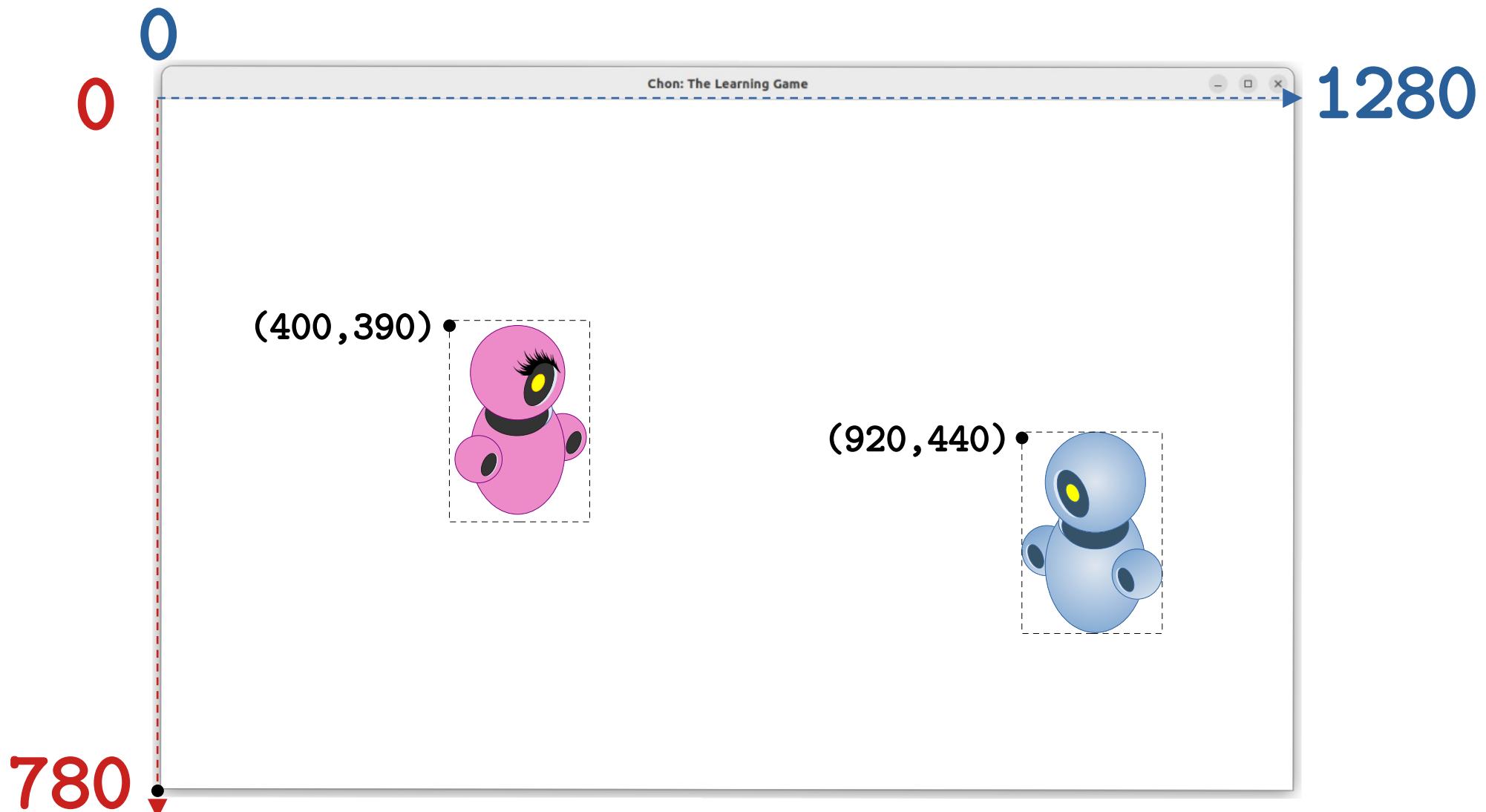
# Moving Another Object to the LEFT



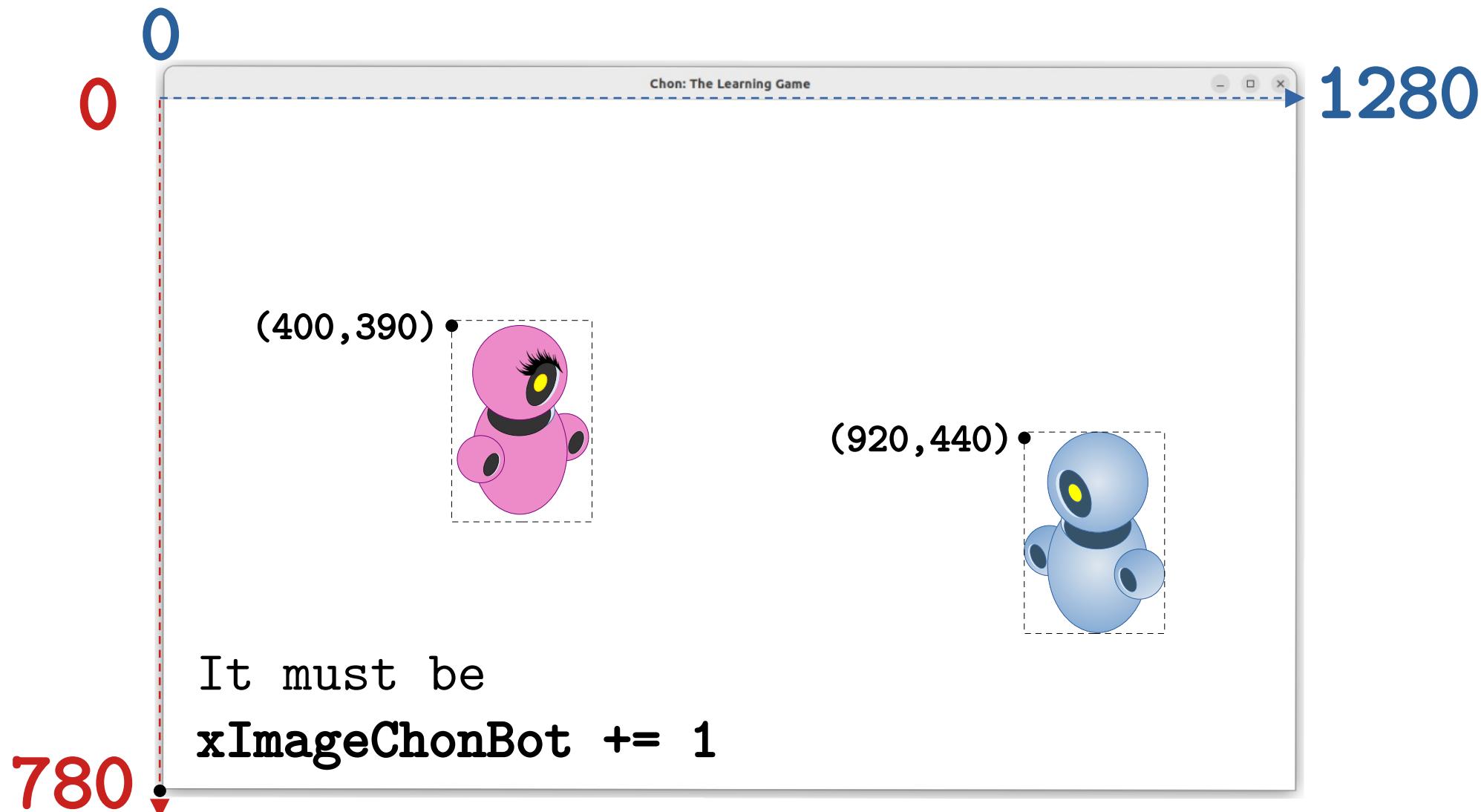
# Moving Another Object to the LEFT



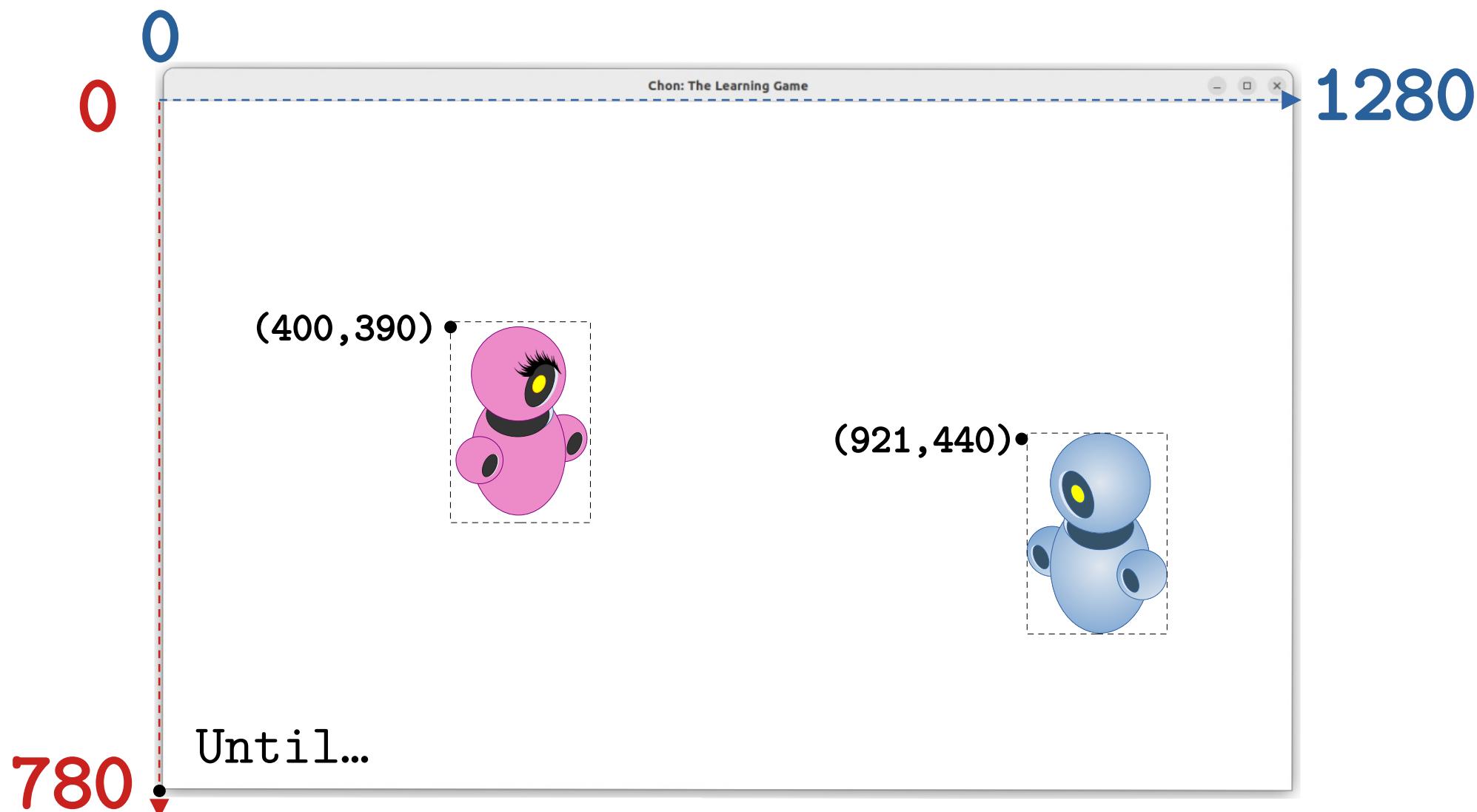
# Moving Another Object to the RIGHT



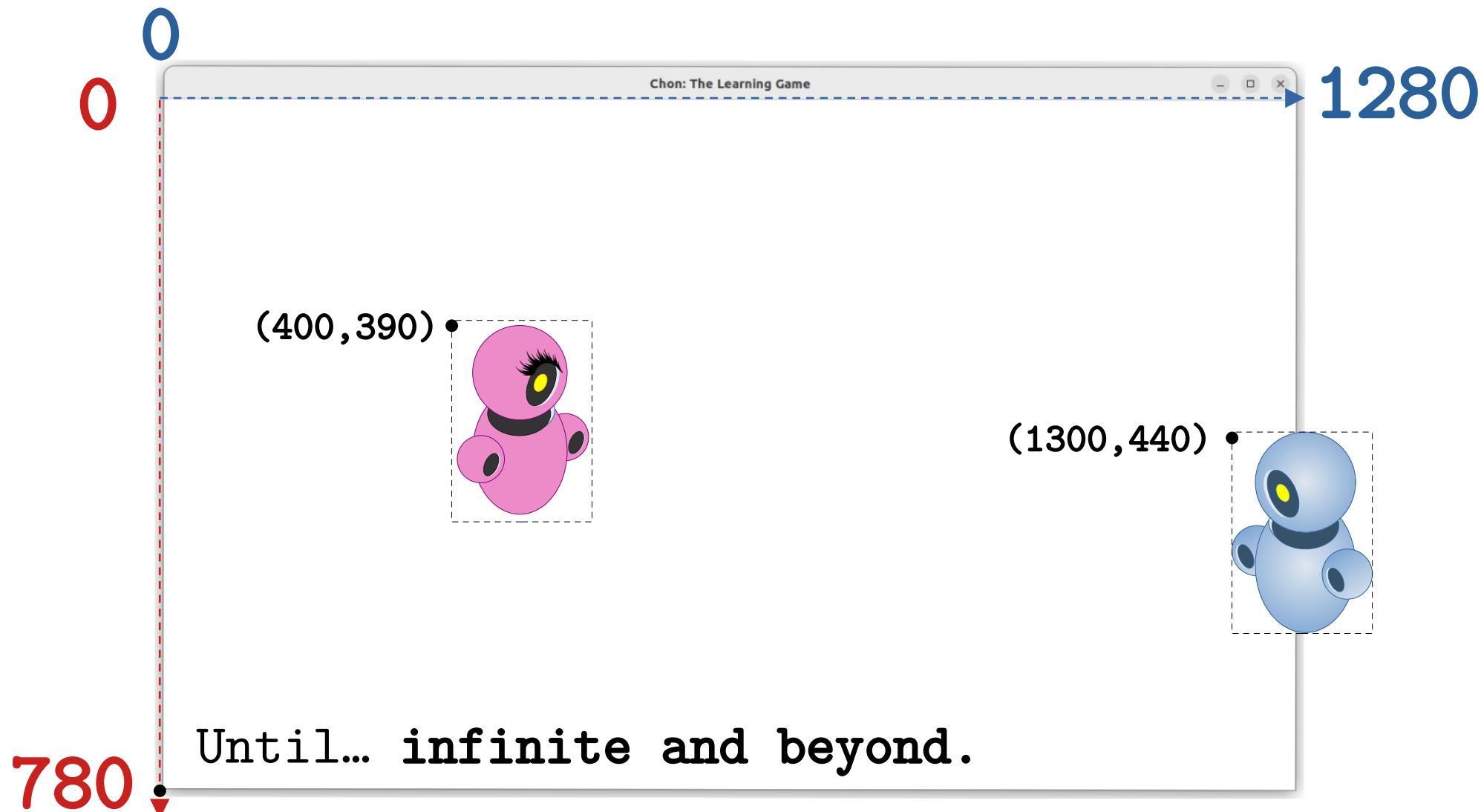
# Moving Another Object to the RIGHT



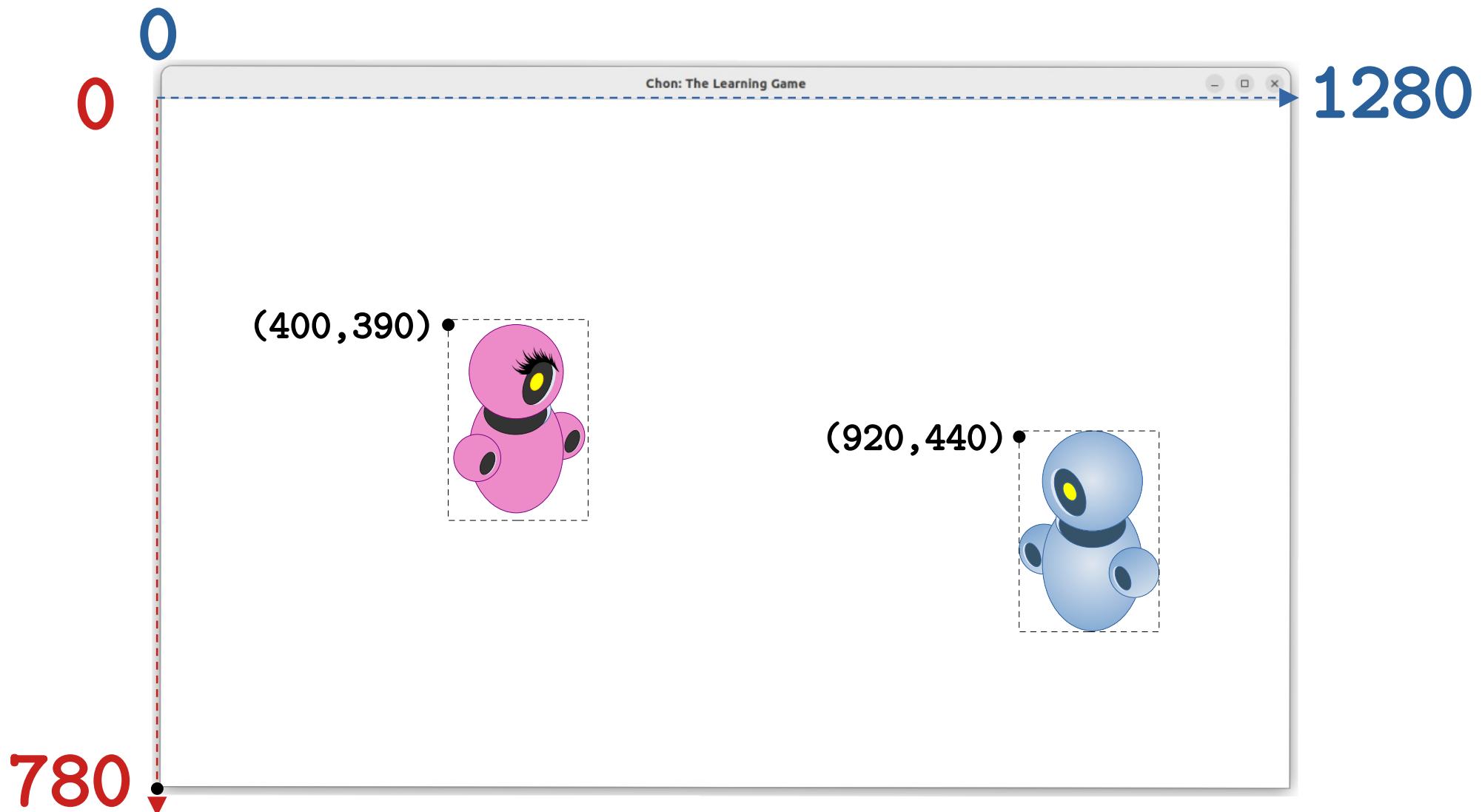
# Moving Another Object to the RIGHT



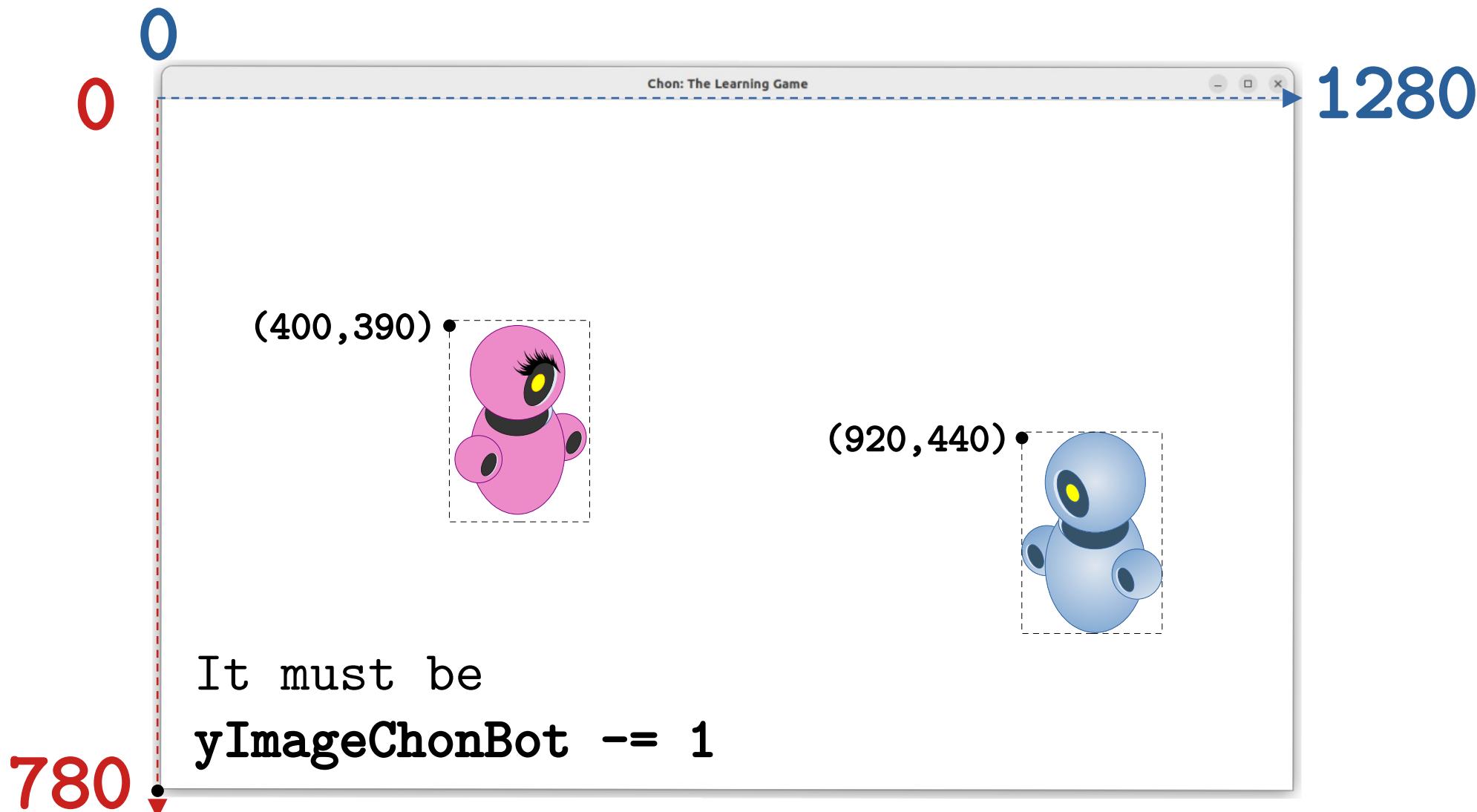
# Moving Another Object to the RIGHT



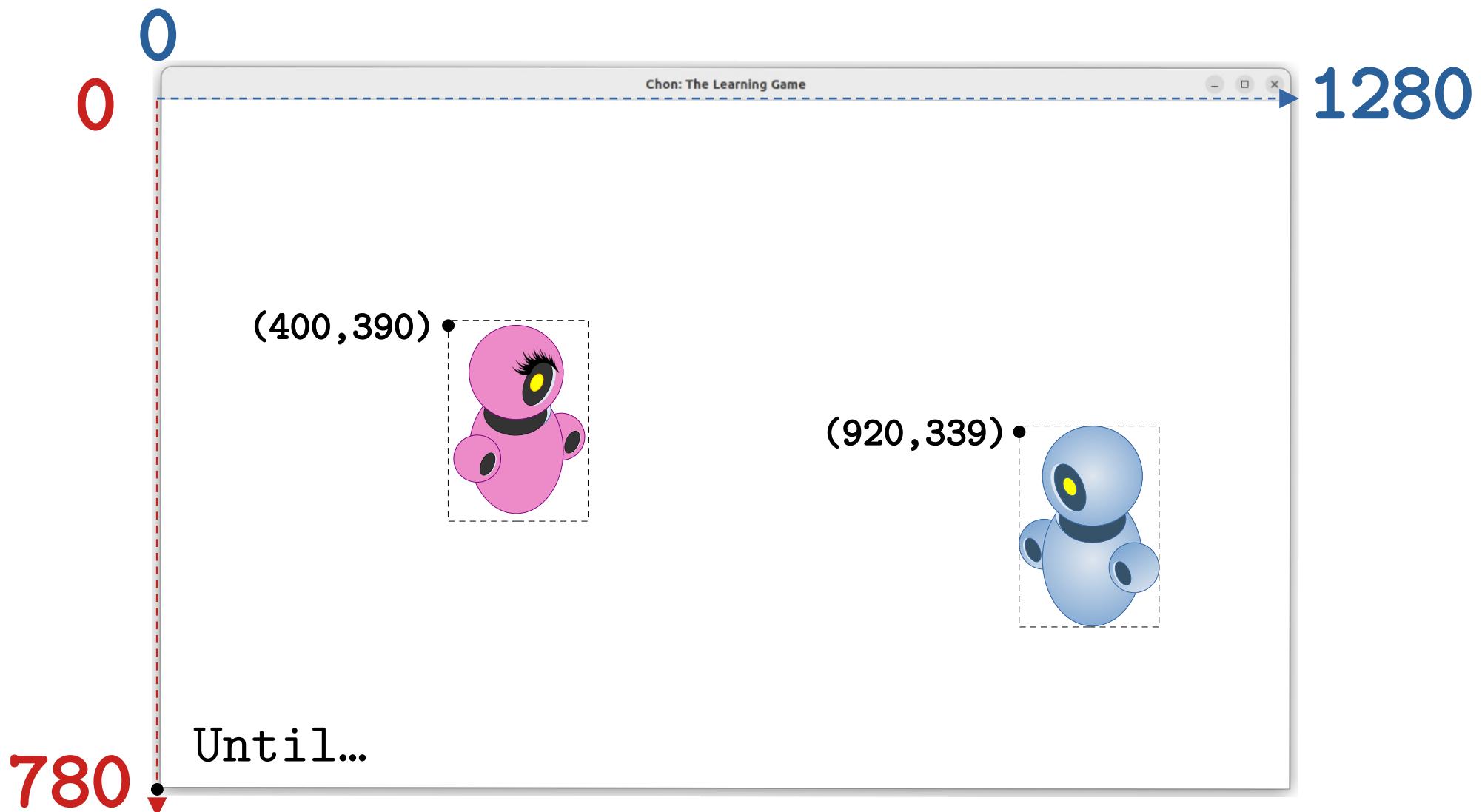
# Moving UP Another Object



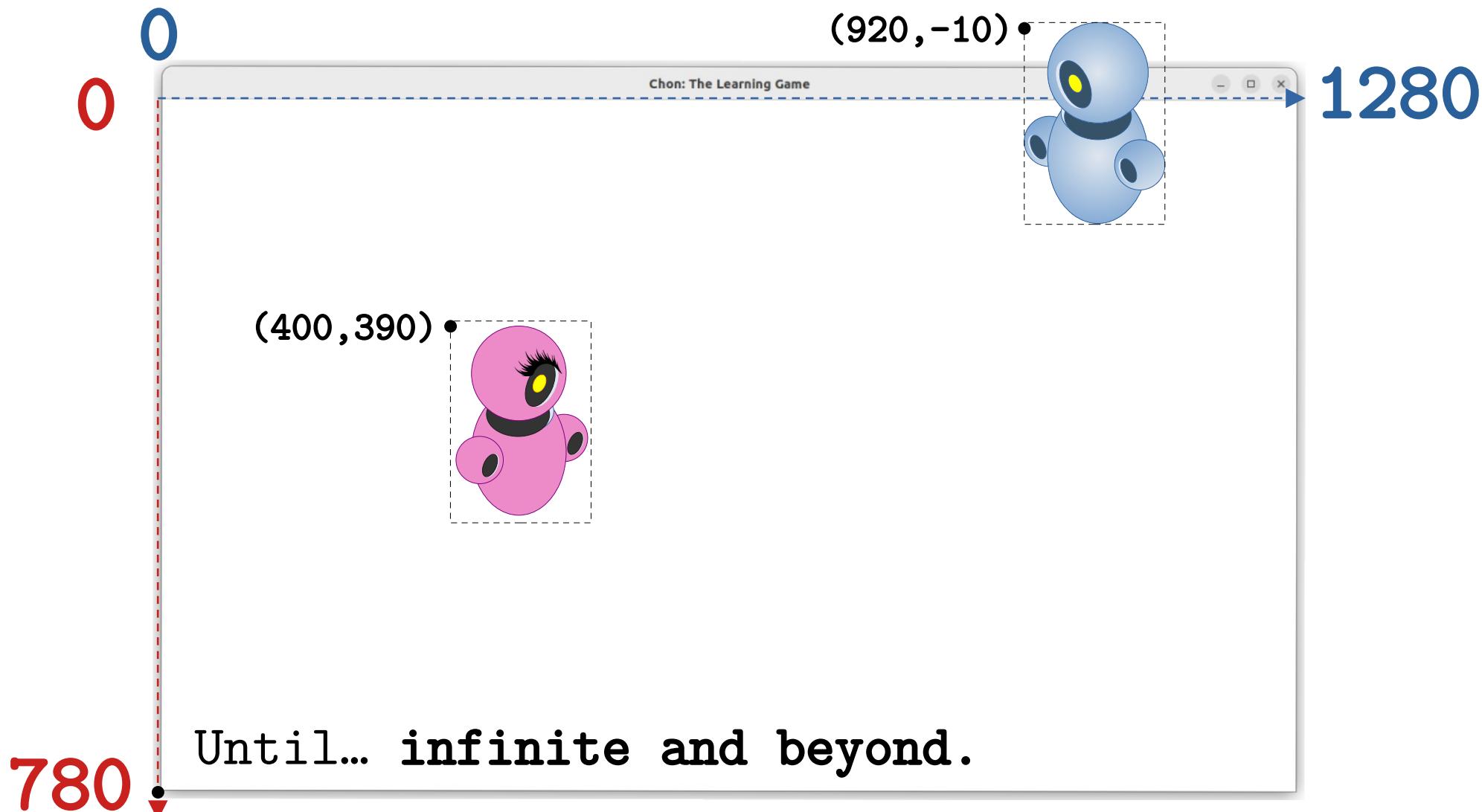
# Moving UP Another Object



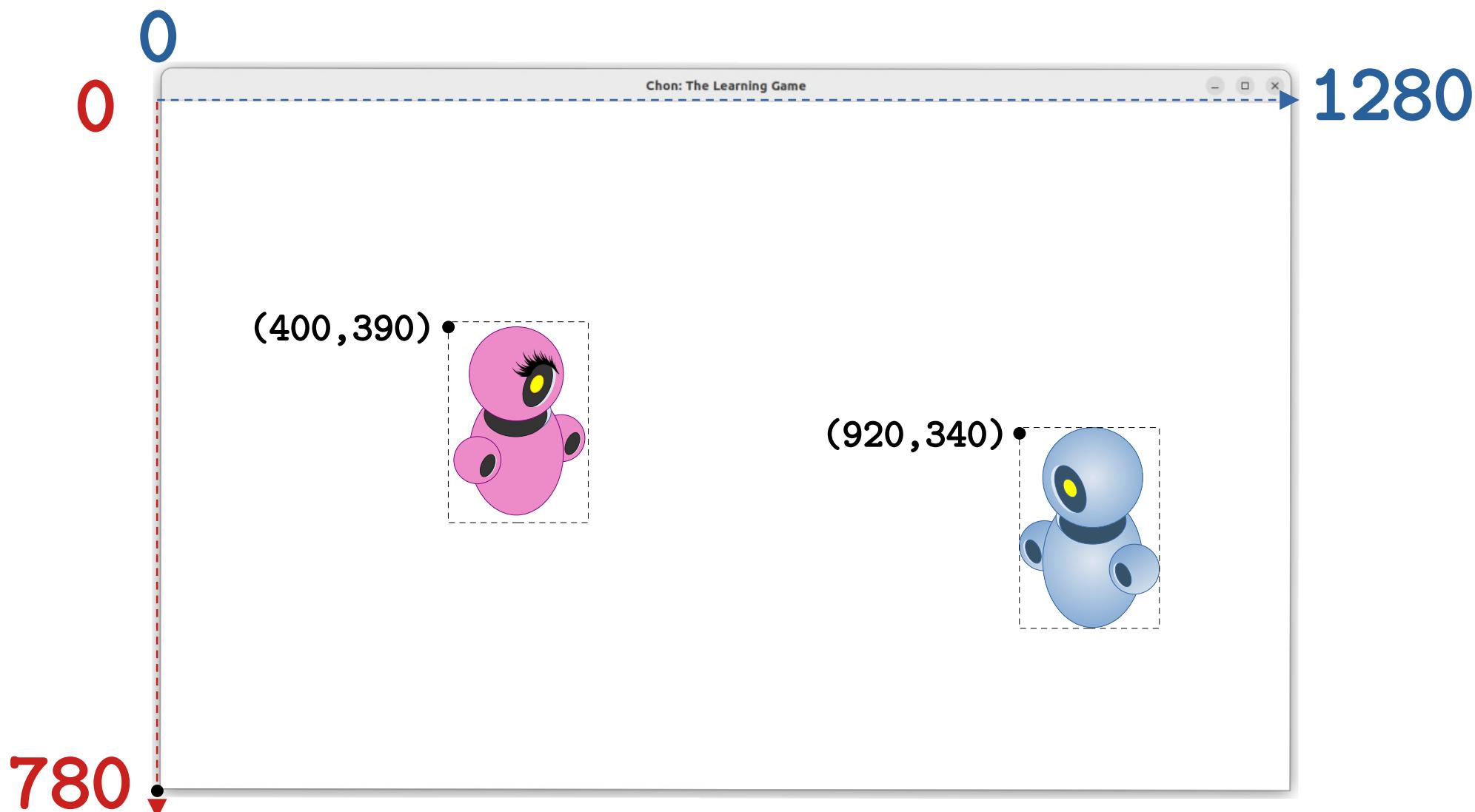
# Moving UP Another Object



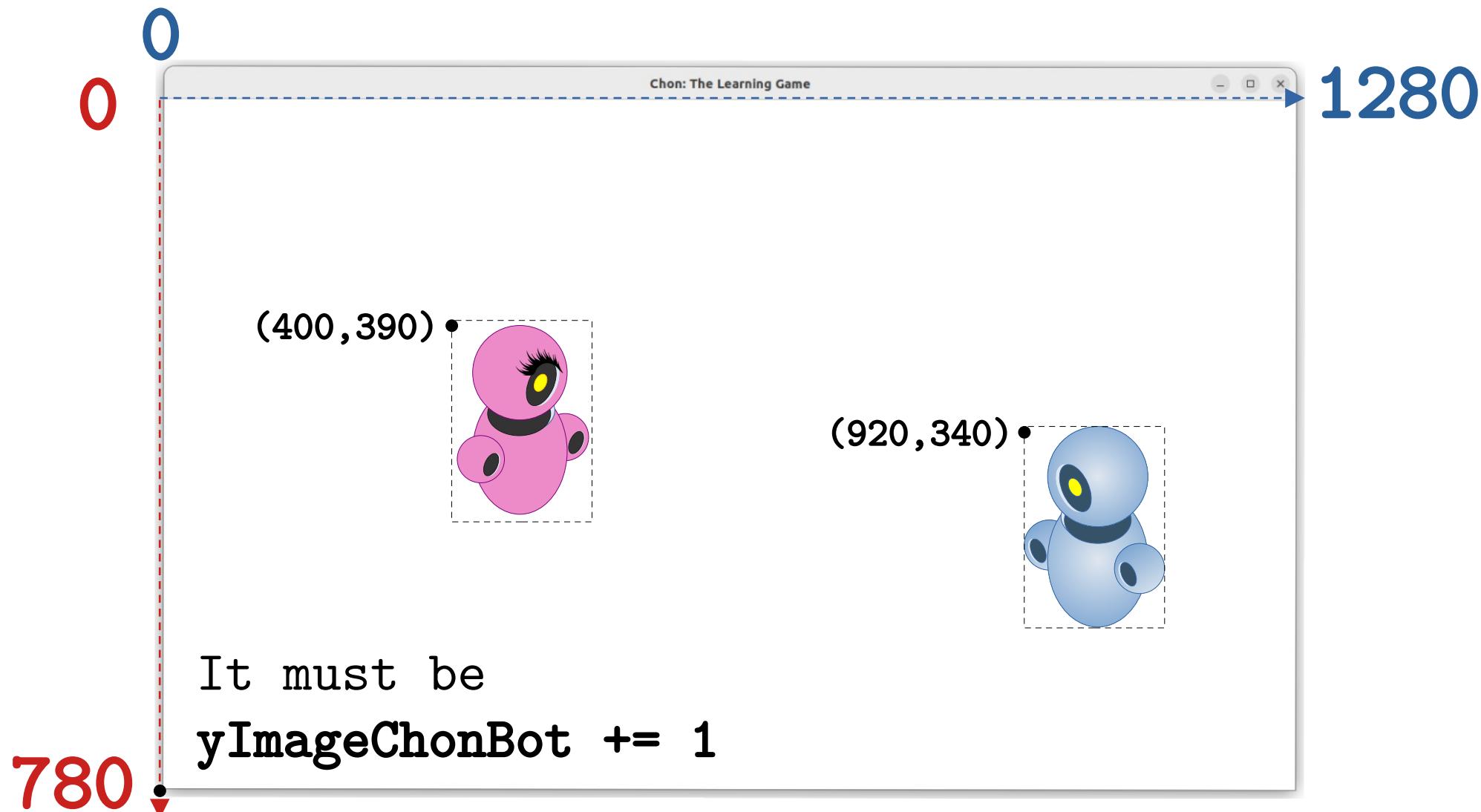
# Moving UP Another Object



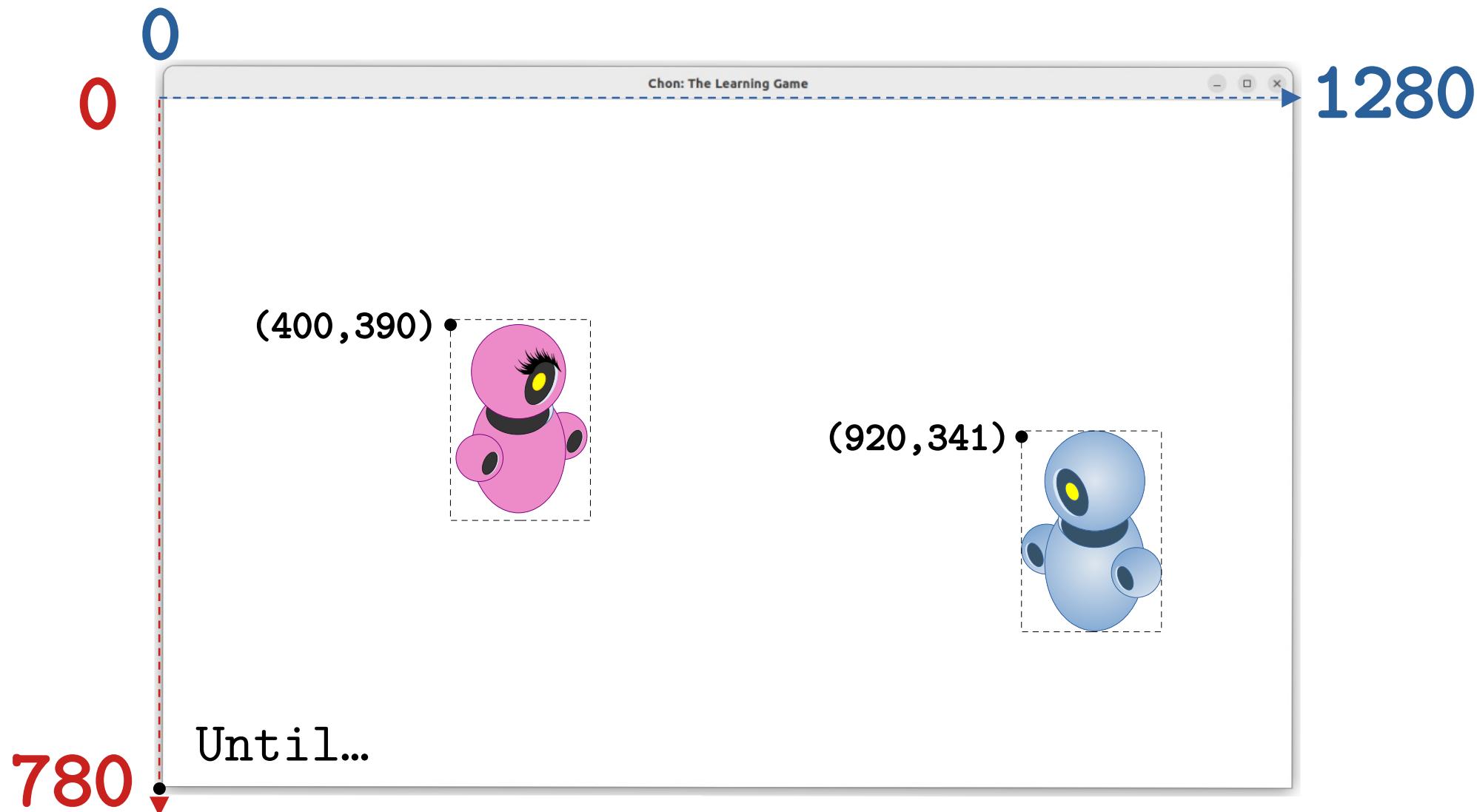
# Moving DOWN Another Object



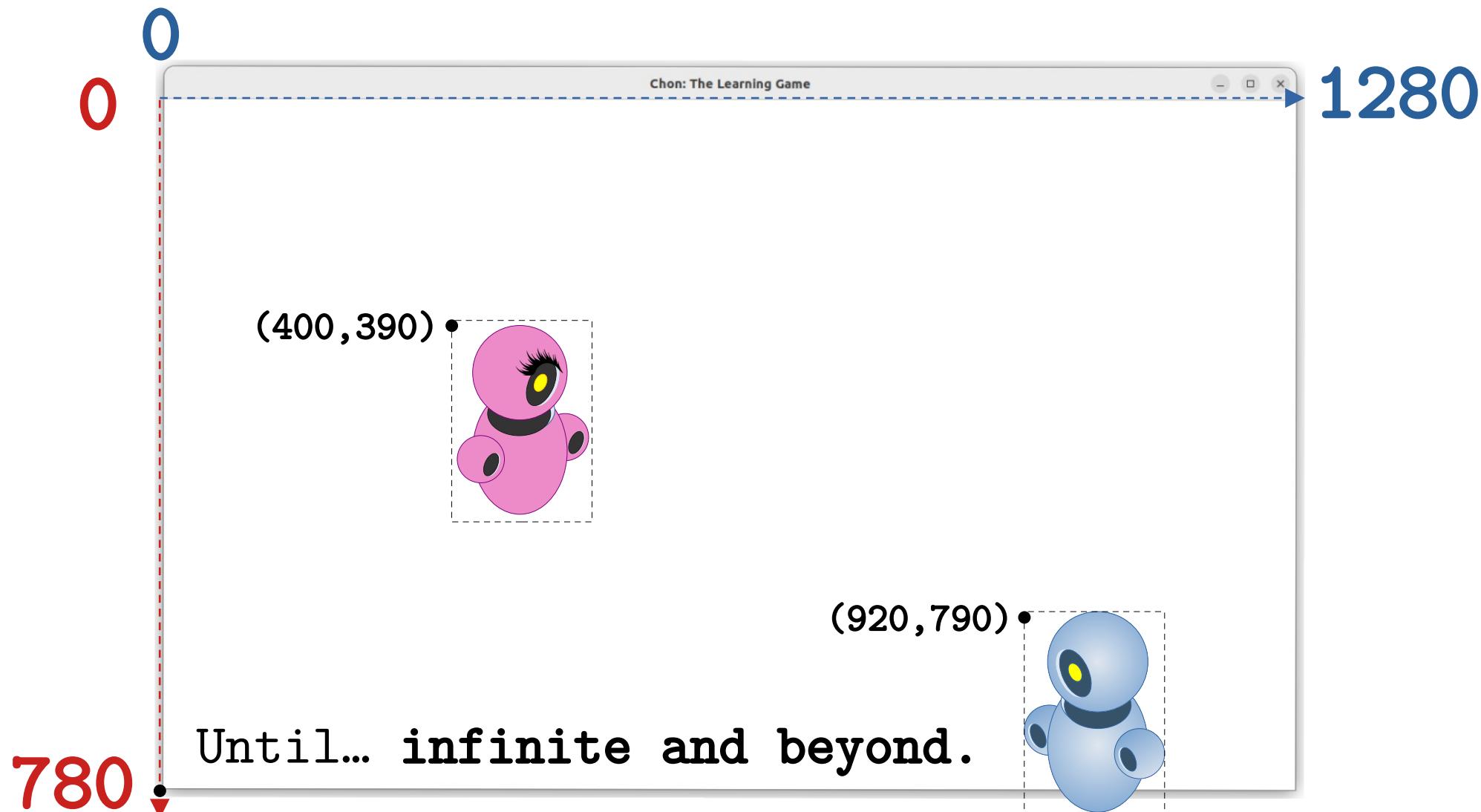
# Moving DOWN Another Object



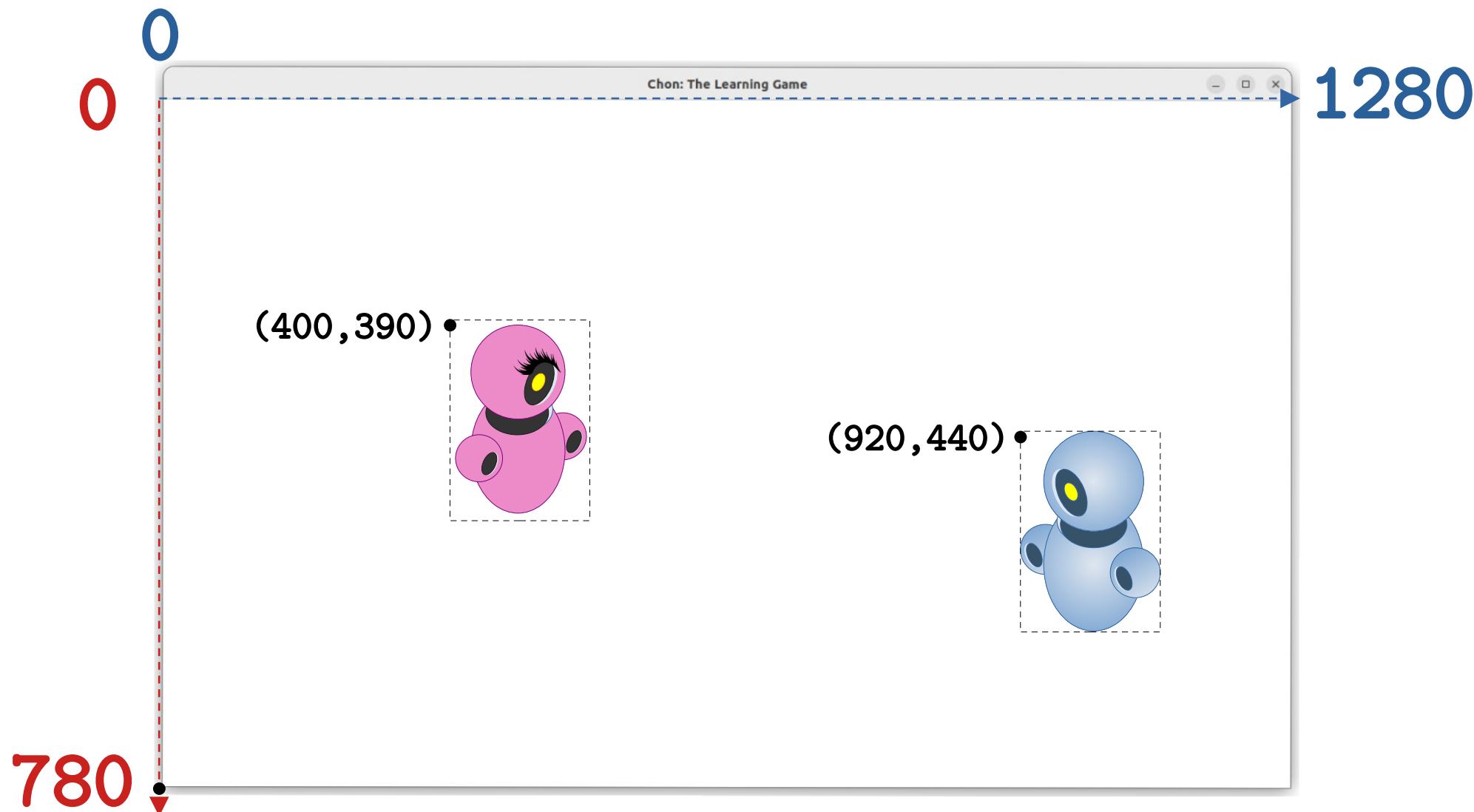
# Moving DOWN Another Object



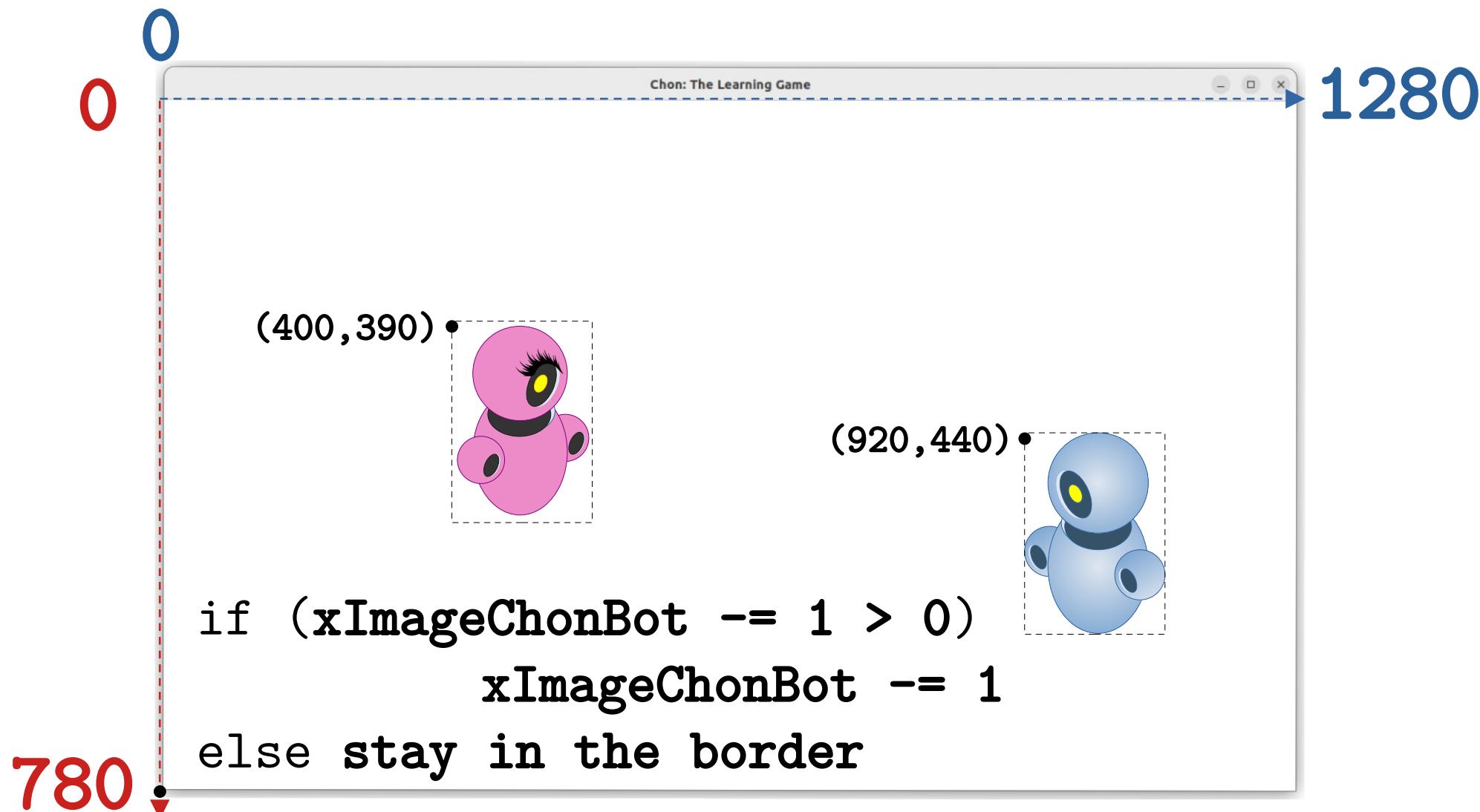
# Moving DOWN Another Object



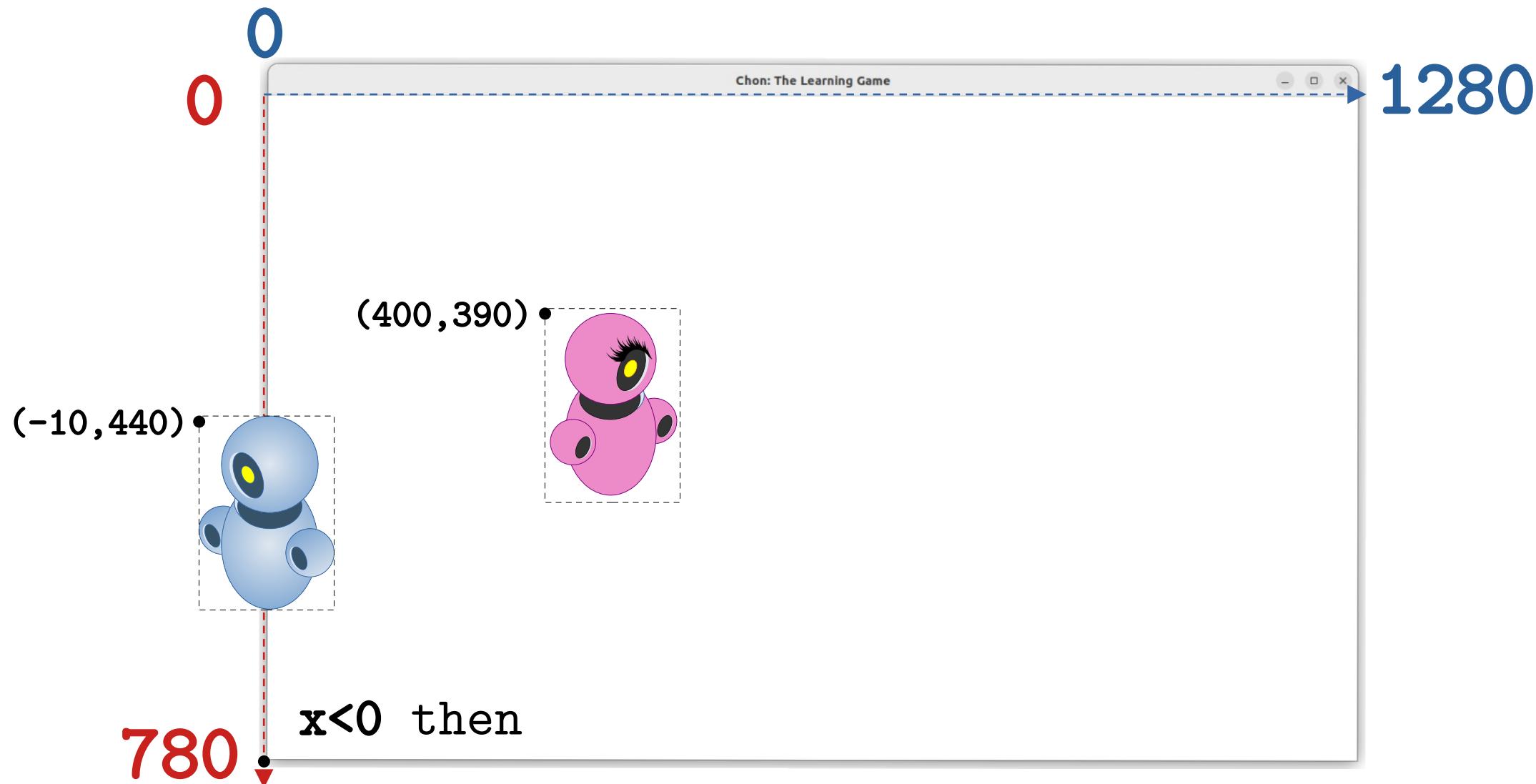
# Moving Until the Border



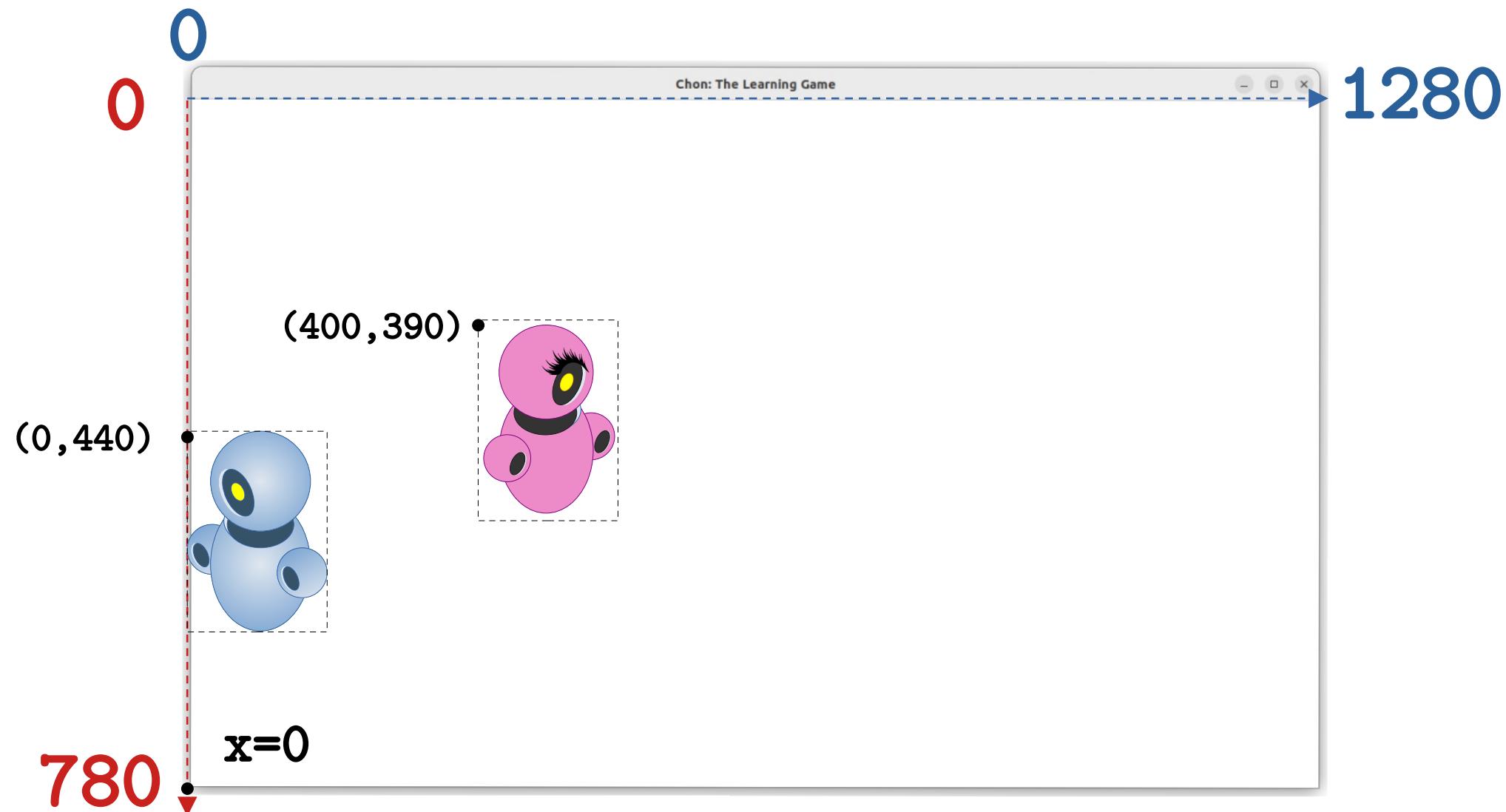
# Moving Until the Border



# Moving Until the Border



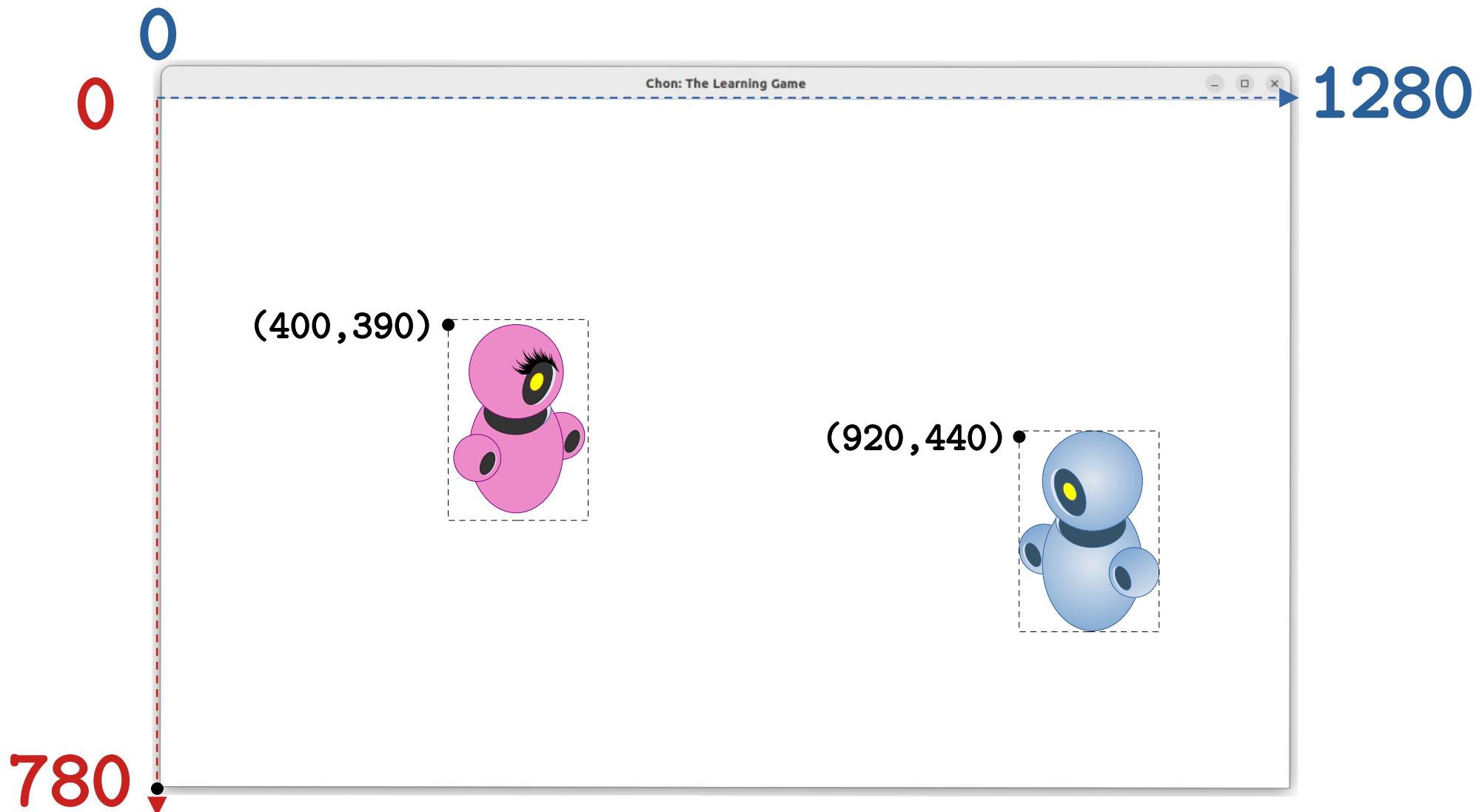
# Moving Until the Border



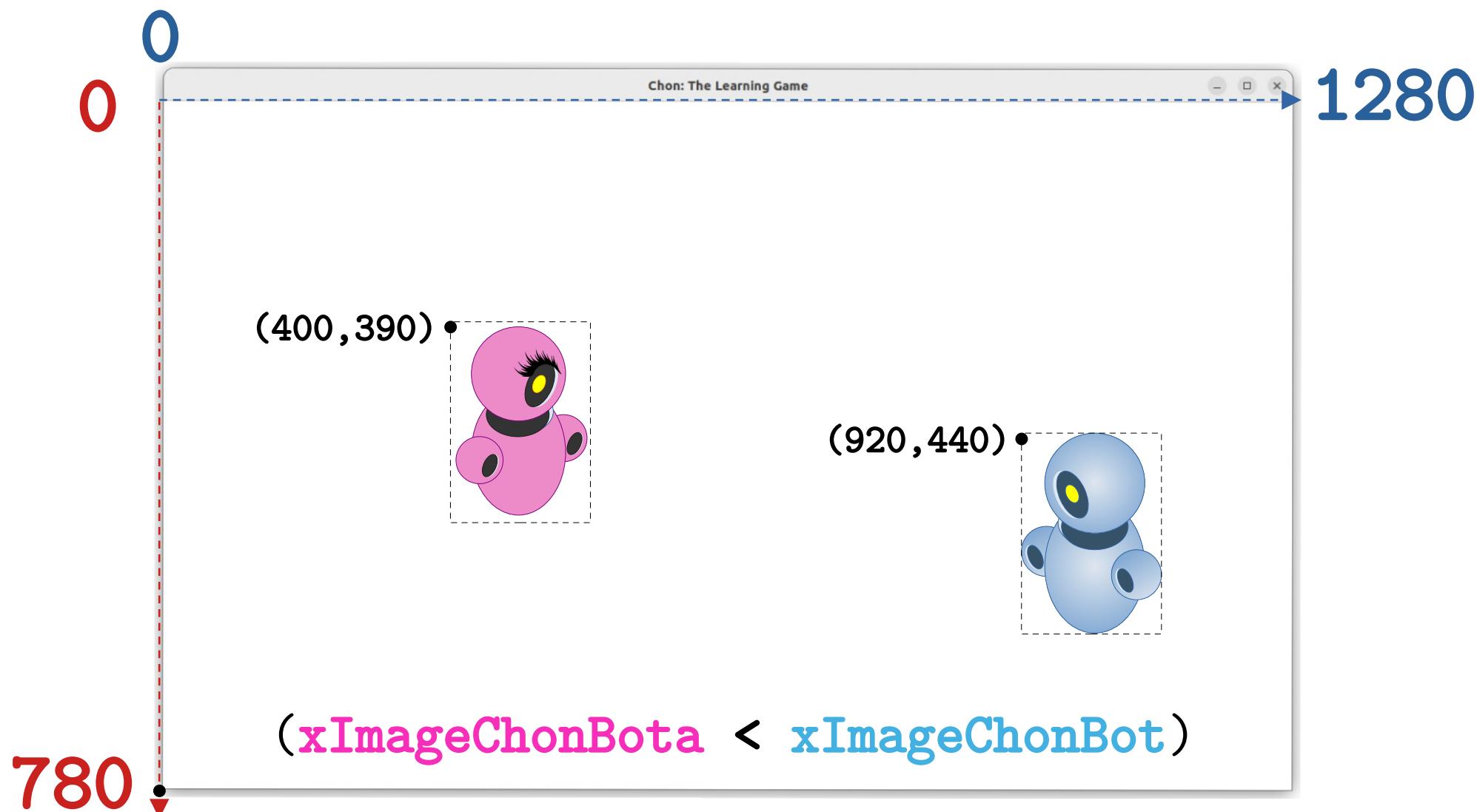
# Moving Until the Border

The same mechanics  
applies to the  
**RIGHT, UP and DOWN.**

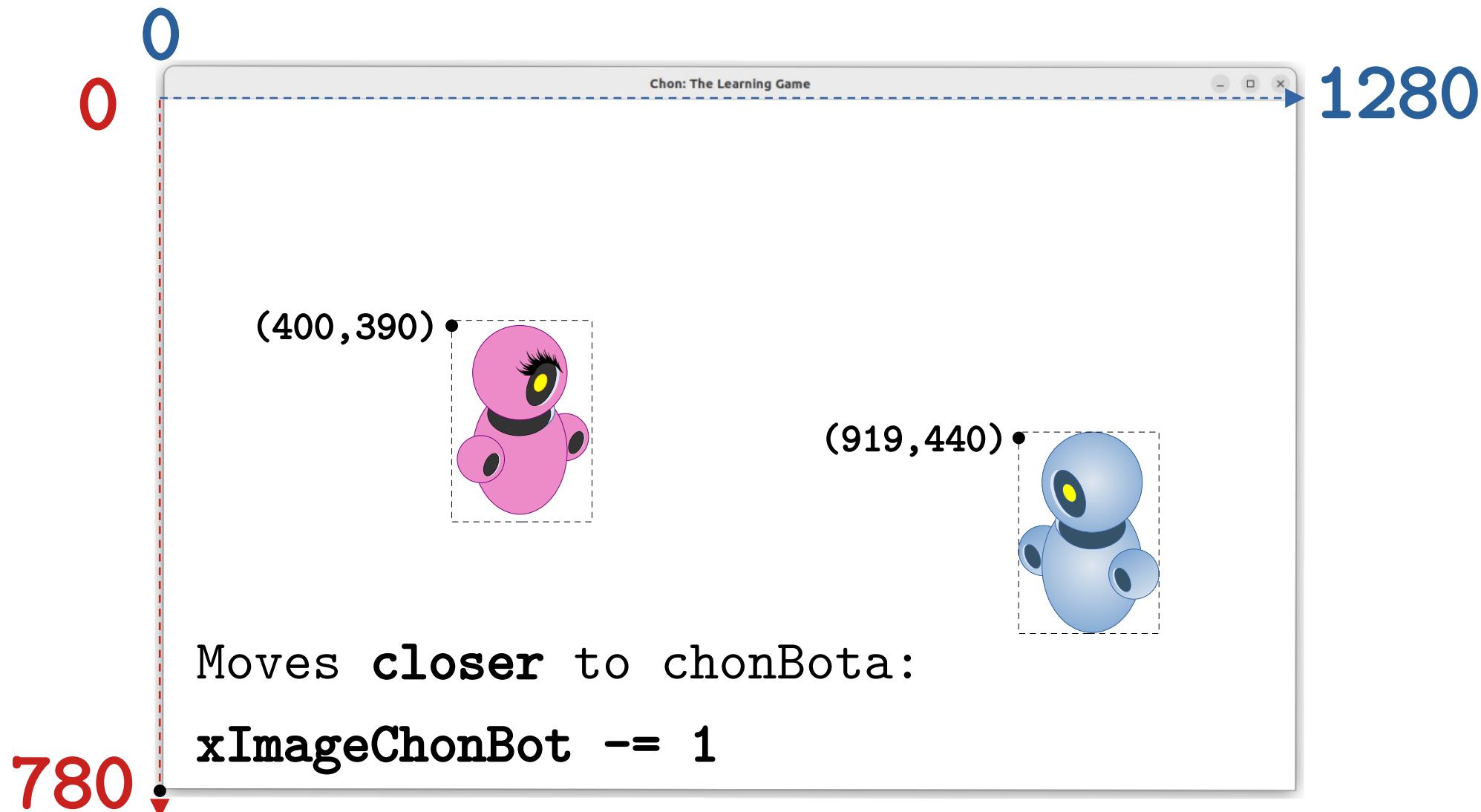
# Chasing the Player



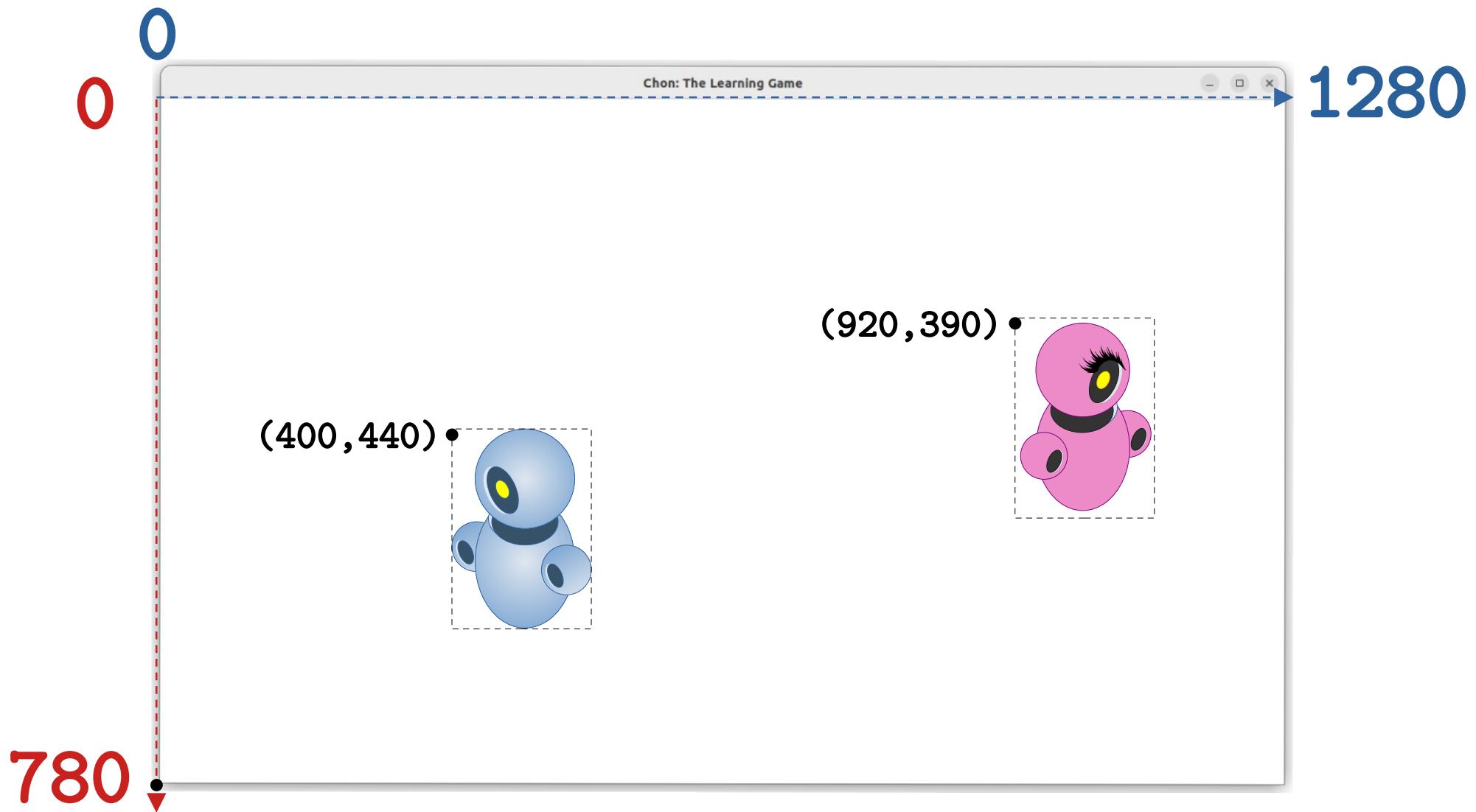
# Chasing the Player



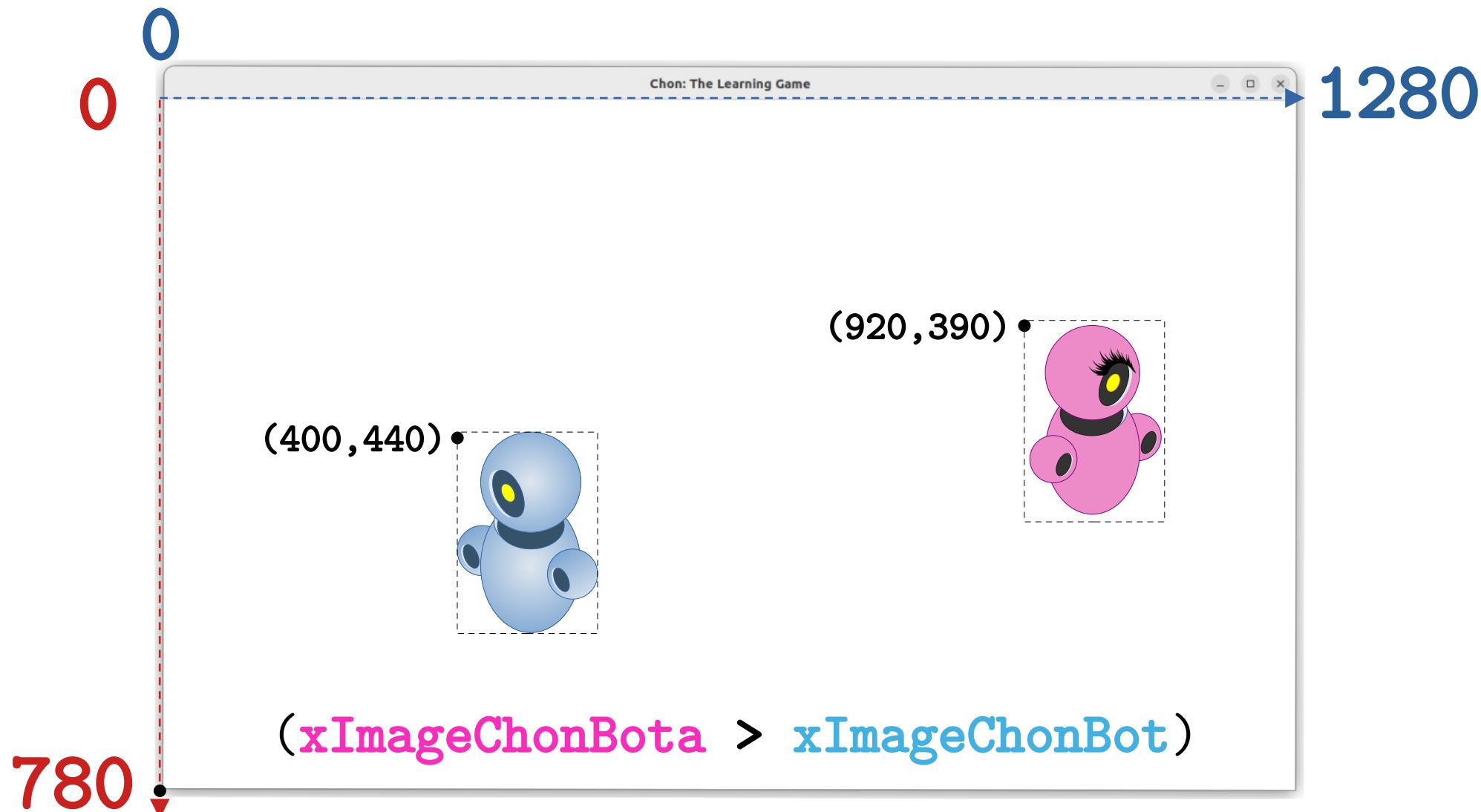
# Chasing the Player



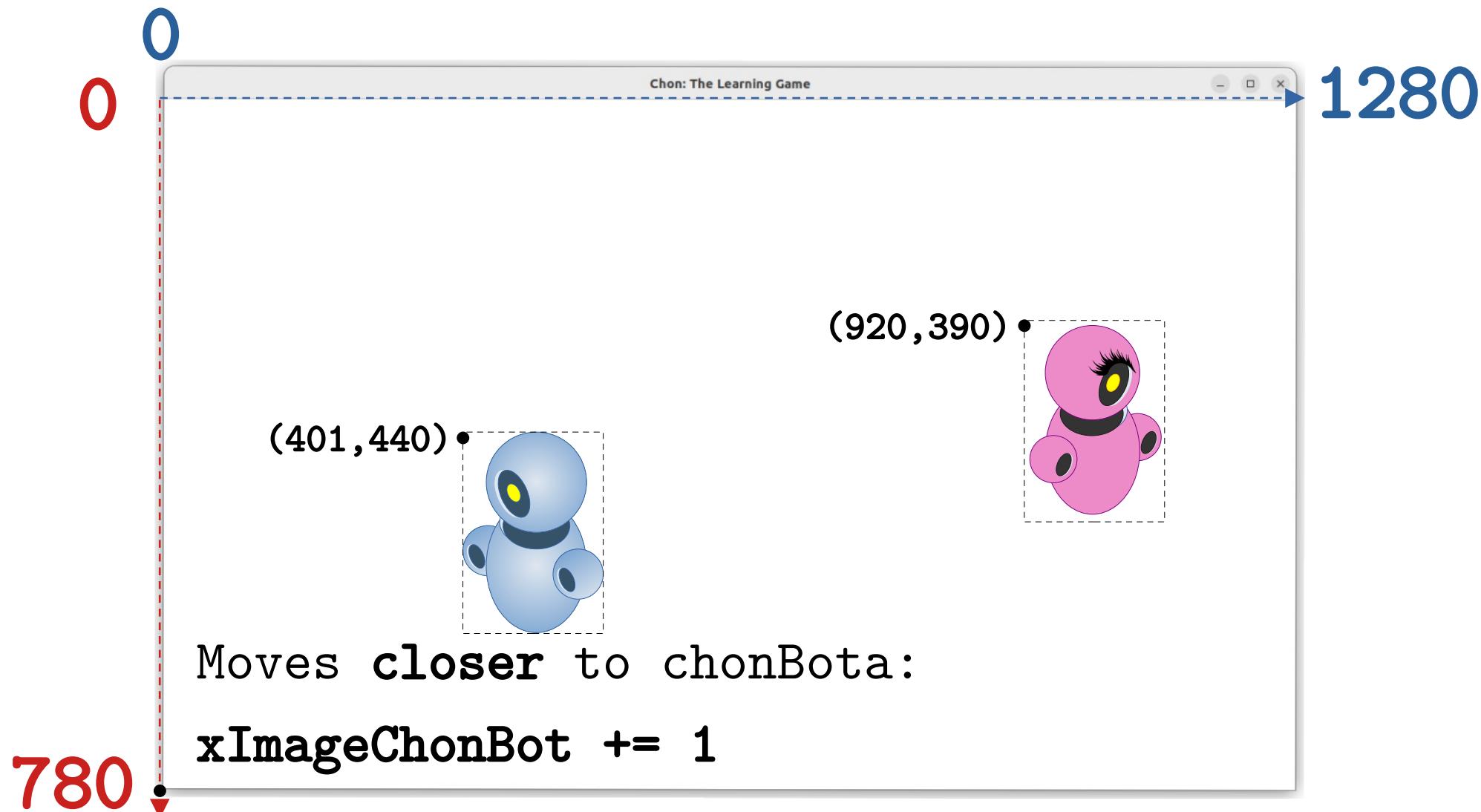
# Chasing the Player



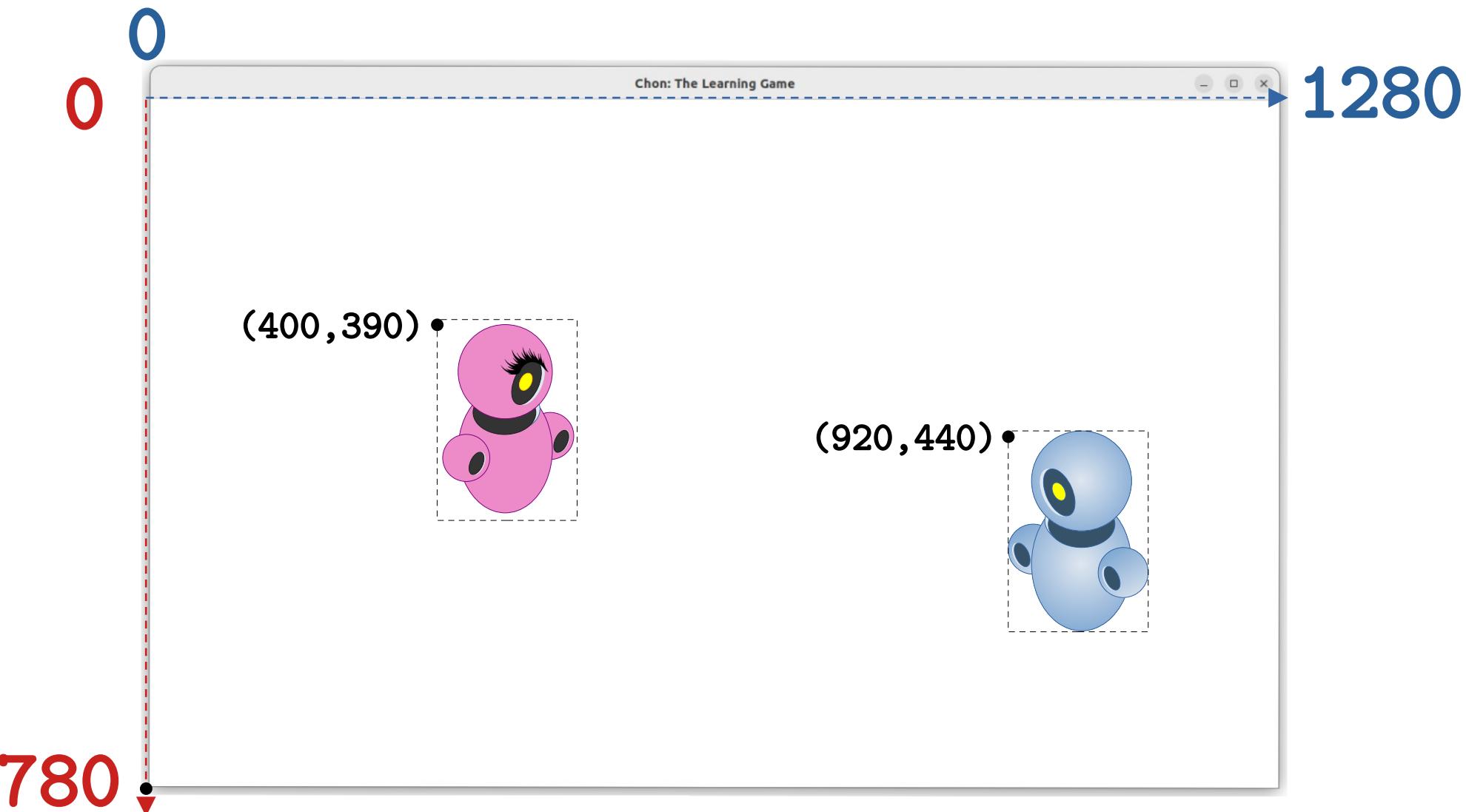
# Chasing the Player



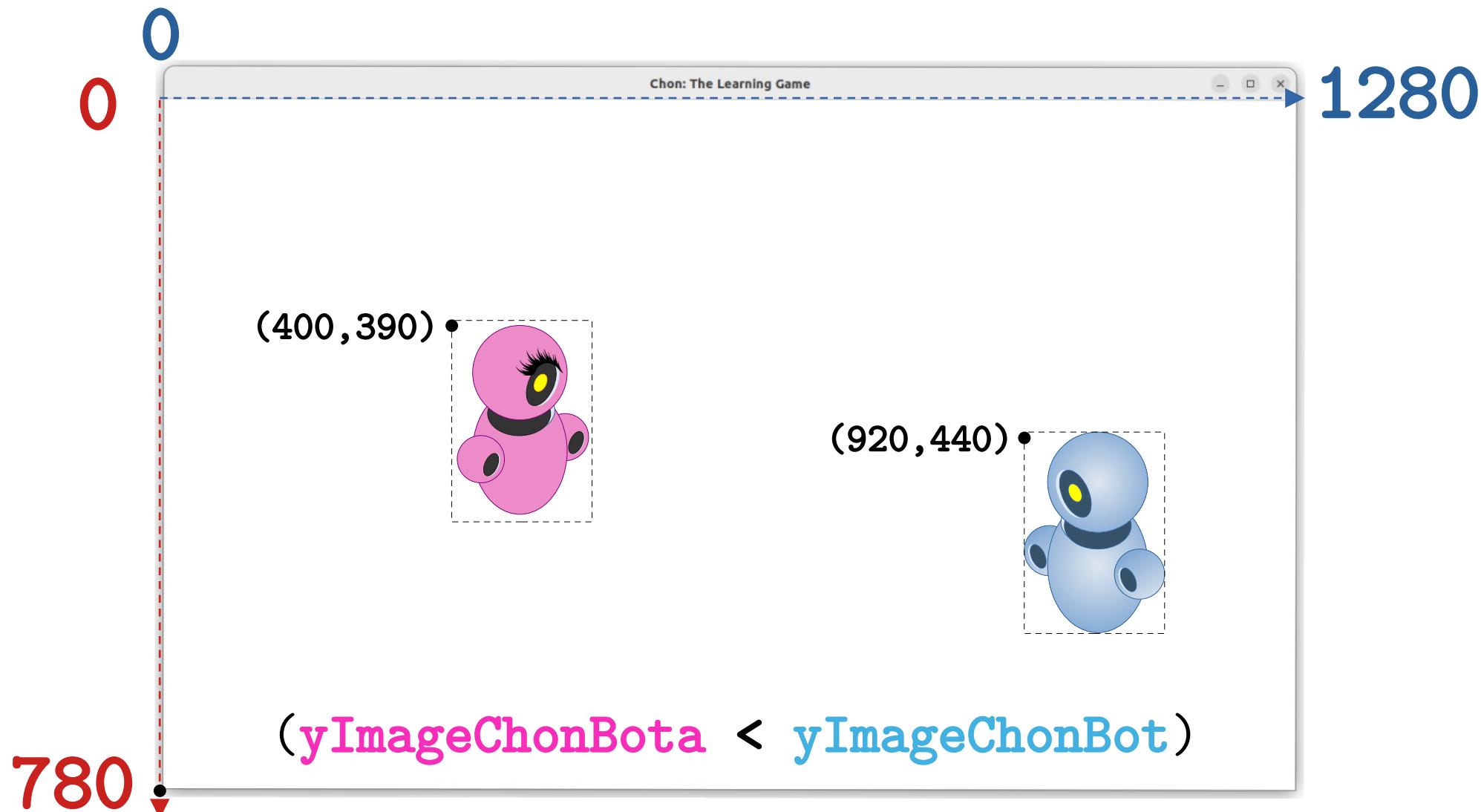
# Chasing the Player



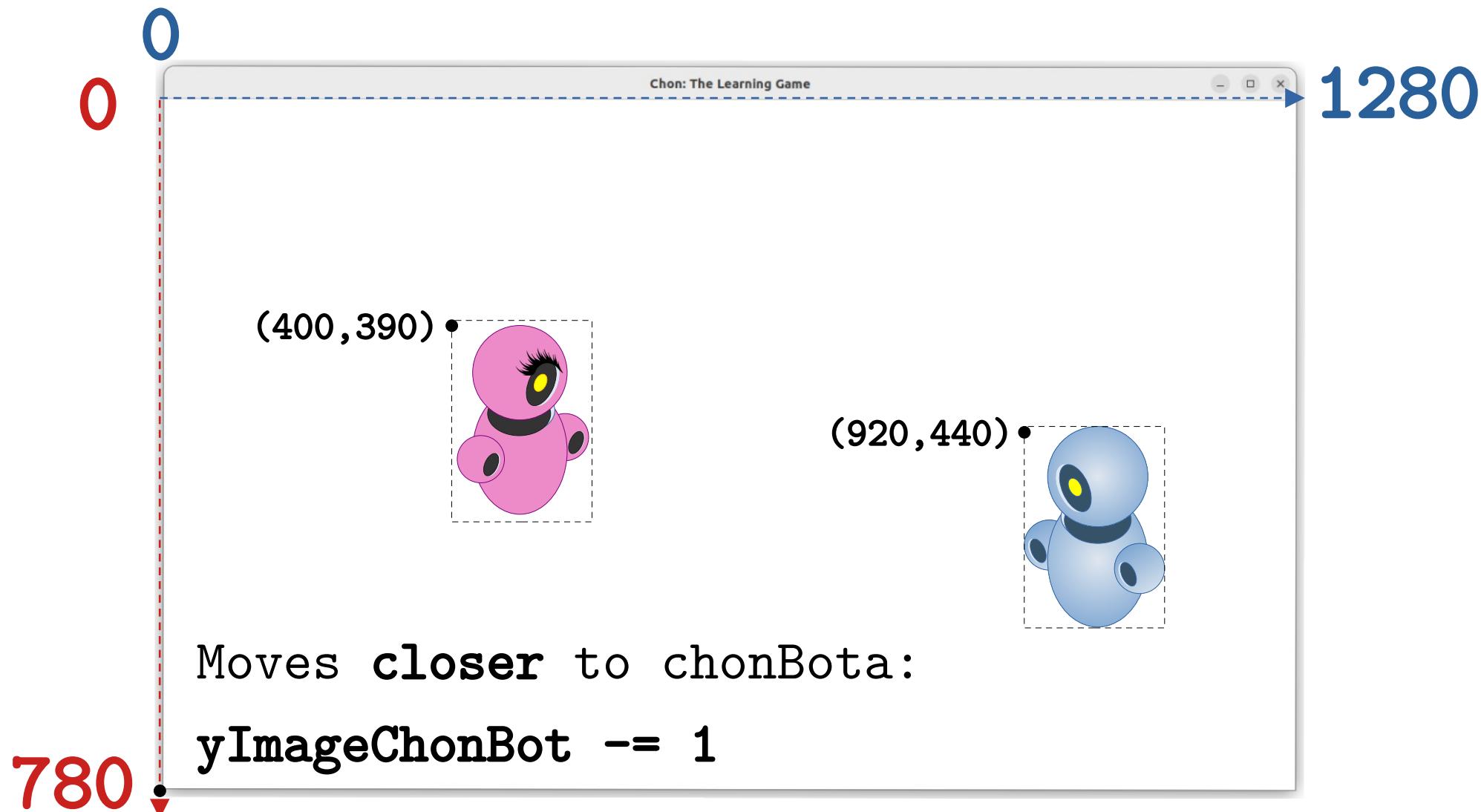
# Chasing the Player



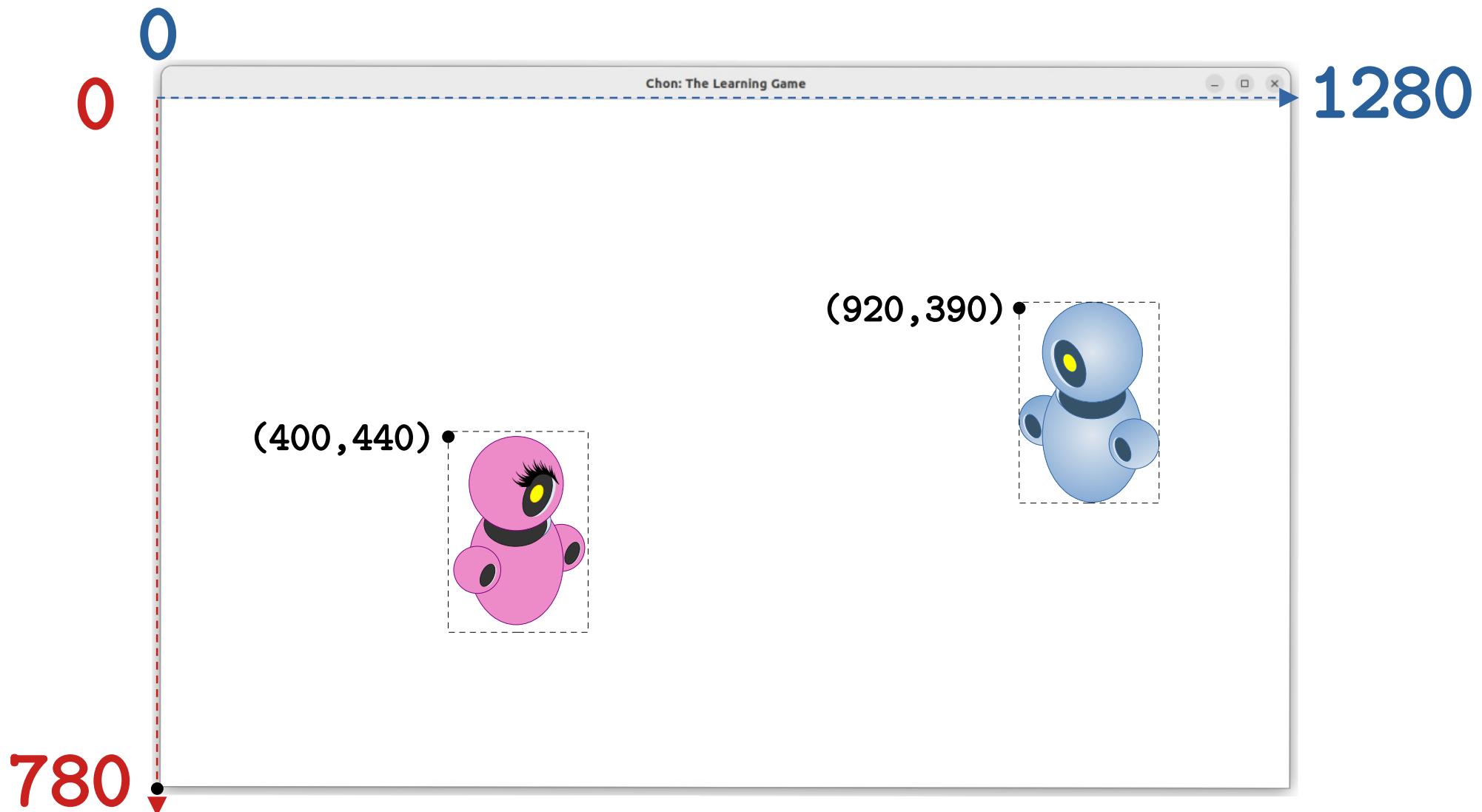
# Chasing the Player



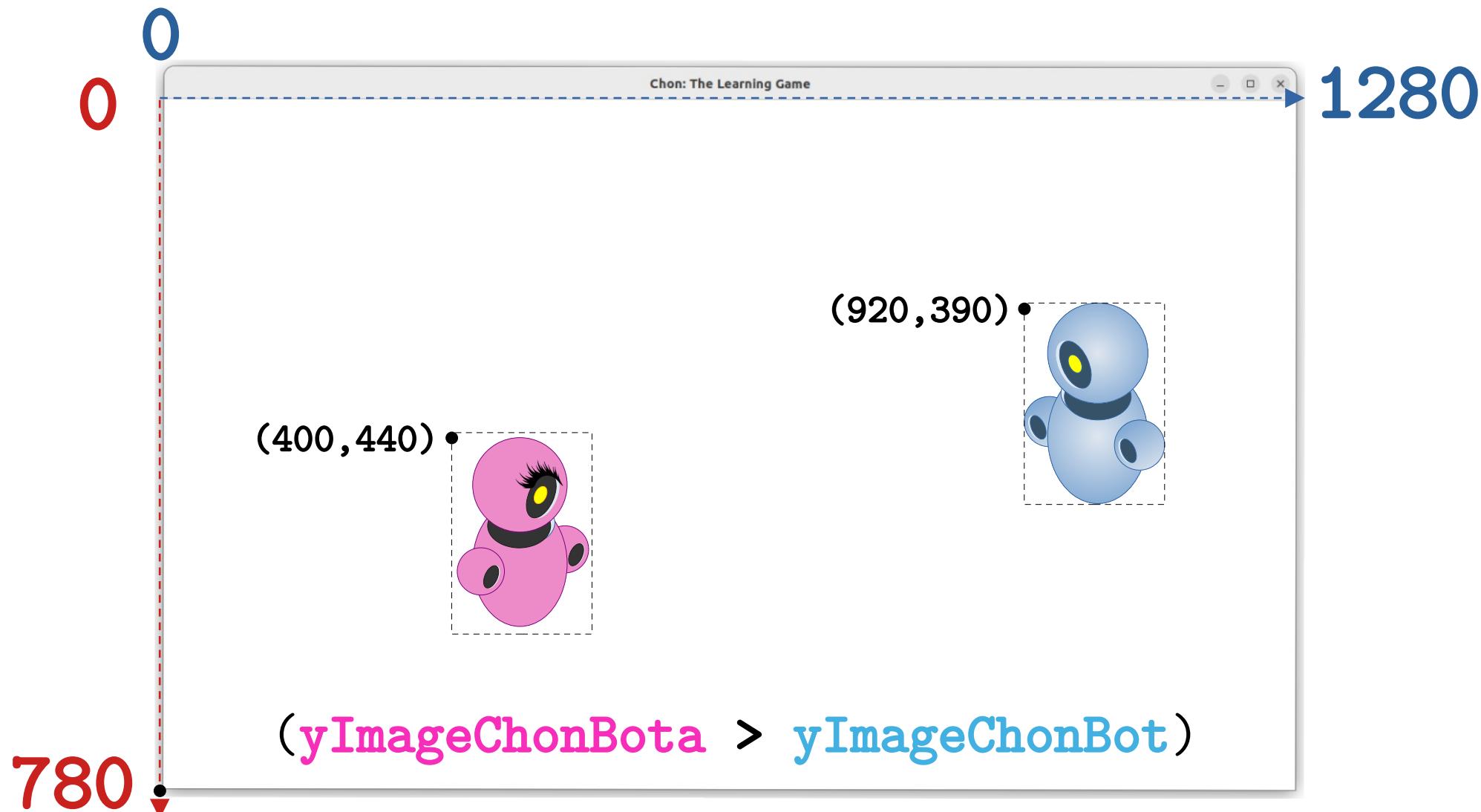
# Chasing the Player



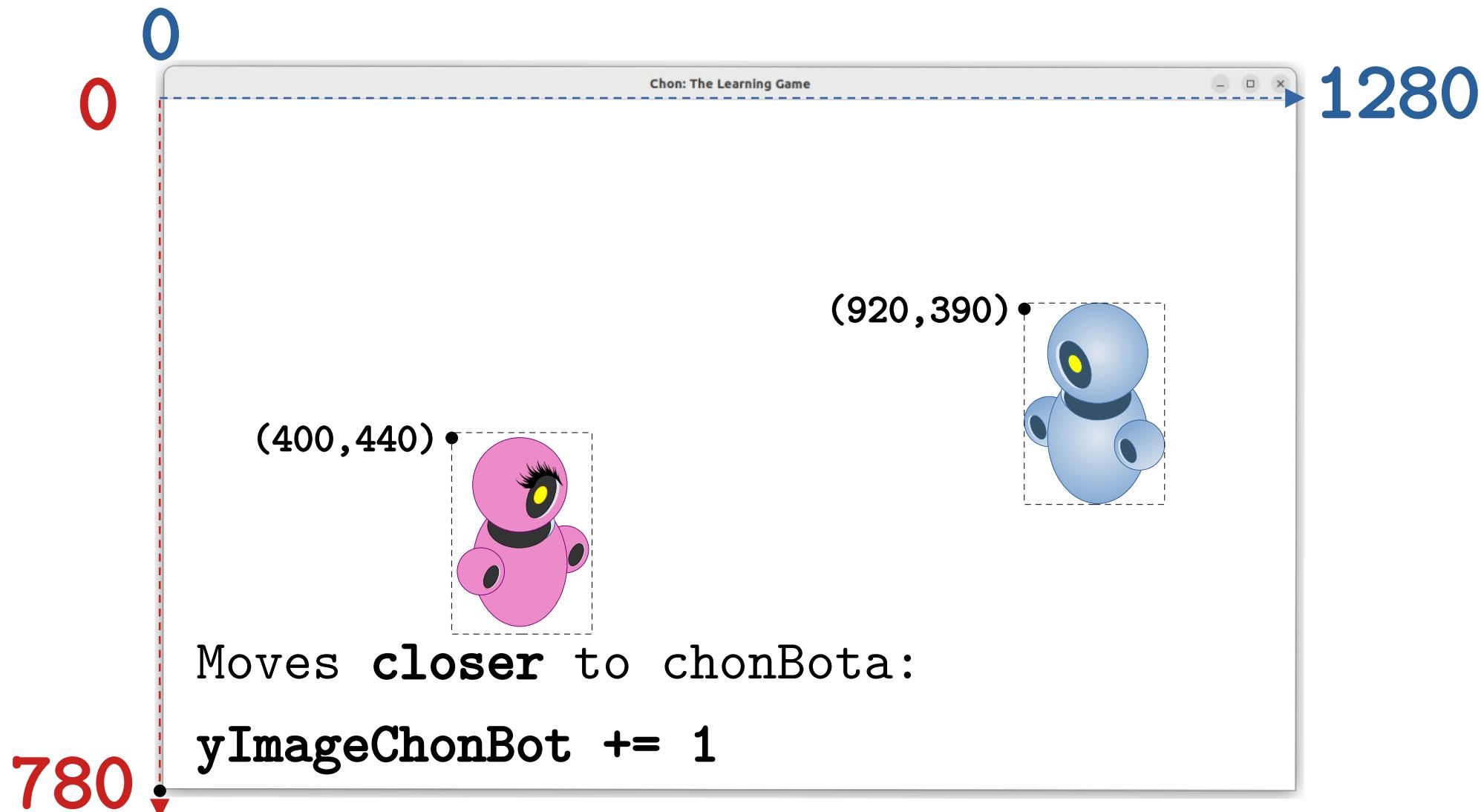
# Chasing the Player



# Chasing the Player



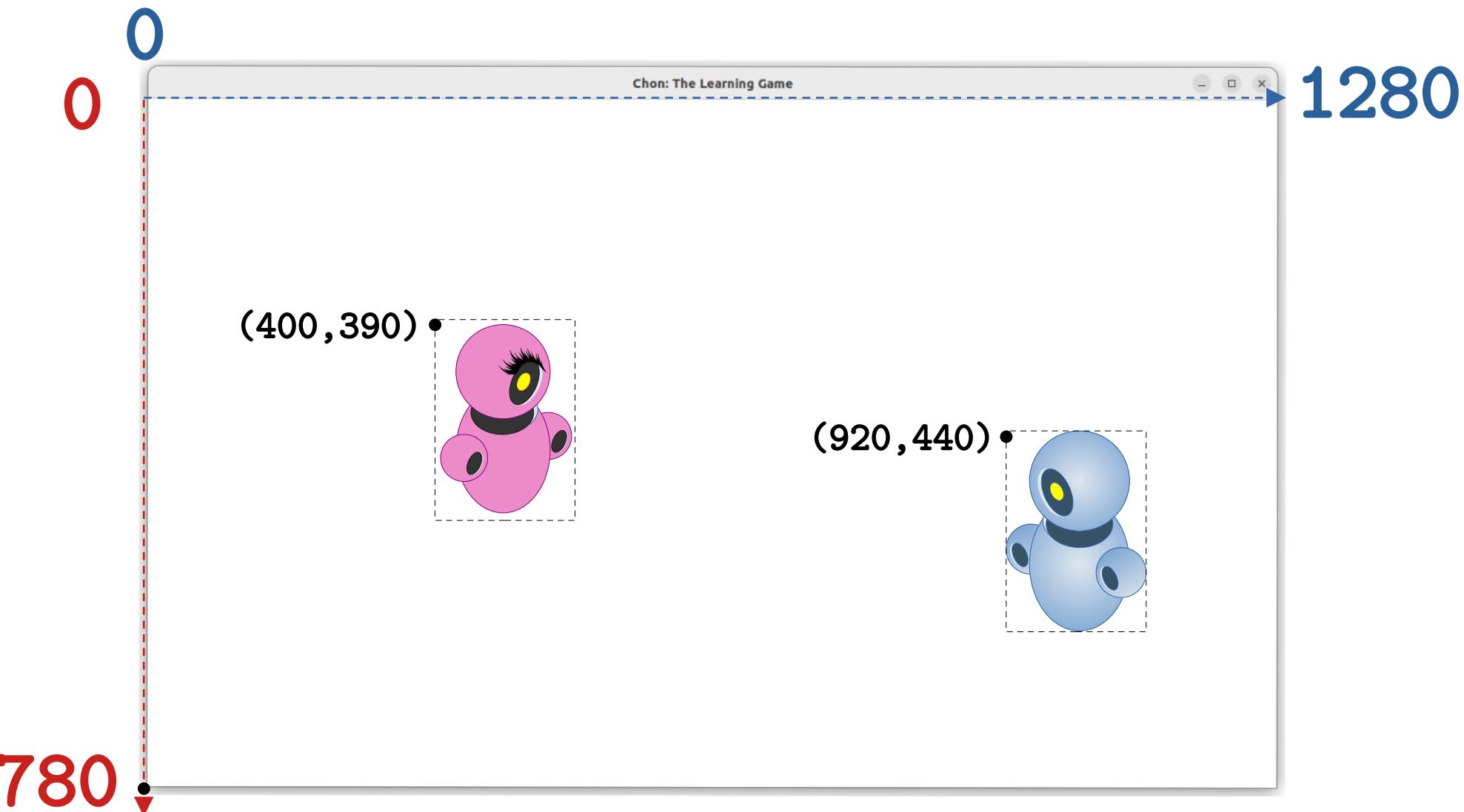
# Chasing the Player



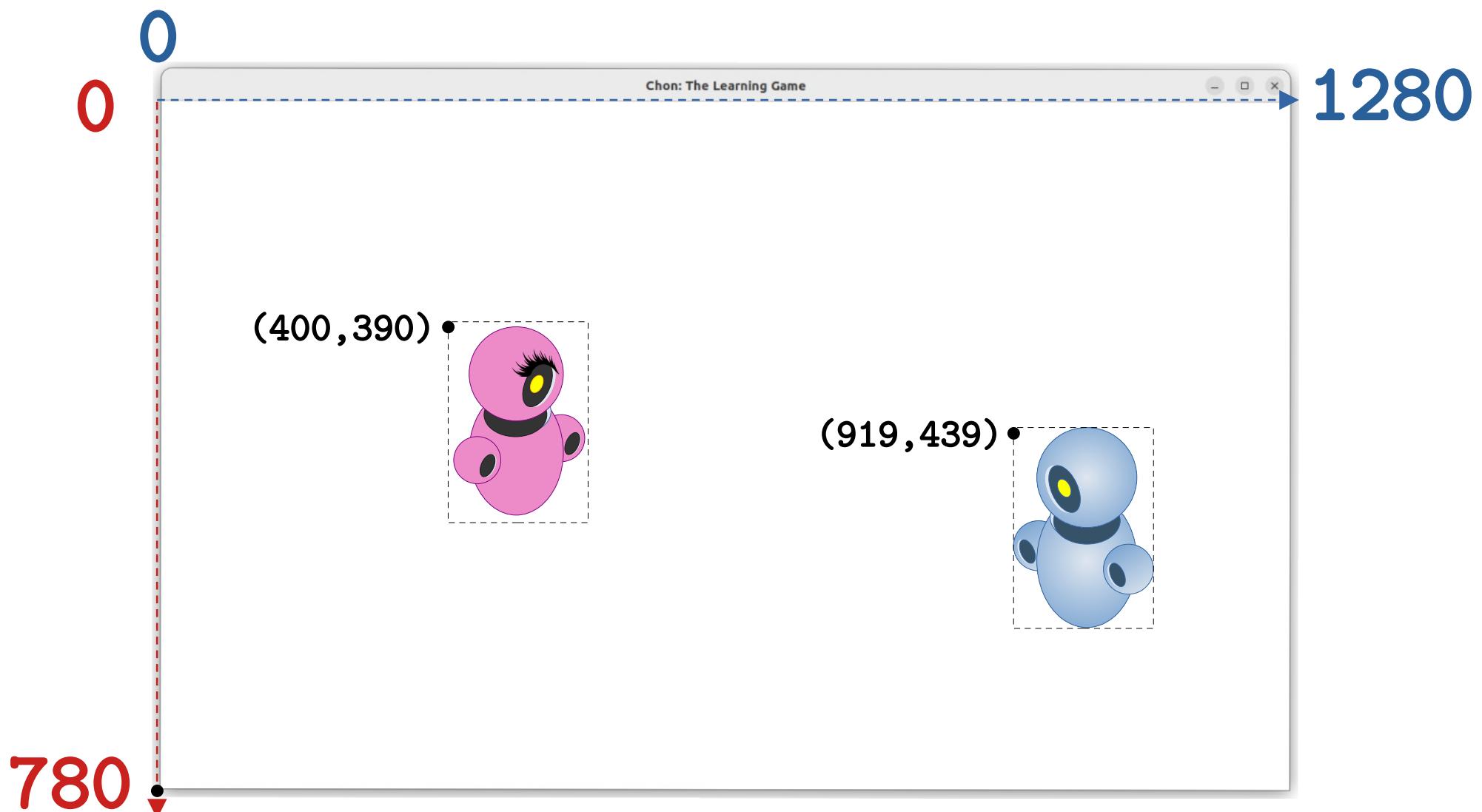
# Chasing the Player

The **two conditions**  
for **x** and **y**  
apply at the same time.

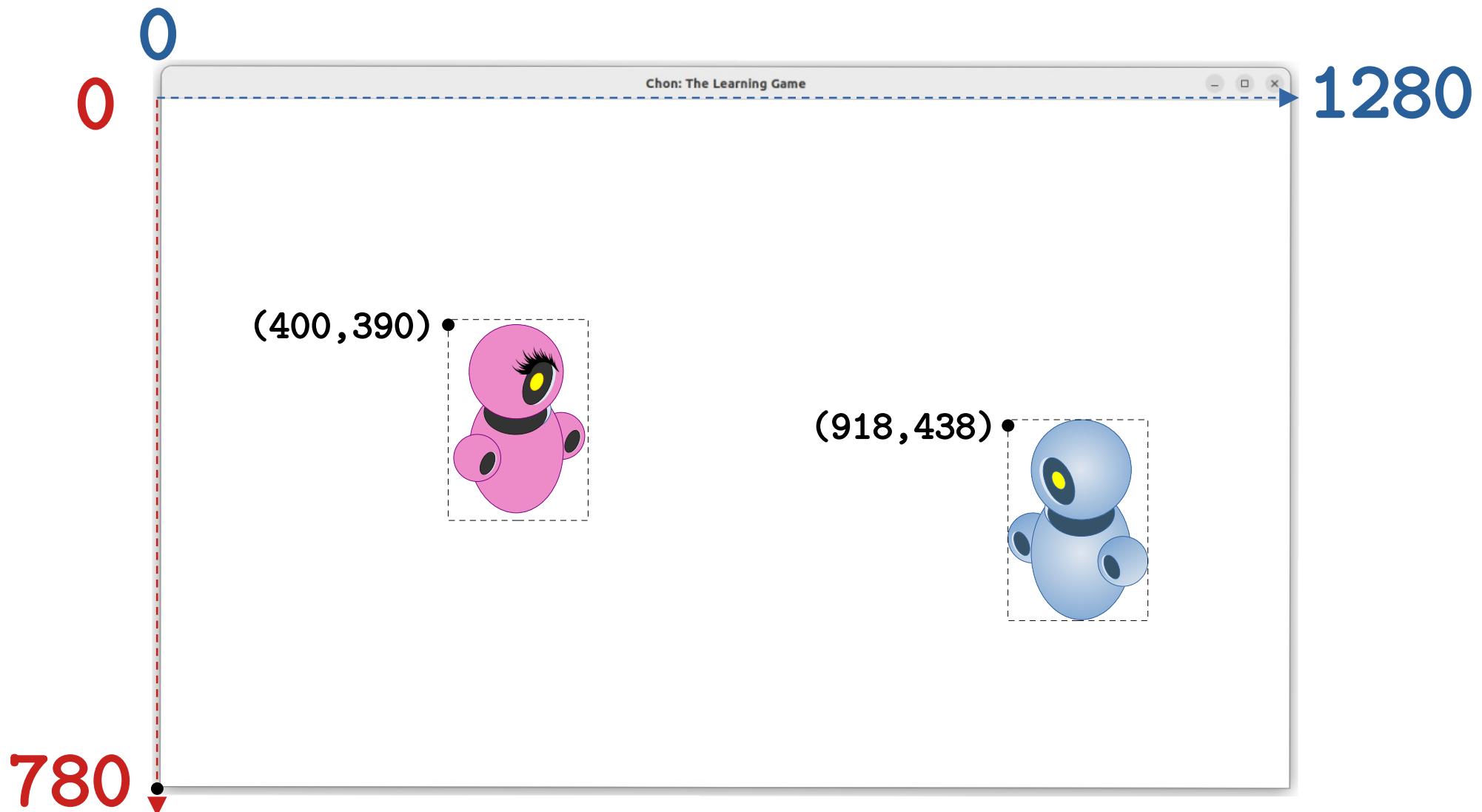
# Chasing the Player



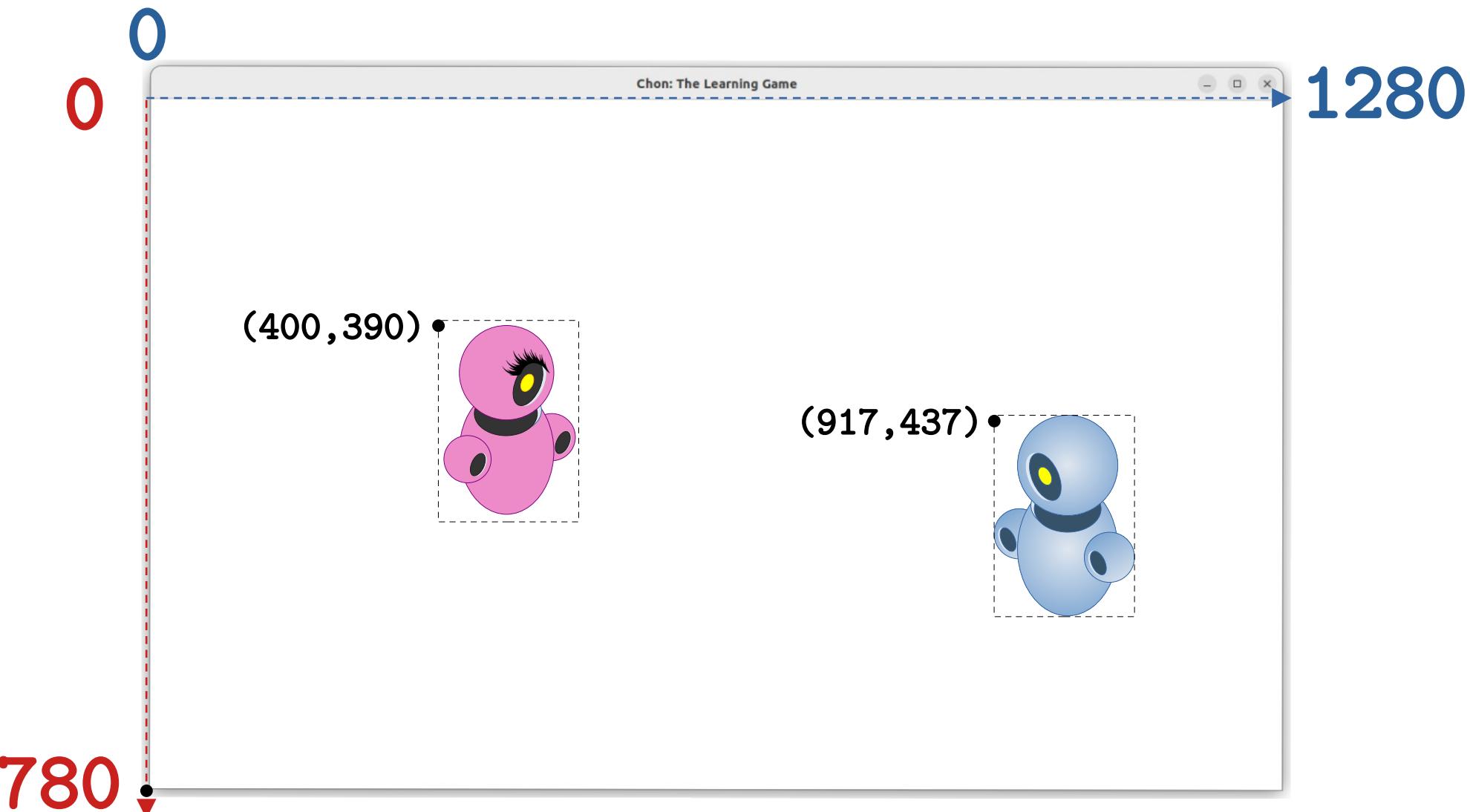
# Chasing the Player



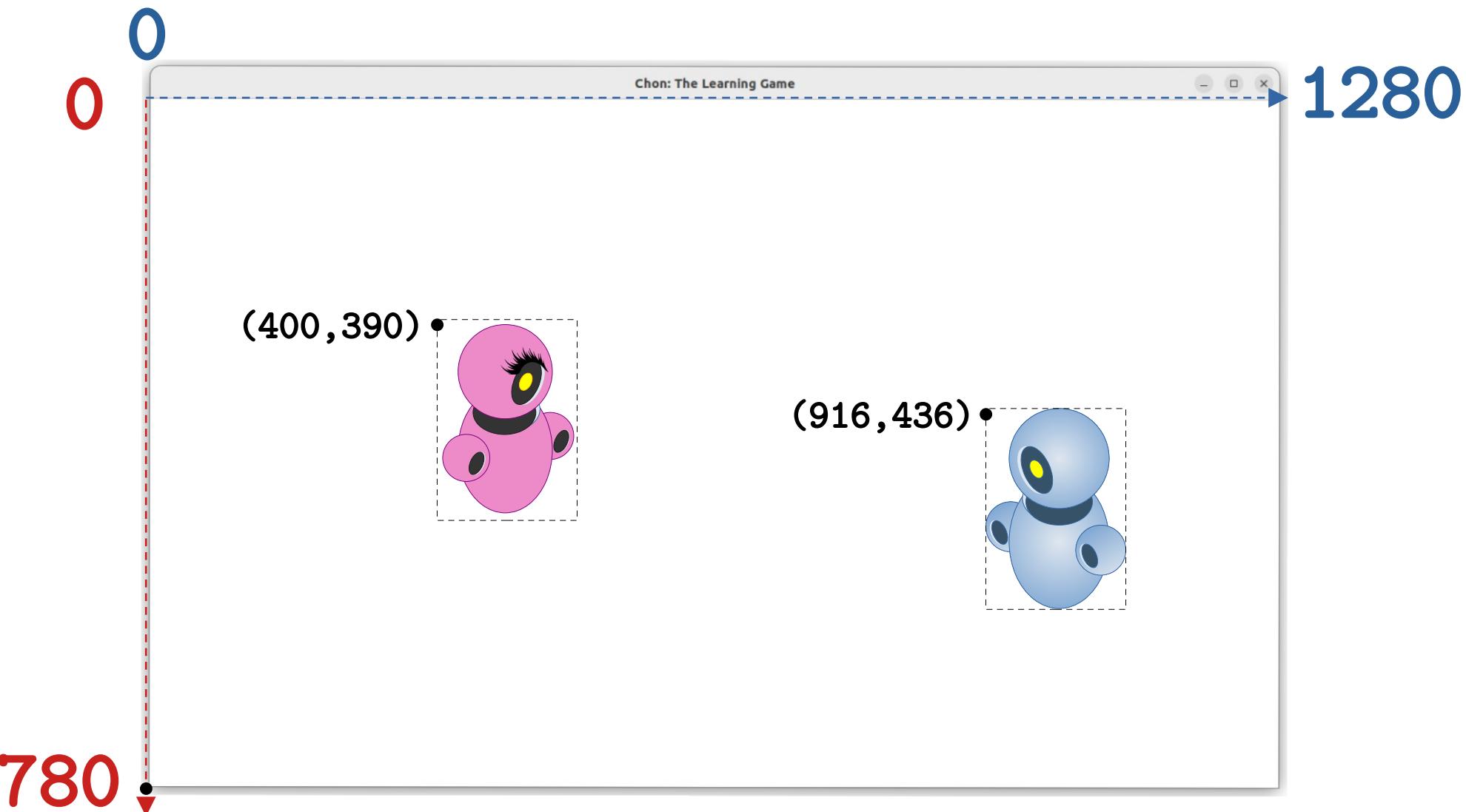
# Chasing the Player



# Chasing the Player



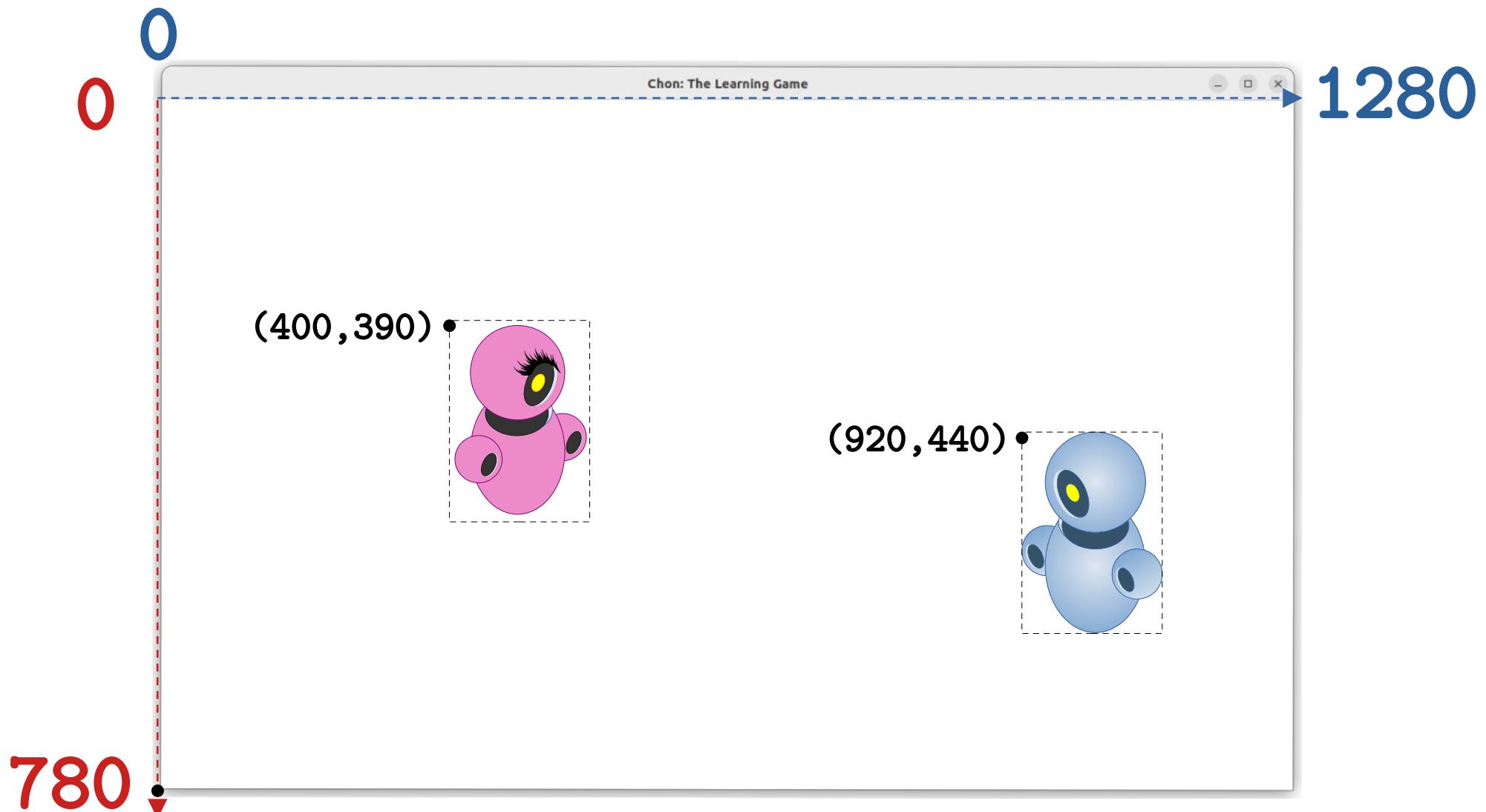
# Chasing the Player



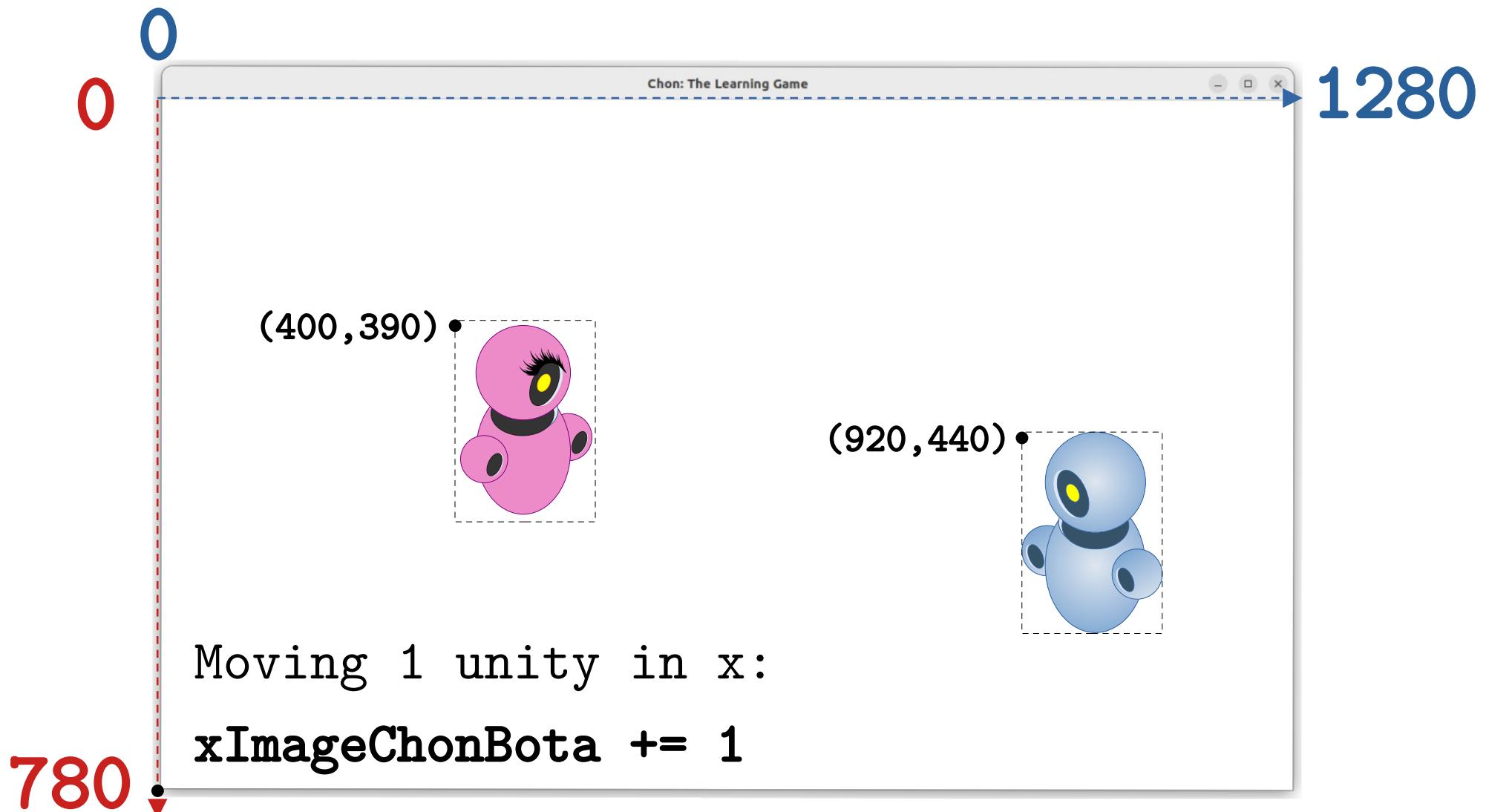
# Defining Speed

The **speed** is related to **how many** unities the object moves in **each cycle** of the **game** loop.

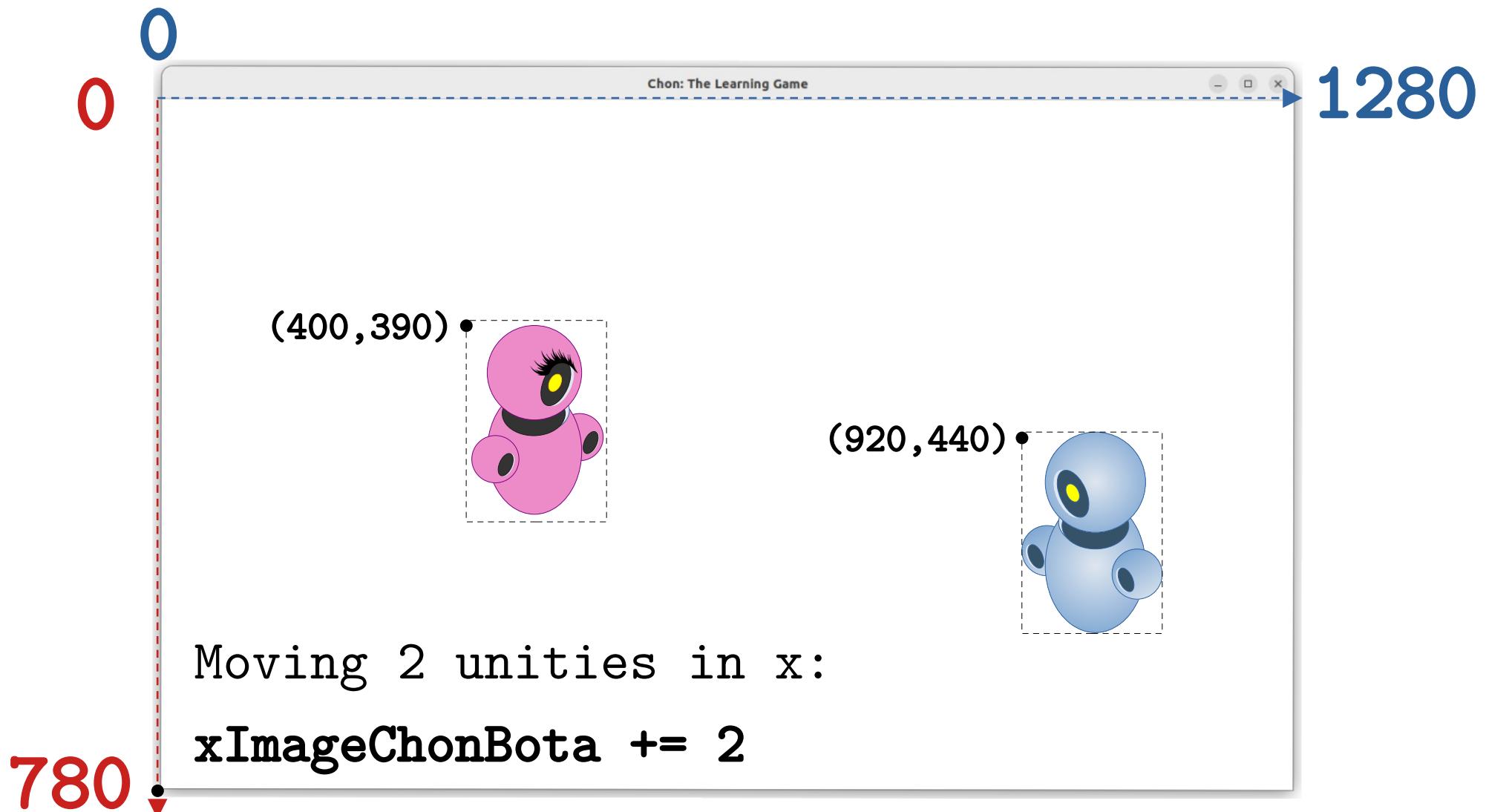
# Defining Speed



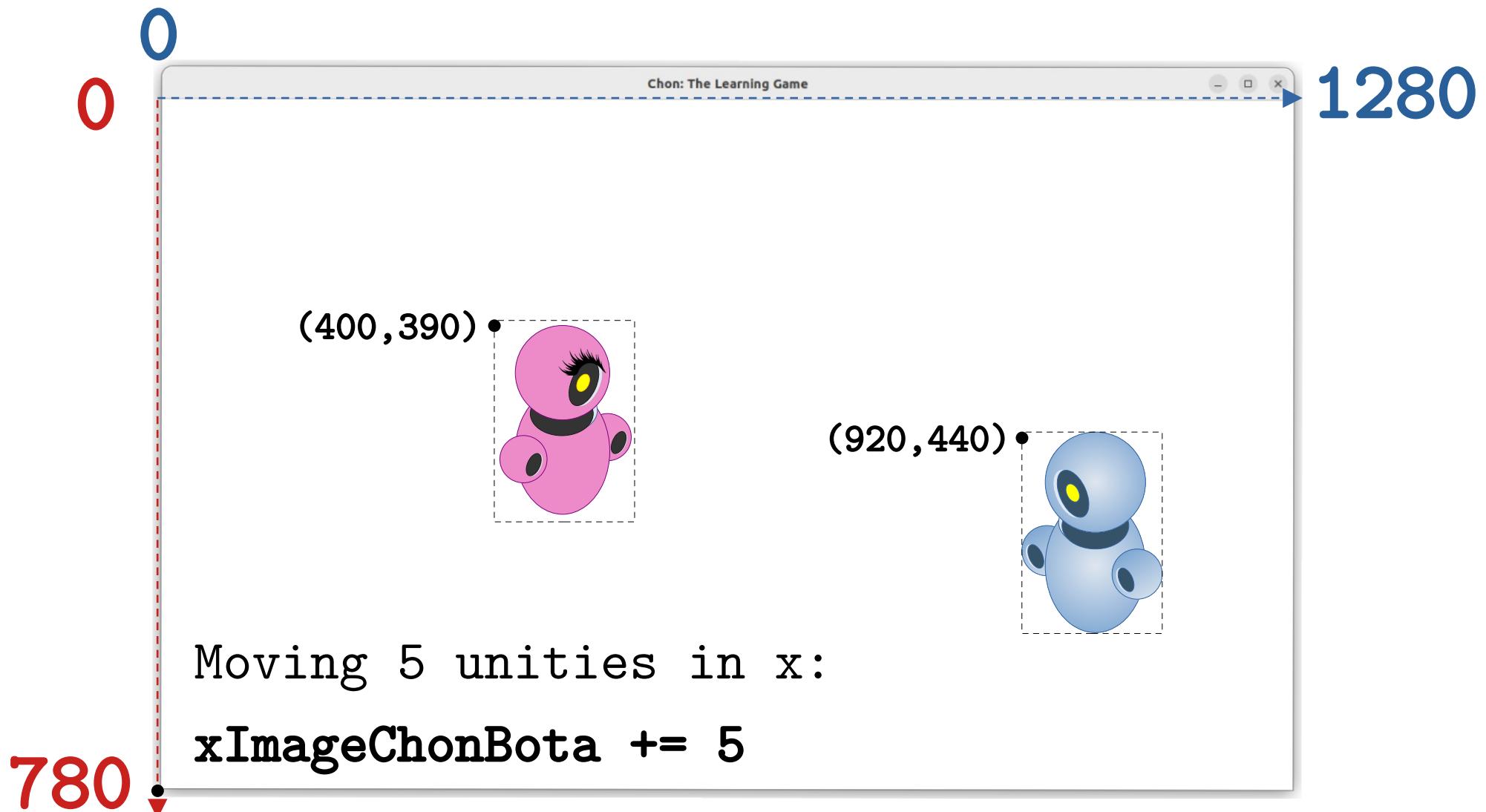
# Defining Speed



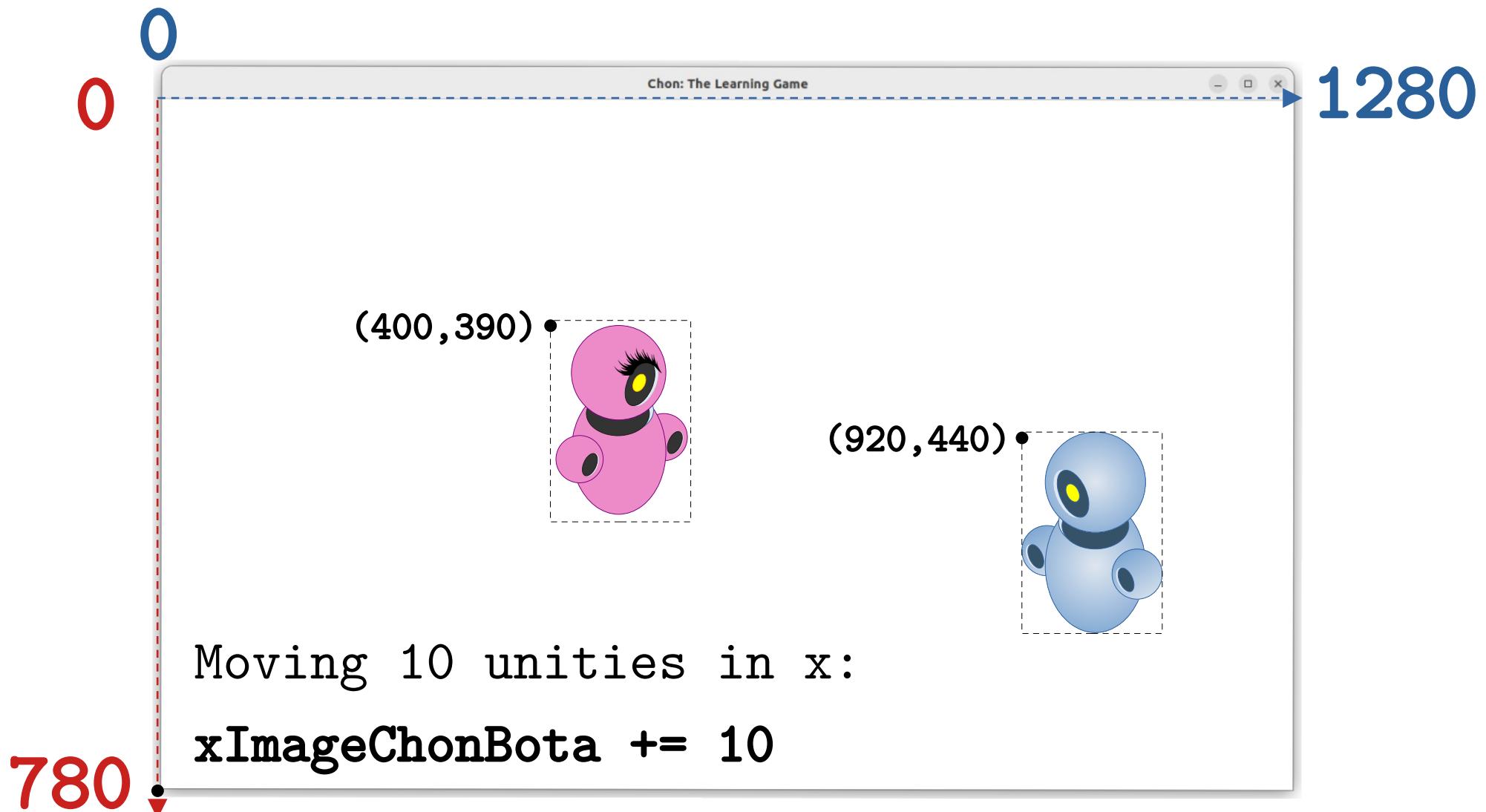
# Defining Speed



# Defining Speed



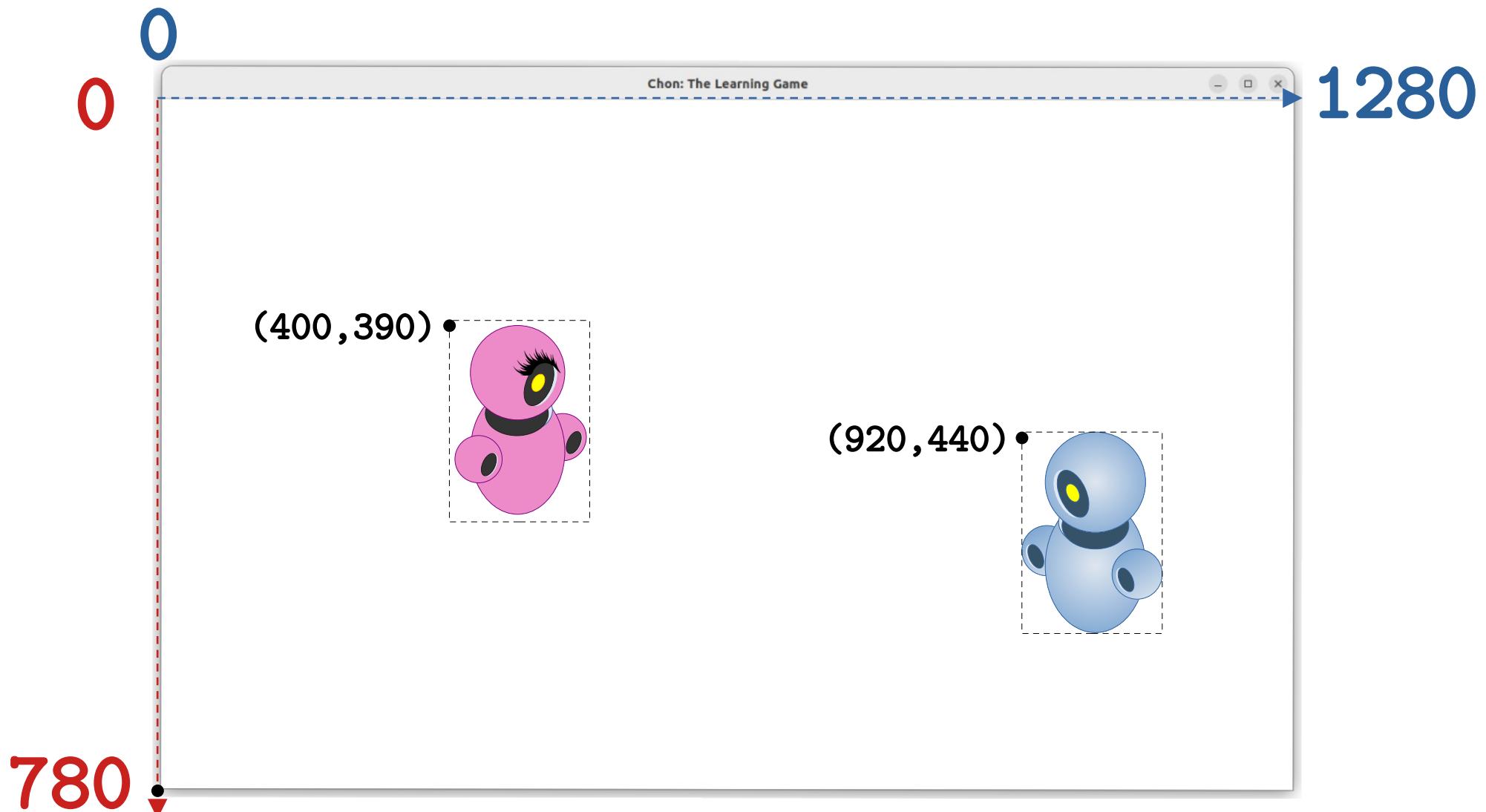
# Defining Speed



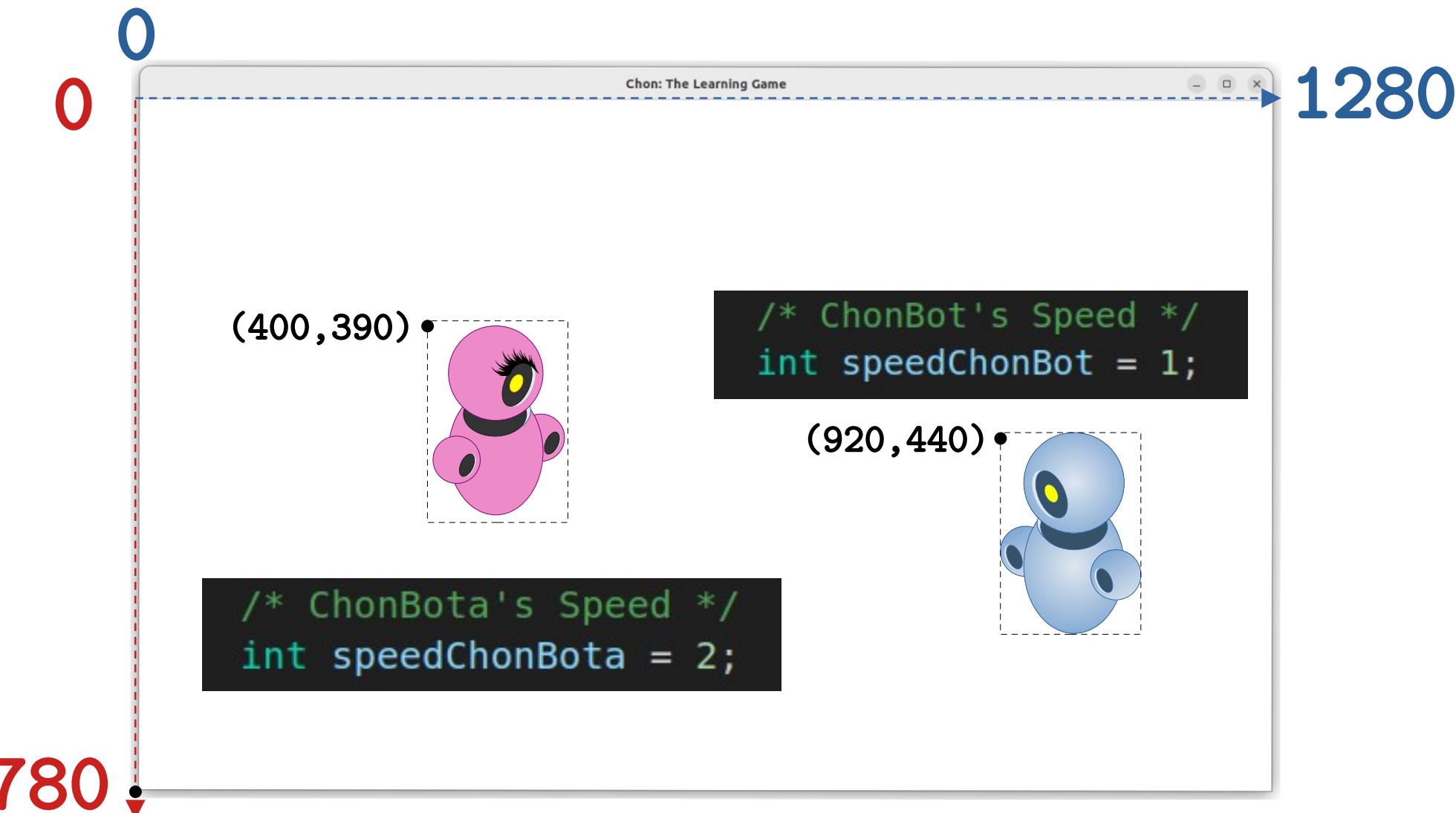
# Defining Speed

So , the **speed** is a pre-defined integer **unity** .

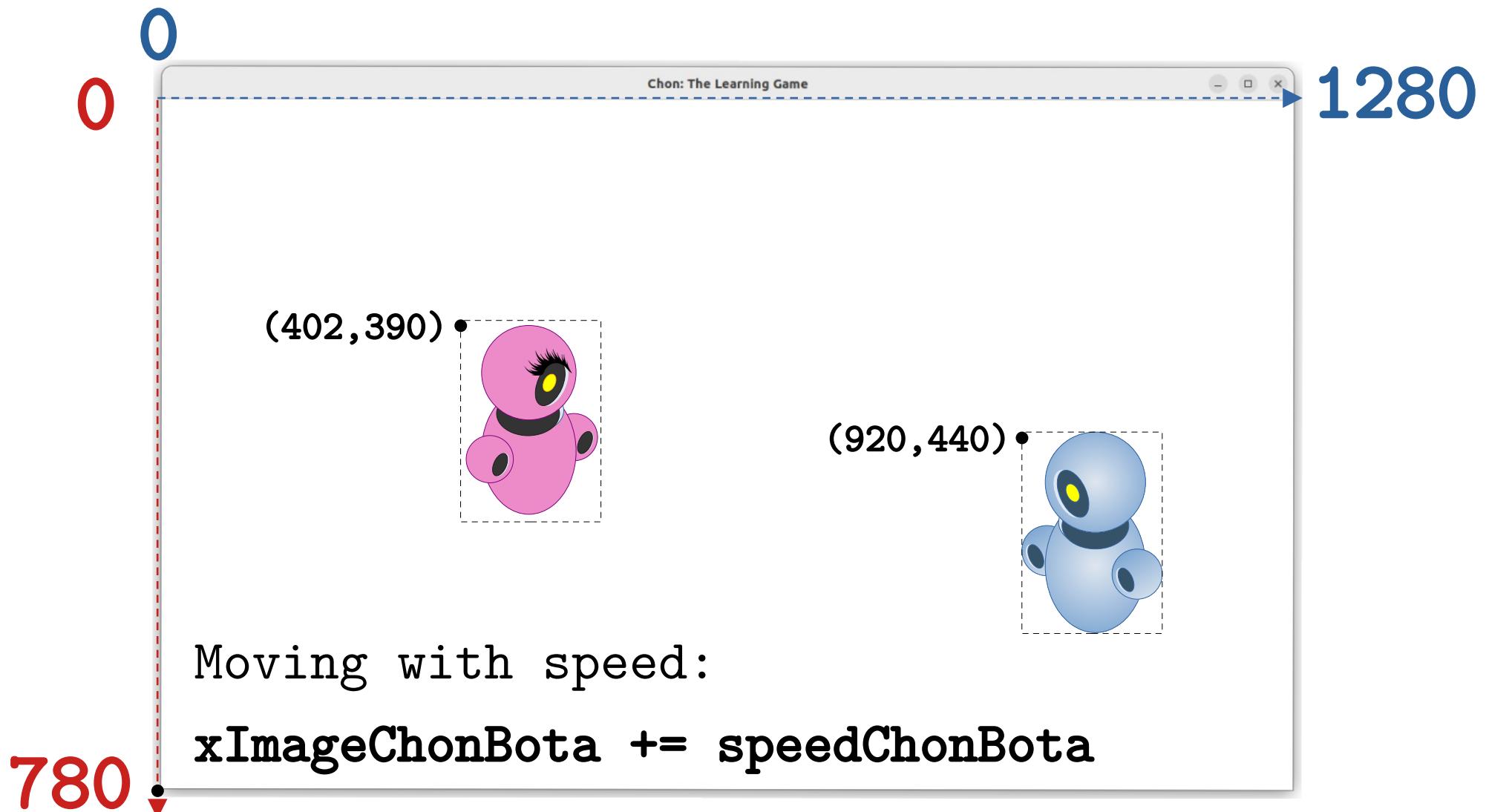
# Defining Speed



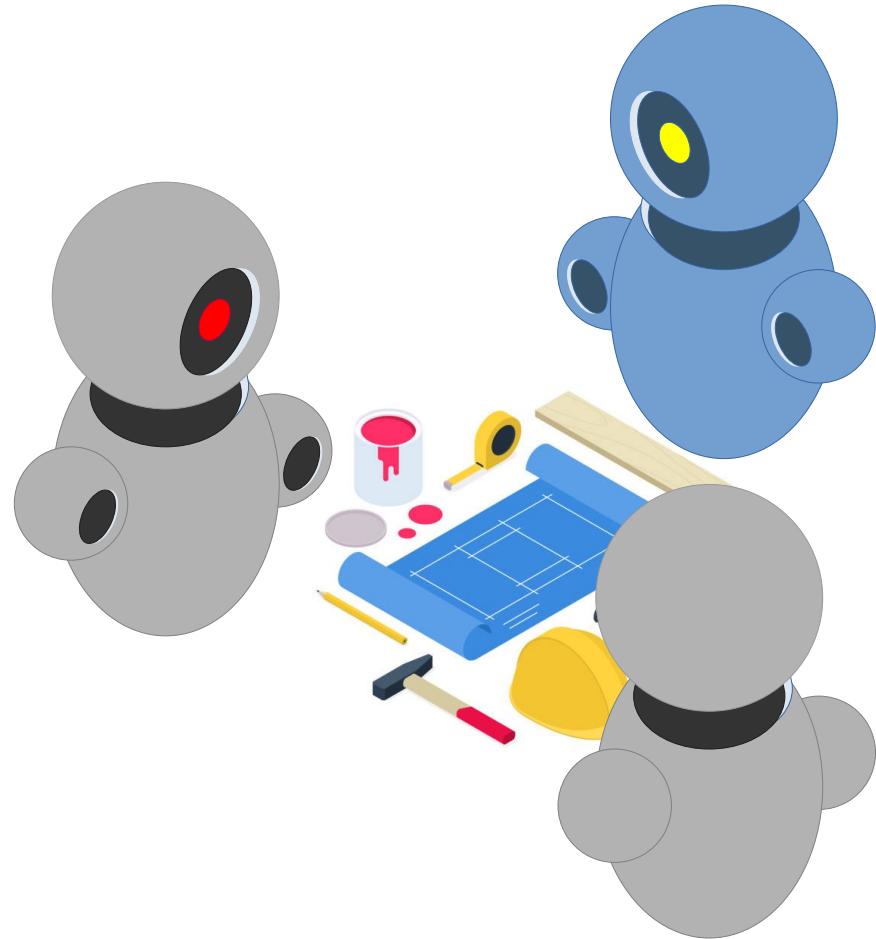
# Defining Speed



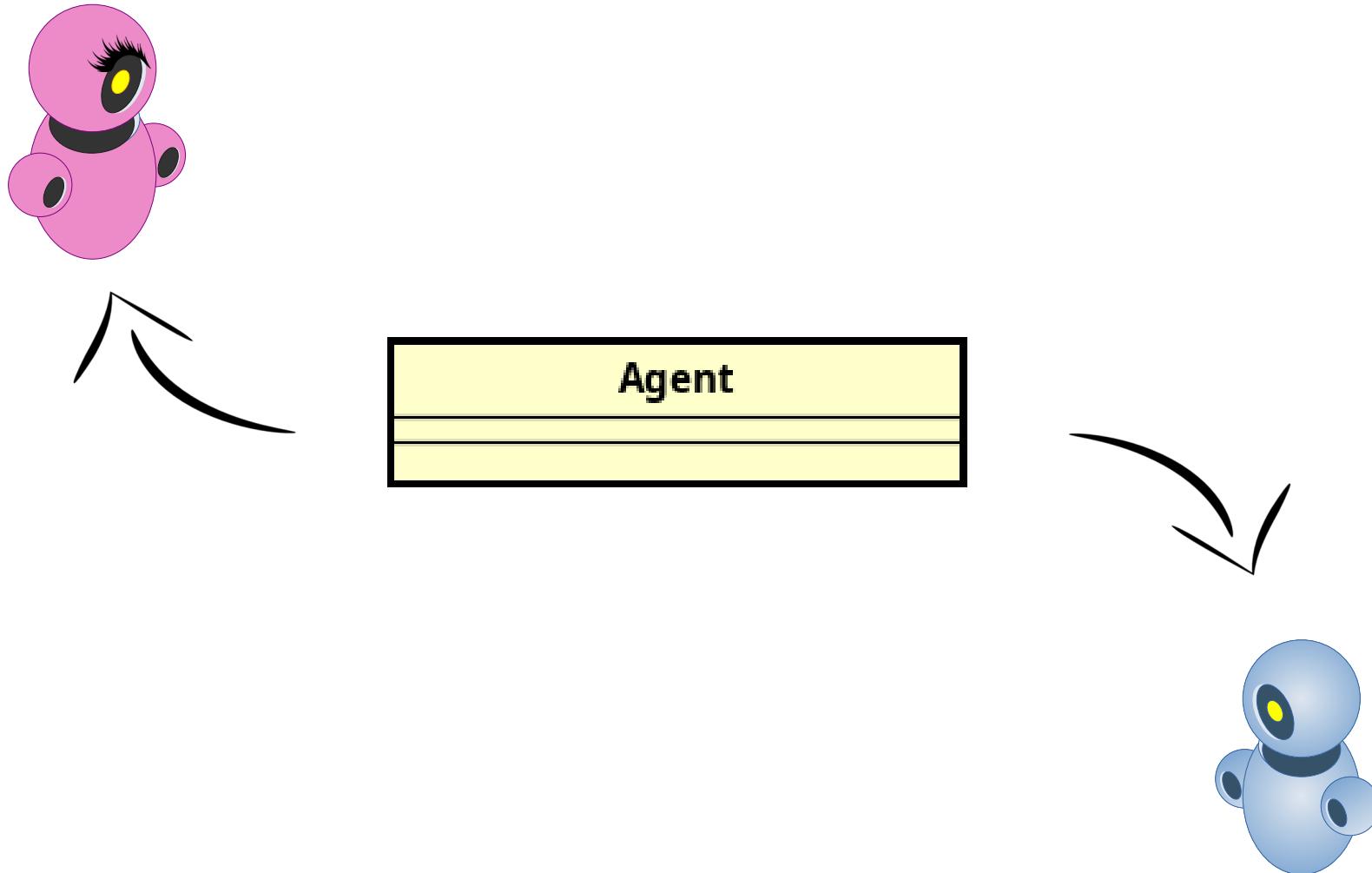
# Defining Speed



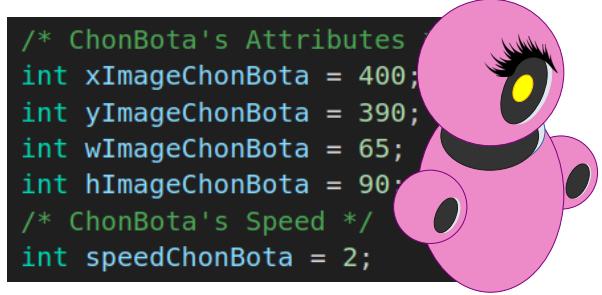
# MODELING THE ENGINE



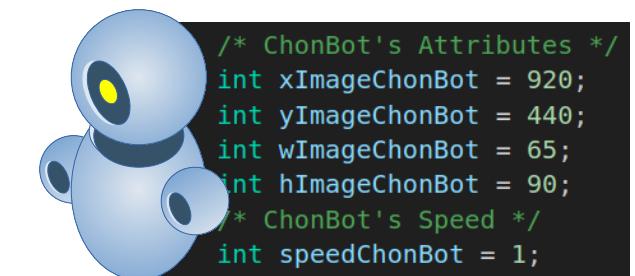
# Class Agent



# Class Agent

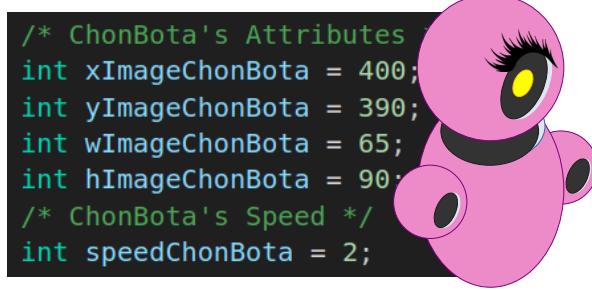


```
/* ChonBota's Attributes  
int xImageChonBota = 400;  
int yImageChonBota = 390;  
int wImageChonBota = 65;  
int hImageChonBota = 90;  
/* ChonBota's Speed */  
int speedChonBota = 2;
```

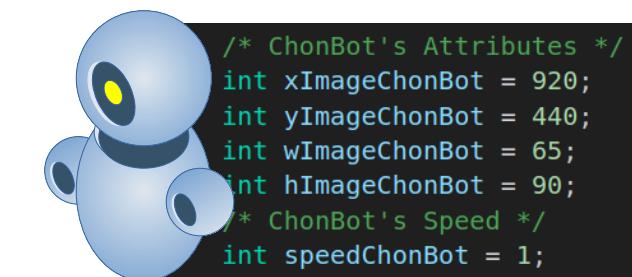
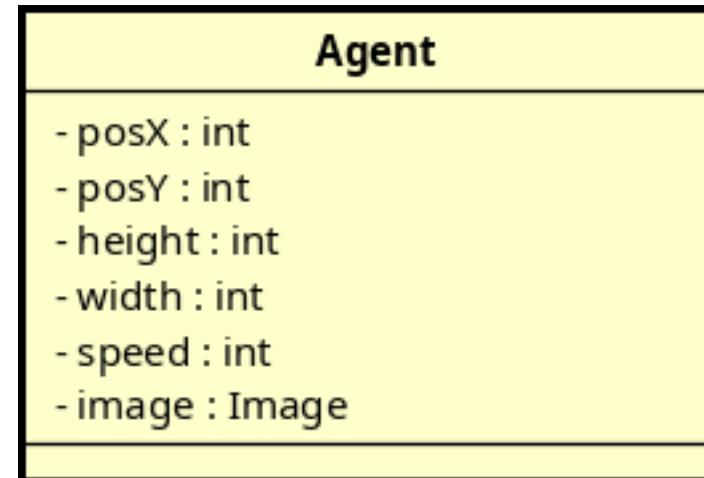


```
/* ChonBot's Attributes */  
int xImageChonBot = 920;  
int yImageChonBot = 440;  
int wImageChonBot = 65;  
int hImageChonBot = 90;  
/* ChonBot's Speed */  
int speedChonBot = 1;
```

# Class Agent



```
/* ChonBota's Attributes  
int xImageChonBota = 400;  
int yImageChonBota = 390;  
int wImageChonBota = 65;  
int hImageChonBota = 90;  
/* ChonBota's Speed */  
int speedChonBota = 2;
```



```
/* ChonBot's Attributes */  
int xImageChonBot = 920;  
int yImageChonBot = 440;  
int wImageChonBot = 65;  
int hImageChonBot = 90;  
/* ChonBot's Speed */  
int speedChonBot = 1;
```

# Class Environment

```
StackPane root = new StackPane();
Scene scene = new Scene(root, width:1280, height:780);
theStage.setTitle(value:"Chon: The Learning Game");
theStage.setScene(scene);

int wCanvas = 1280;
int hCanvas = 780;
Canvas canvas = new Canvas(wCanvas, hCanvas);
```

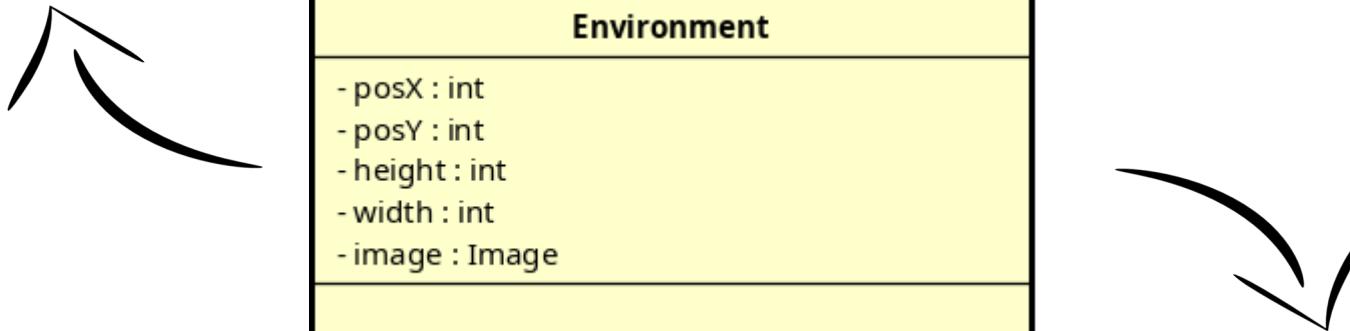


```
Image background = new Image(getClass().getResource("/images/environment/castle.png").toExternalForm());
```

# Class Environment

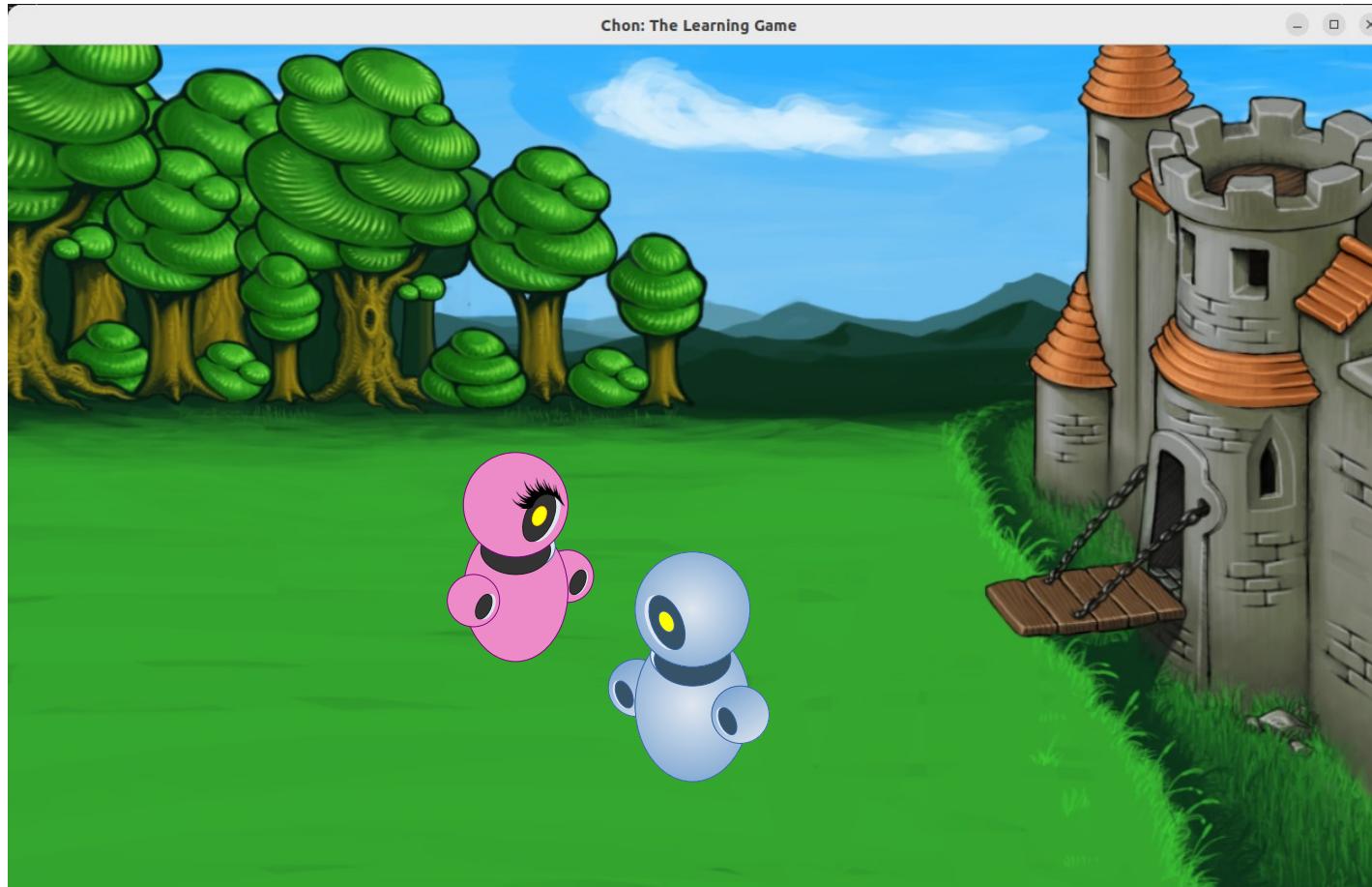
```
StackPane root = new StackPane();
Scene scene = new Scene(root, width:1280, height:780);
theStage.setTitle(value:"Chon: The Learning Game");
theStage.setScene(scene);

int wCanvas = 1280;
int hCanvas = 780;
Canvas canvas = new Canvas(wCanvas, hCanvas);
```

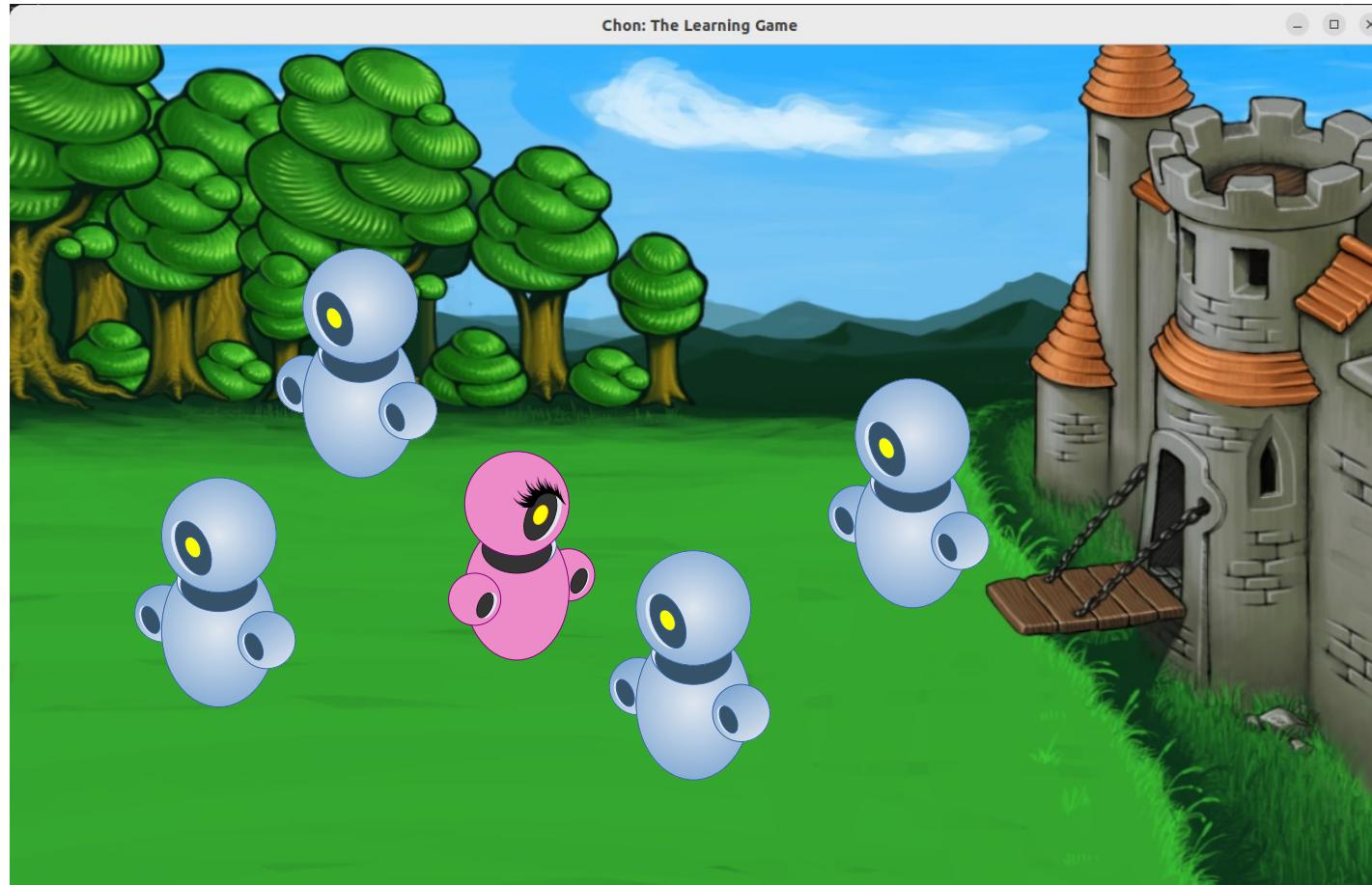


```
Image background = new Image(getClass().getResource("/images/environment/castle.png").toExternalForm());
```

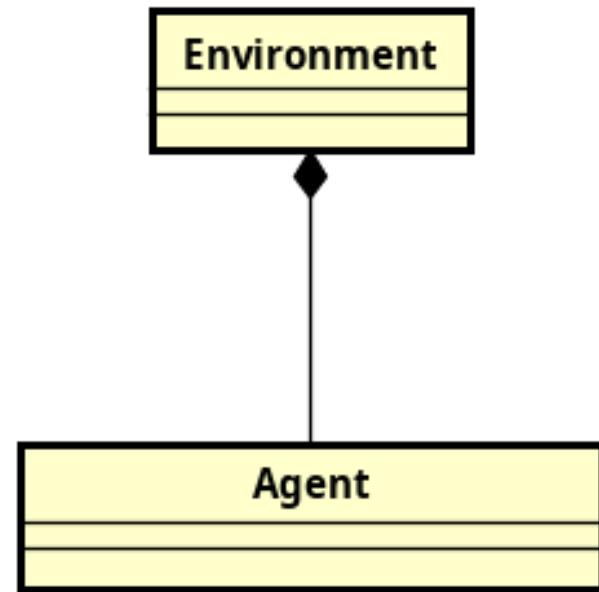
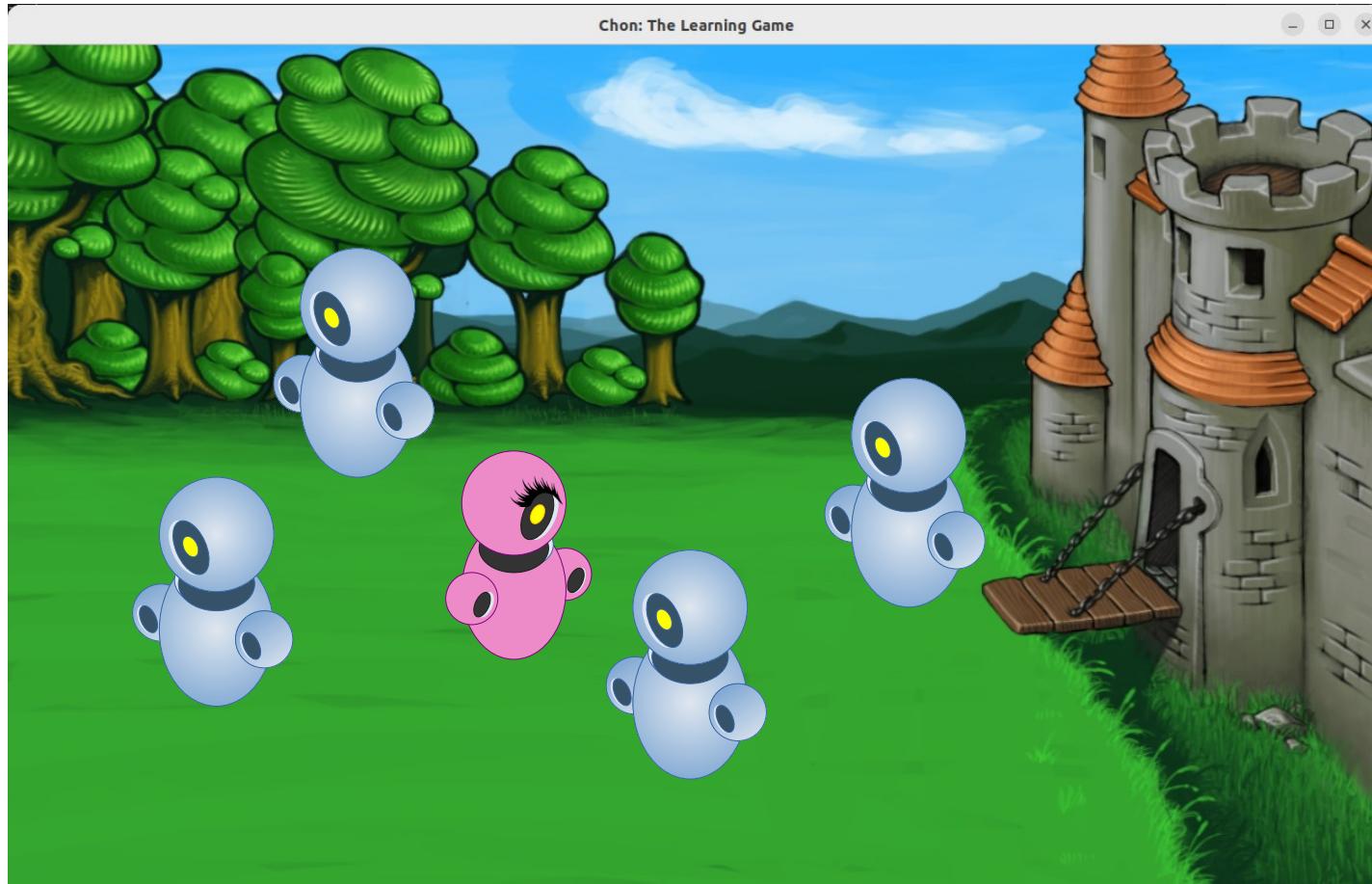
# The Class Diagram



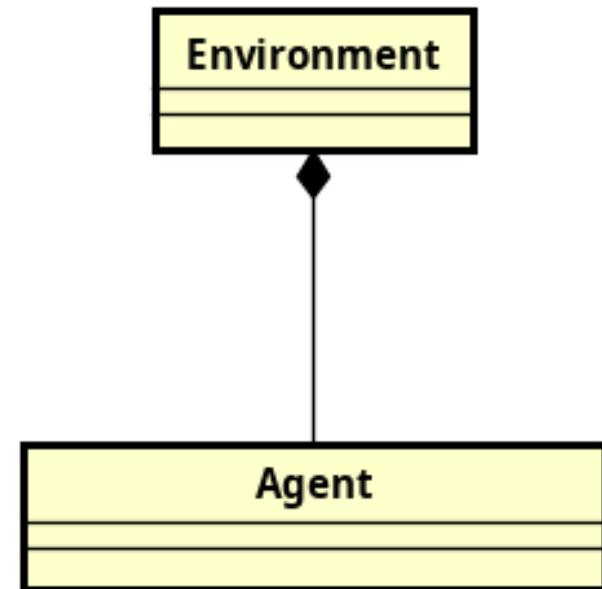
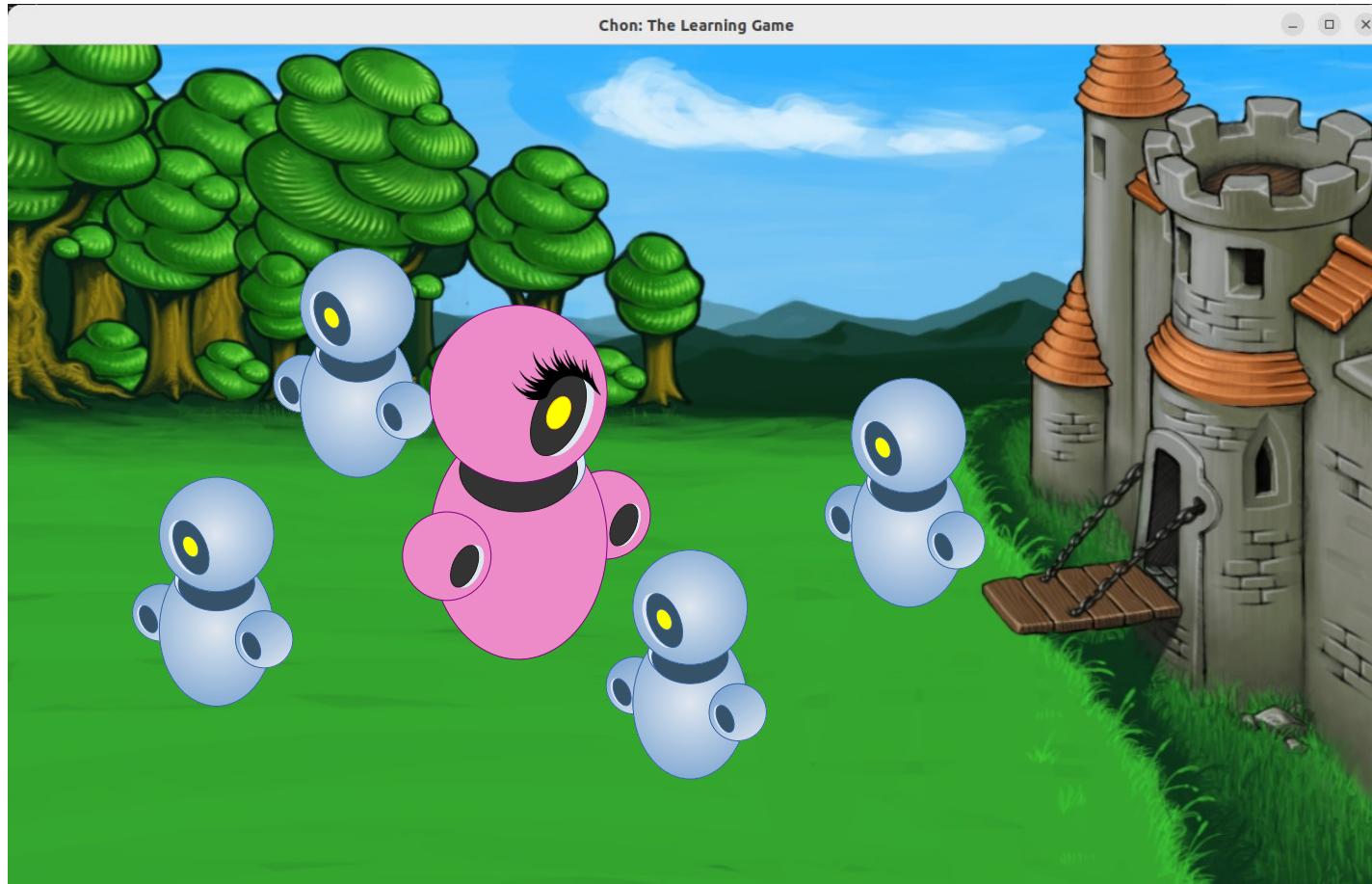
# The Class Diagram



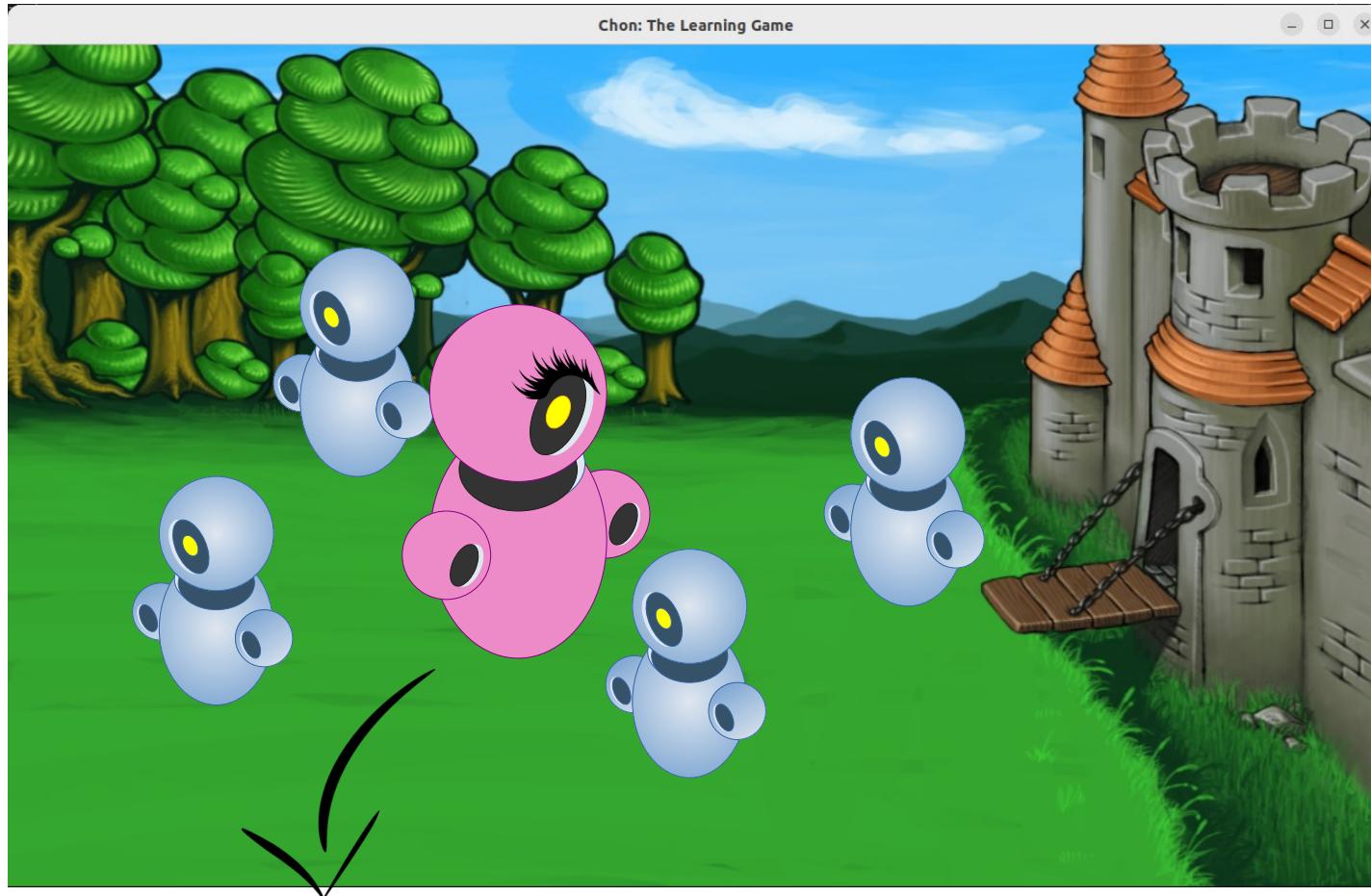
# The Class Diagram



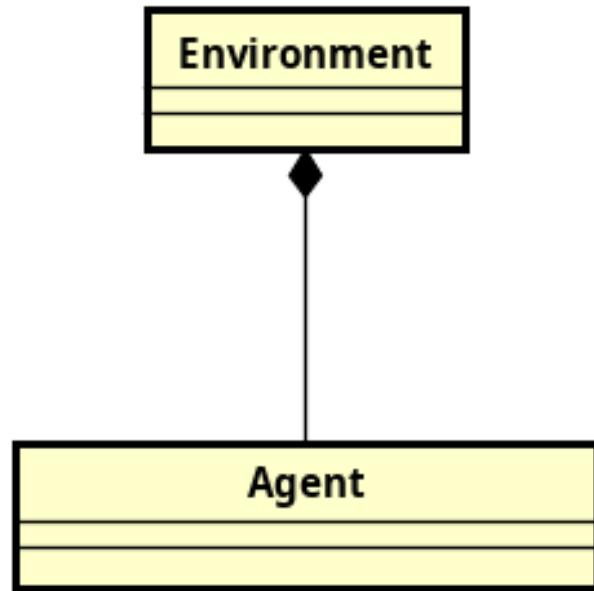
# The Class Diagram



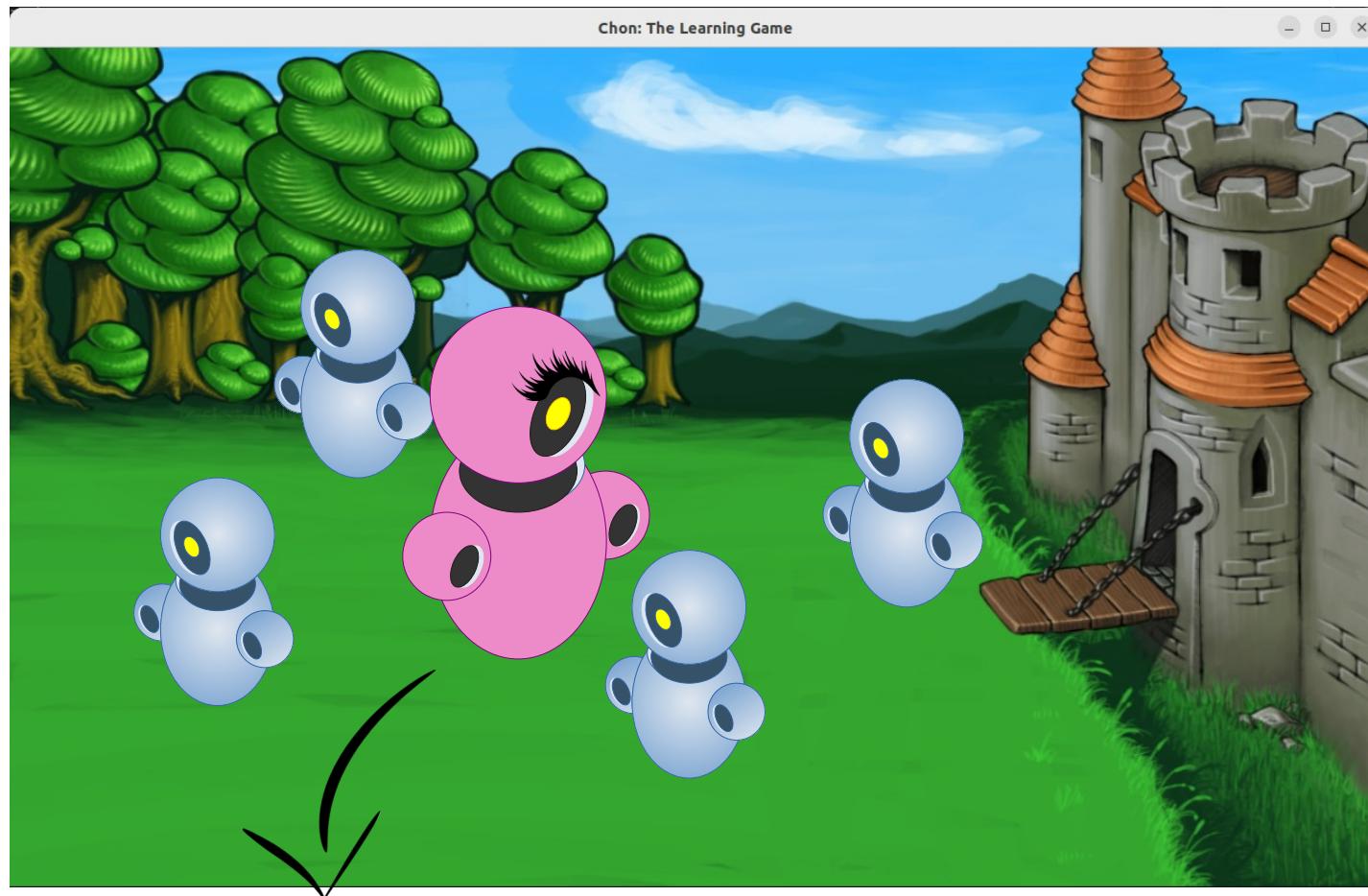
# The Class Diagram



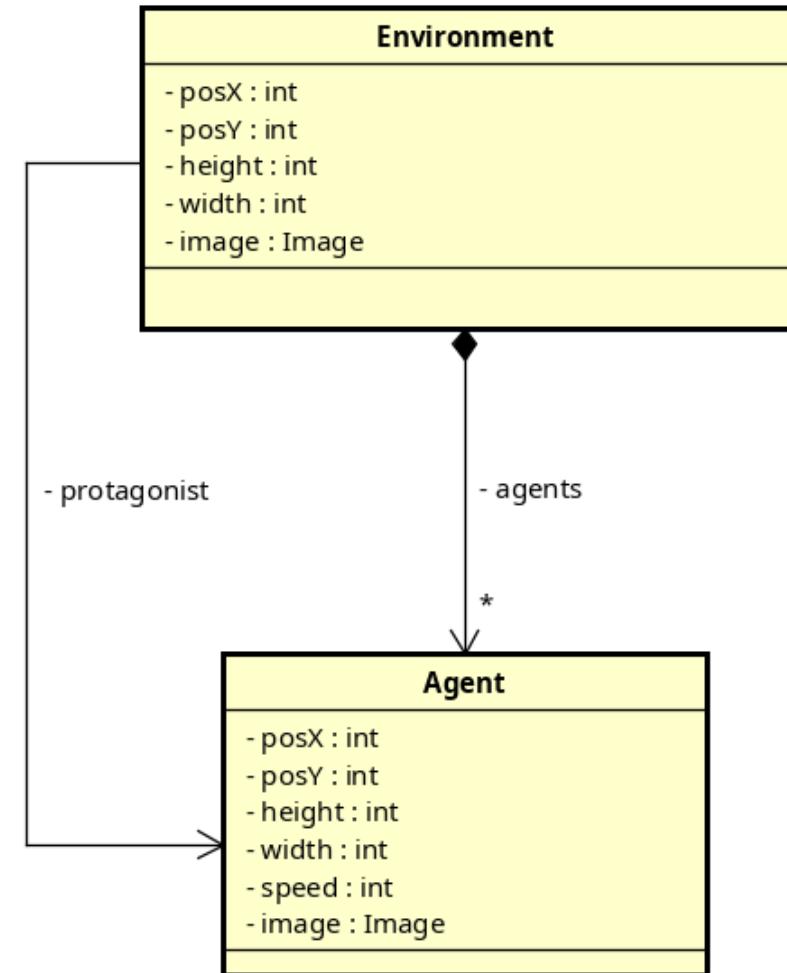
**protagonist**



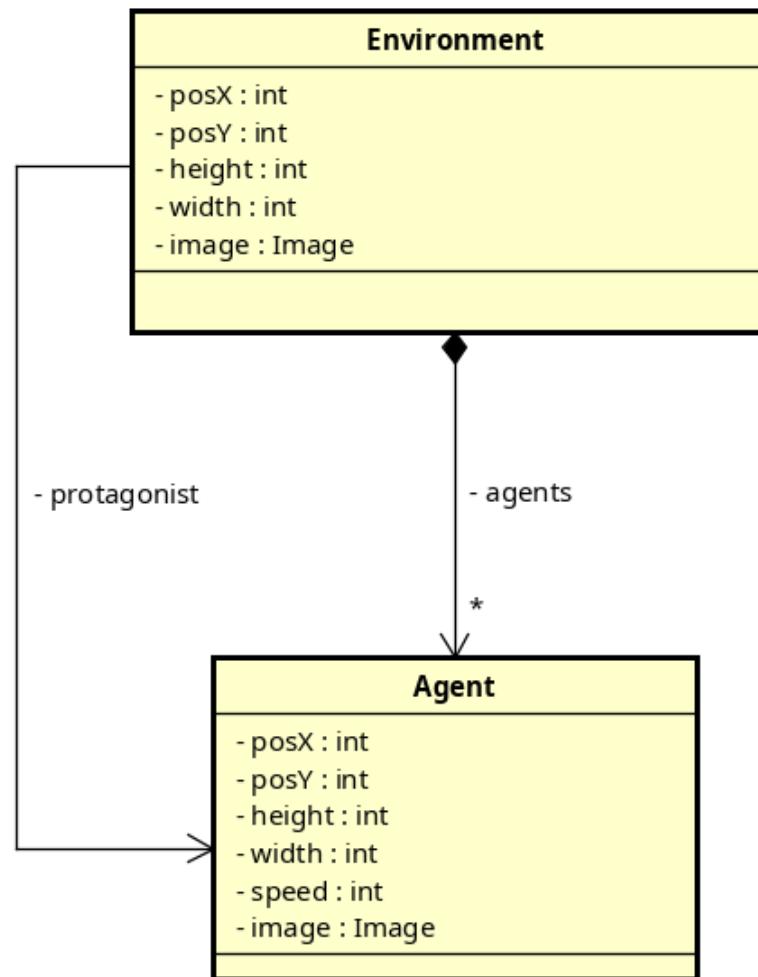
# The Class Diagram



protagonist

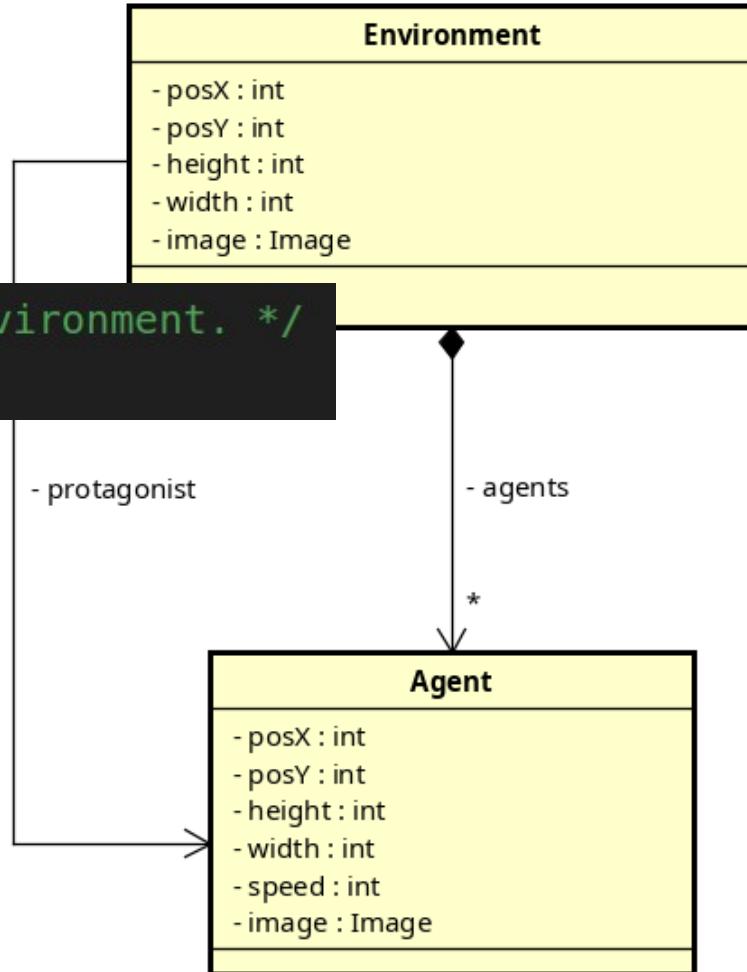


# The Class Diagram

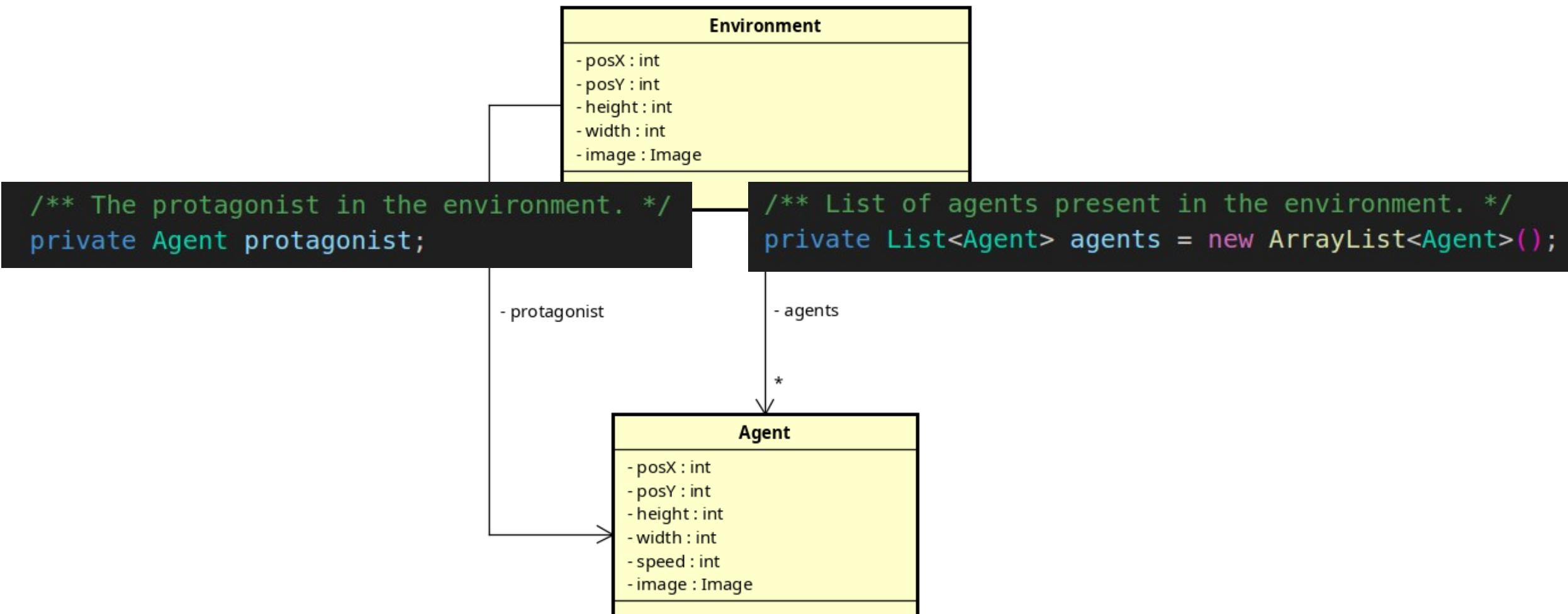


# The Class Diagram

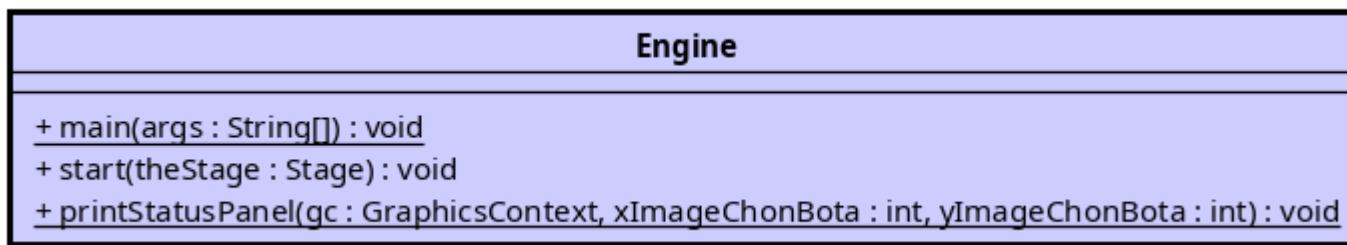
```
/** The protagonist in the environment. */
private Agent protagonist;
```



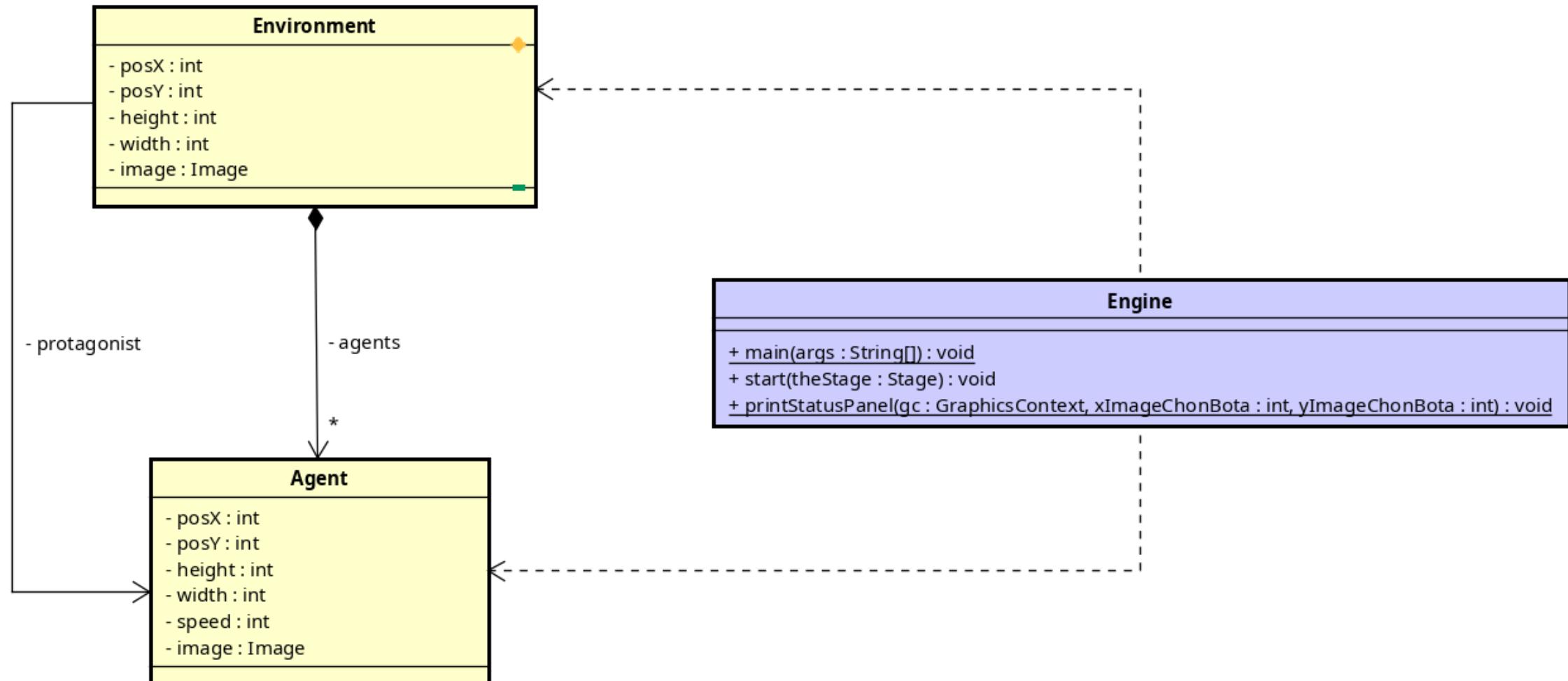
# The Class Diagram



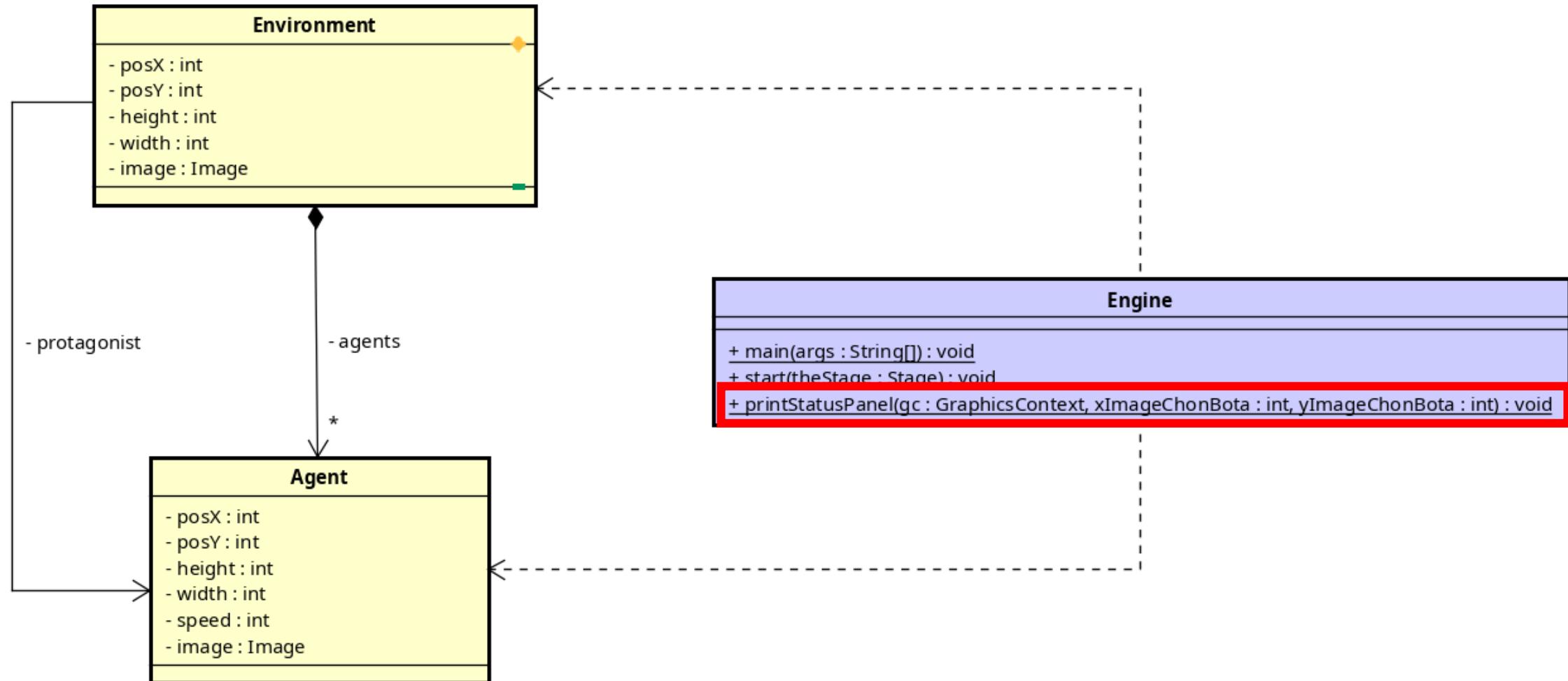
# The Class Diagram



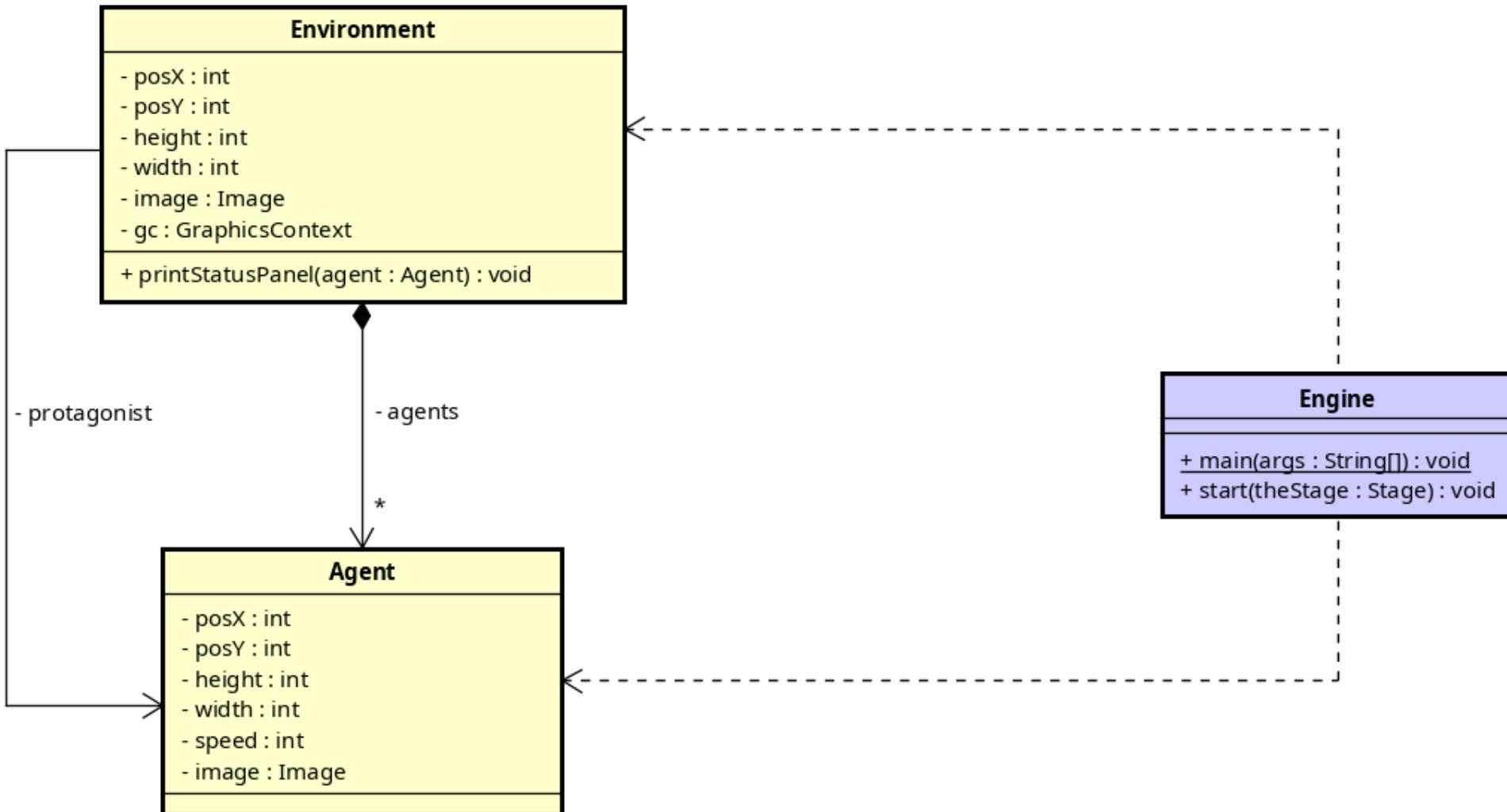
# The Class Diagram



# The Class Diagram

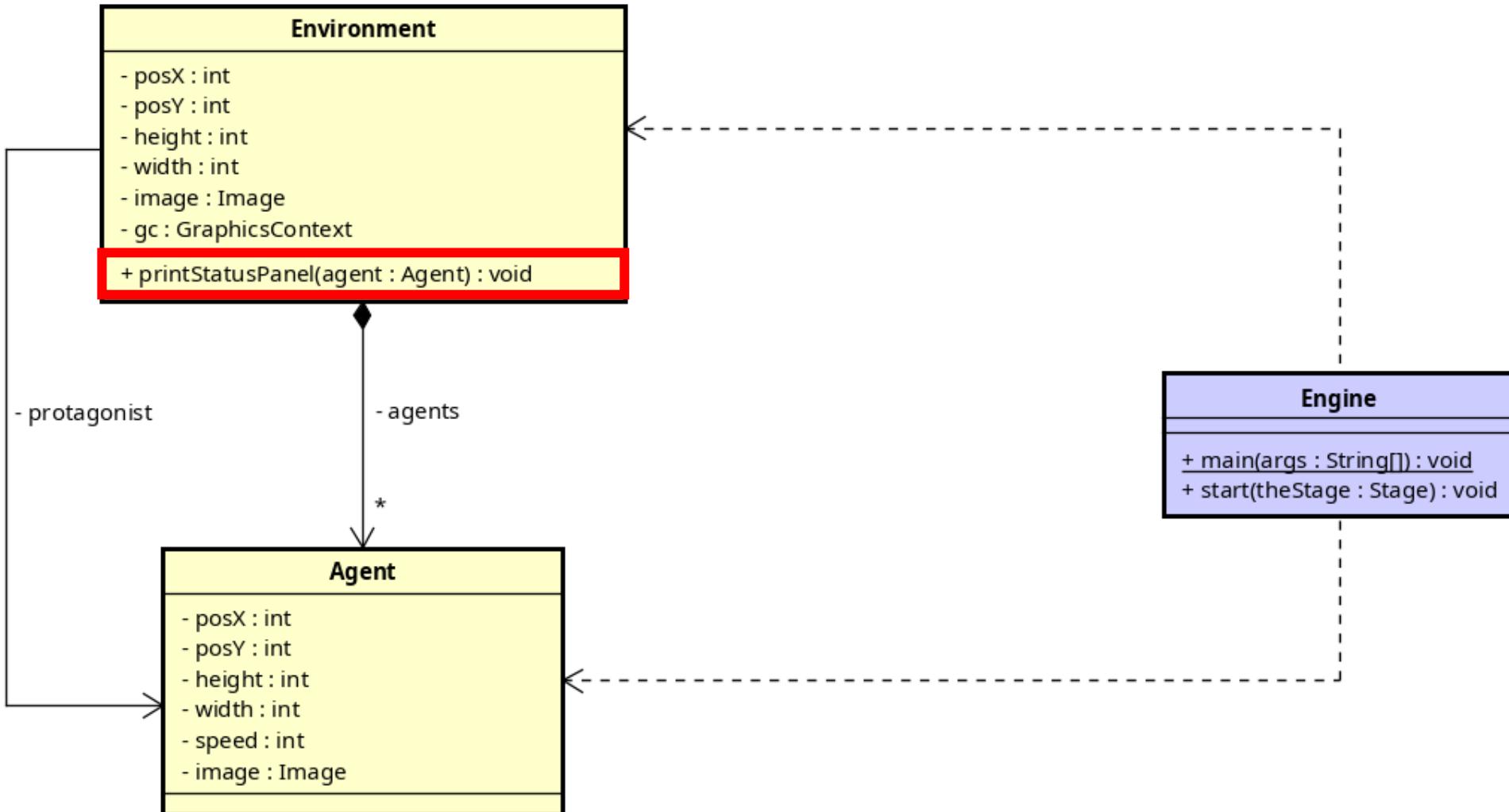


# The Class Diagram

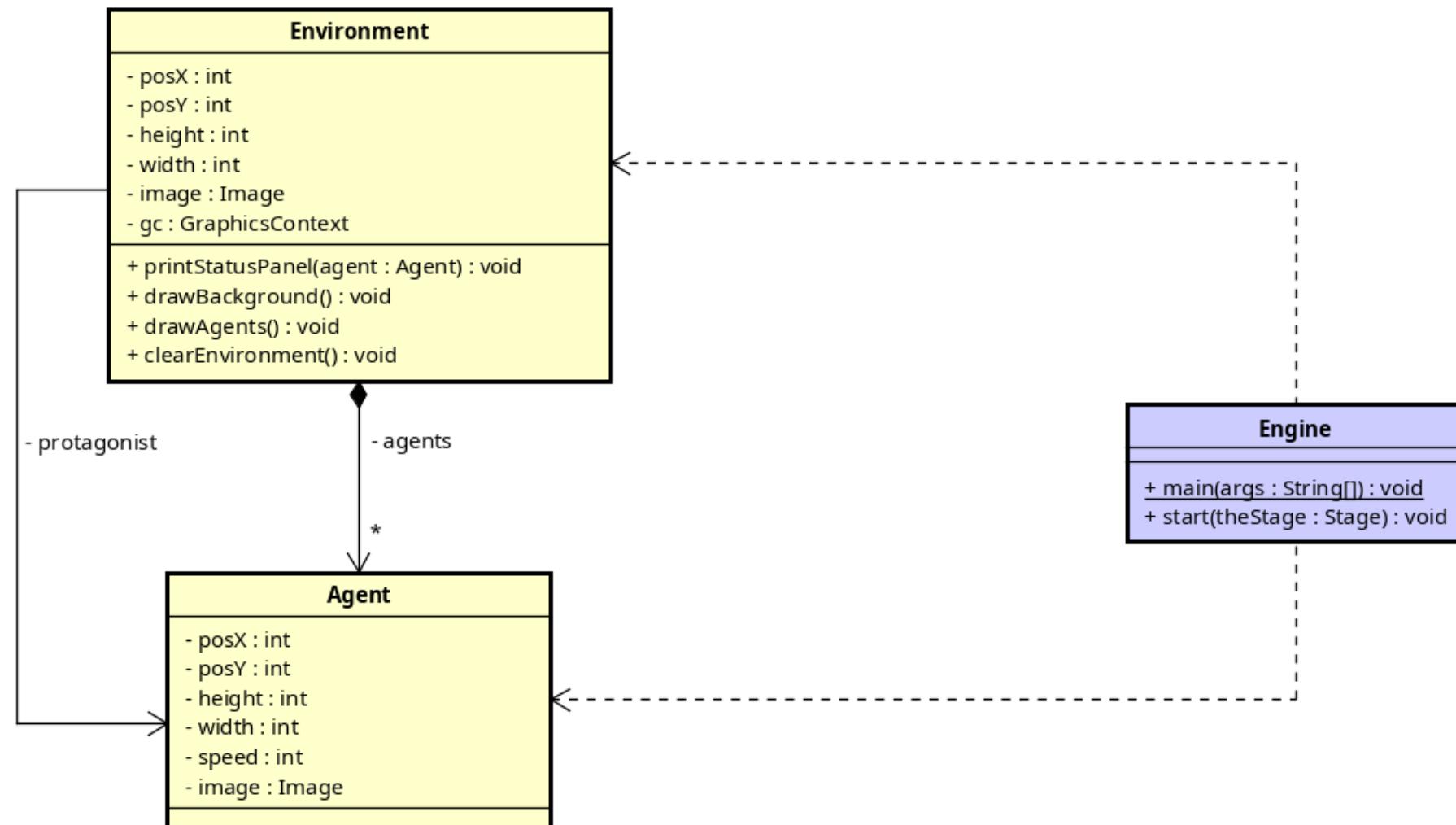


# The Class Diagram

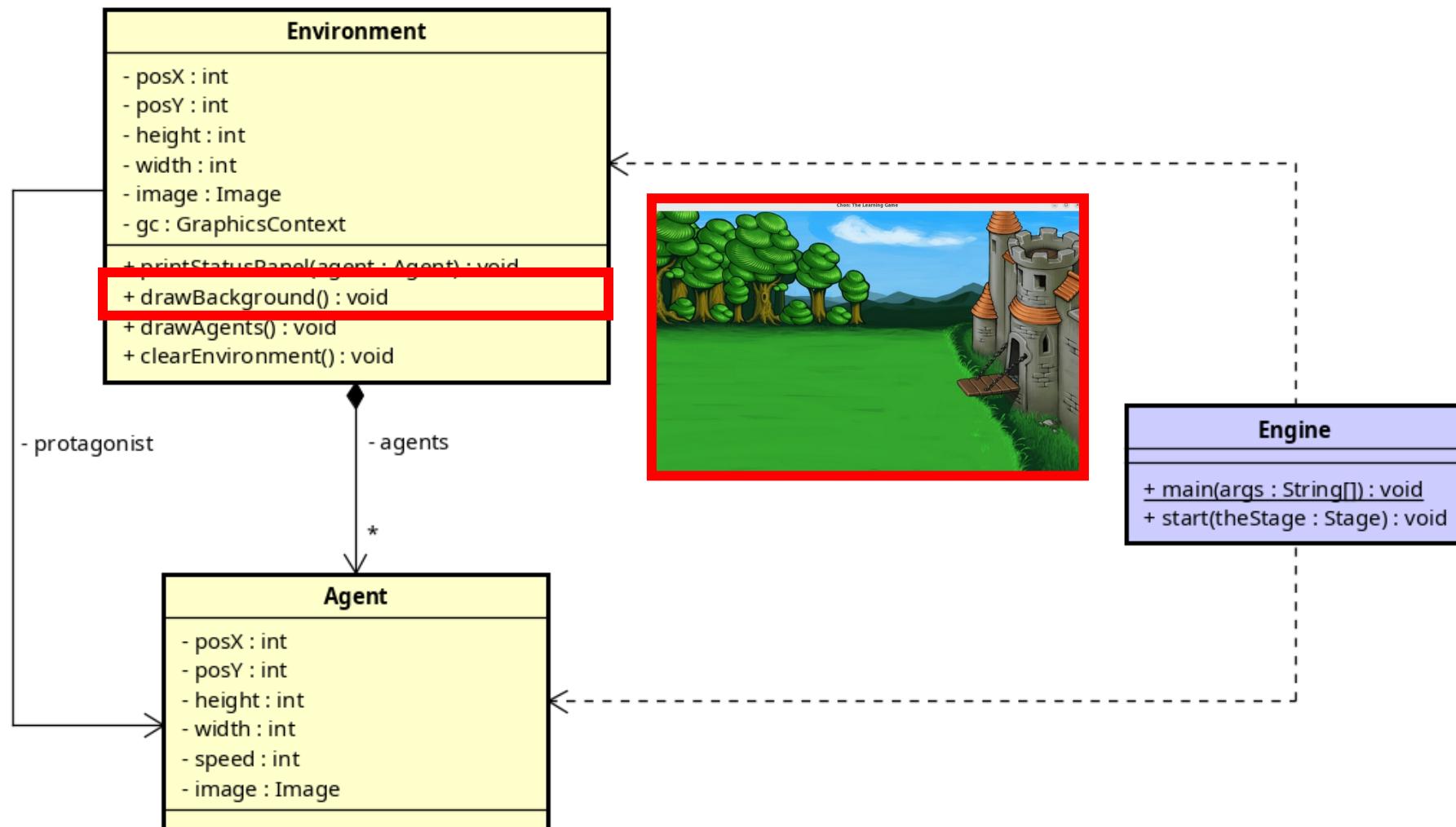
X: 400  
Y: 390



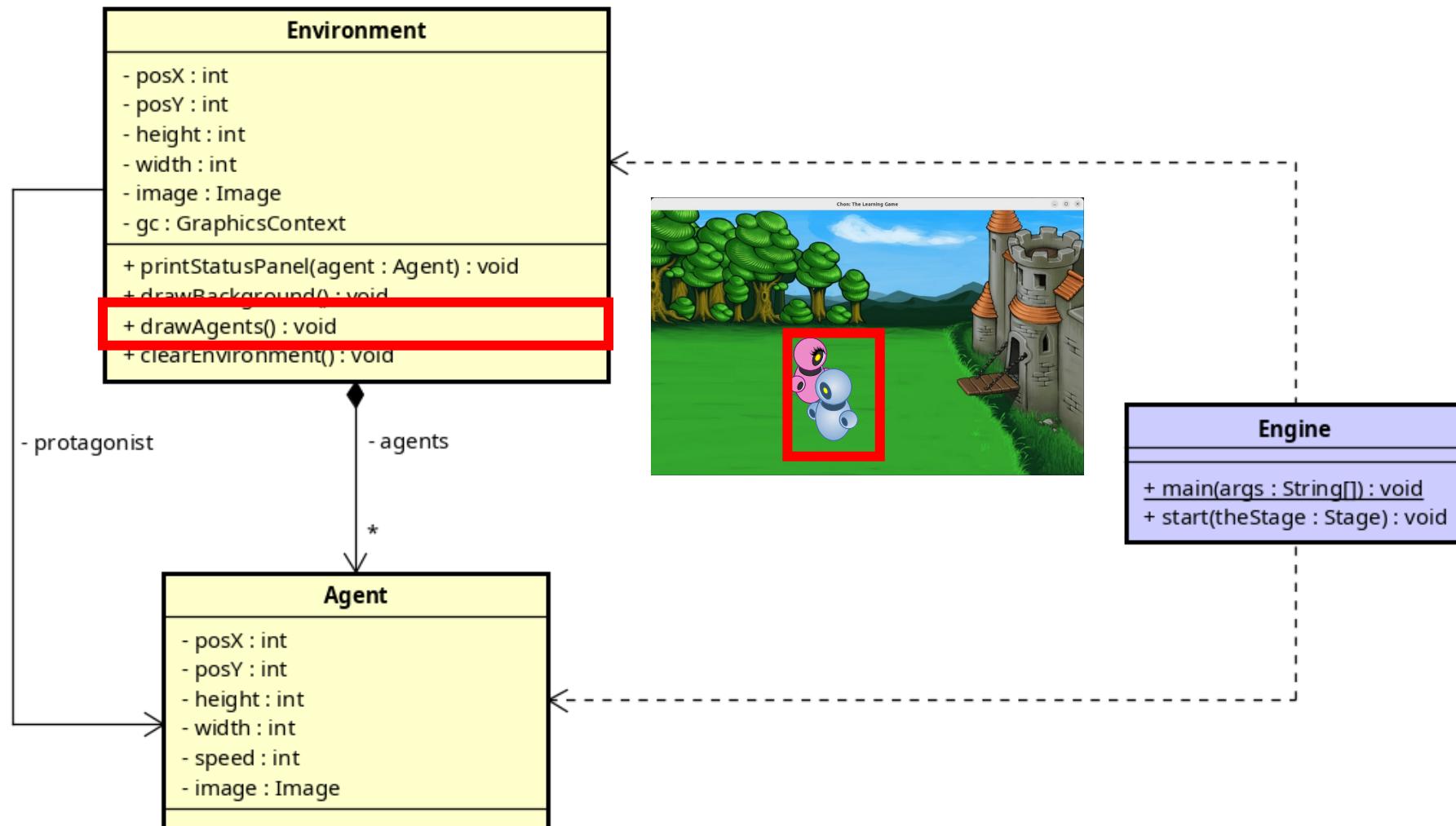
# The Class Diagram



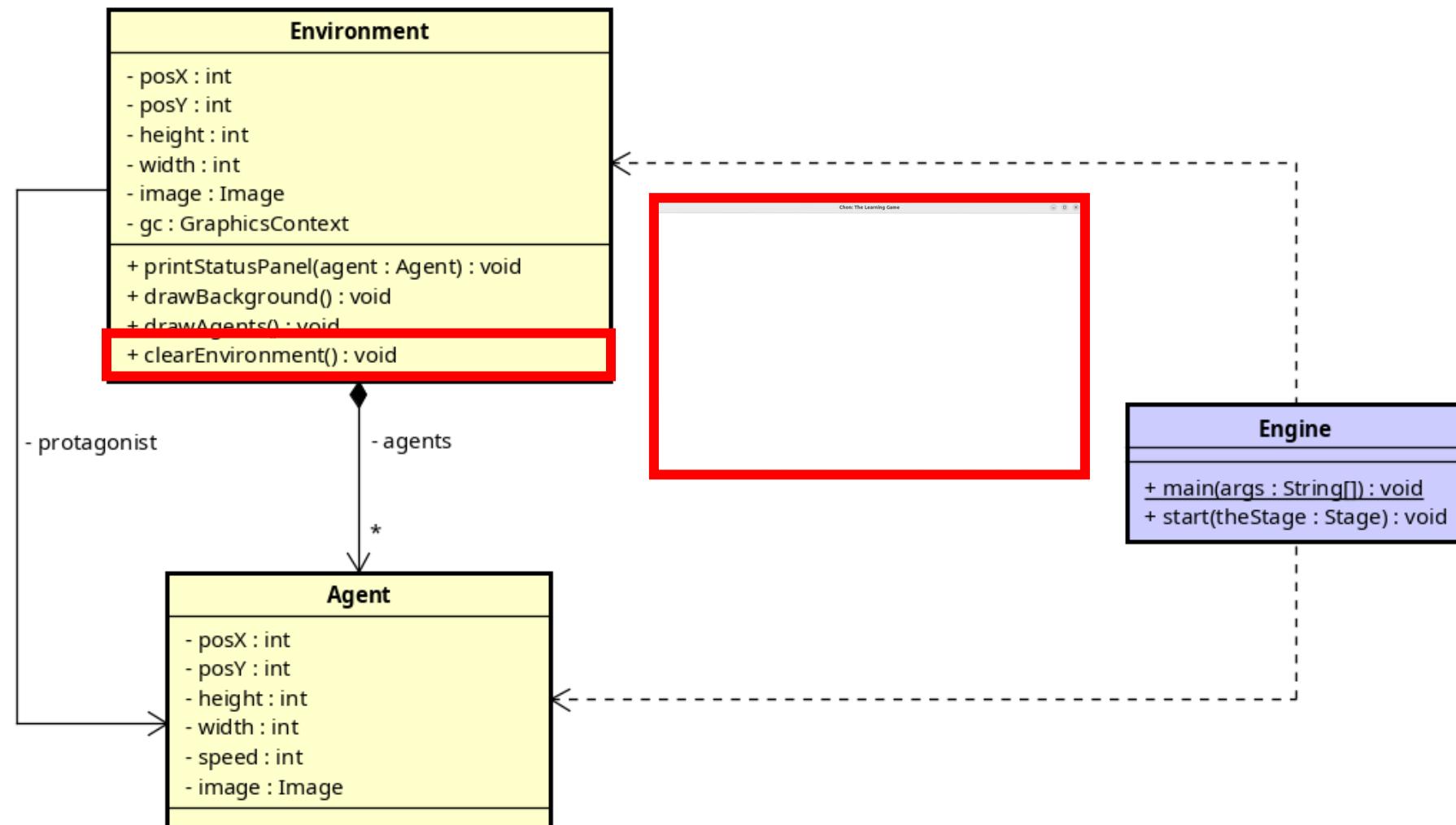
# The Class Diagram



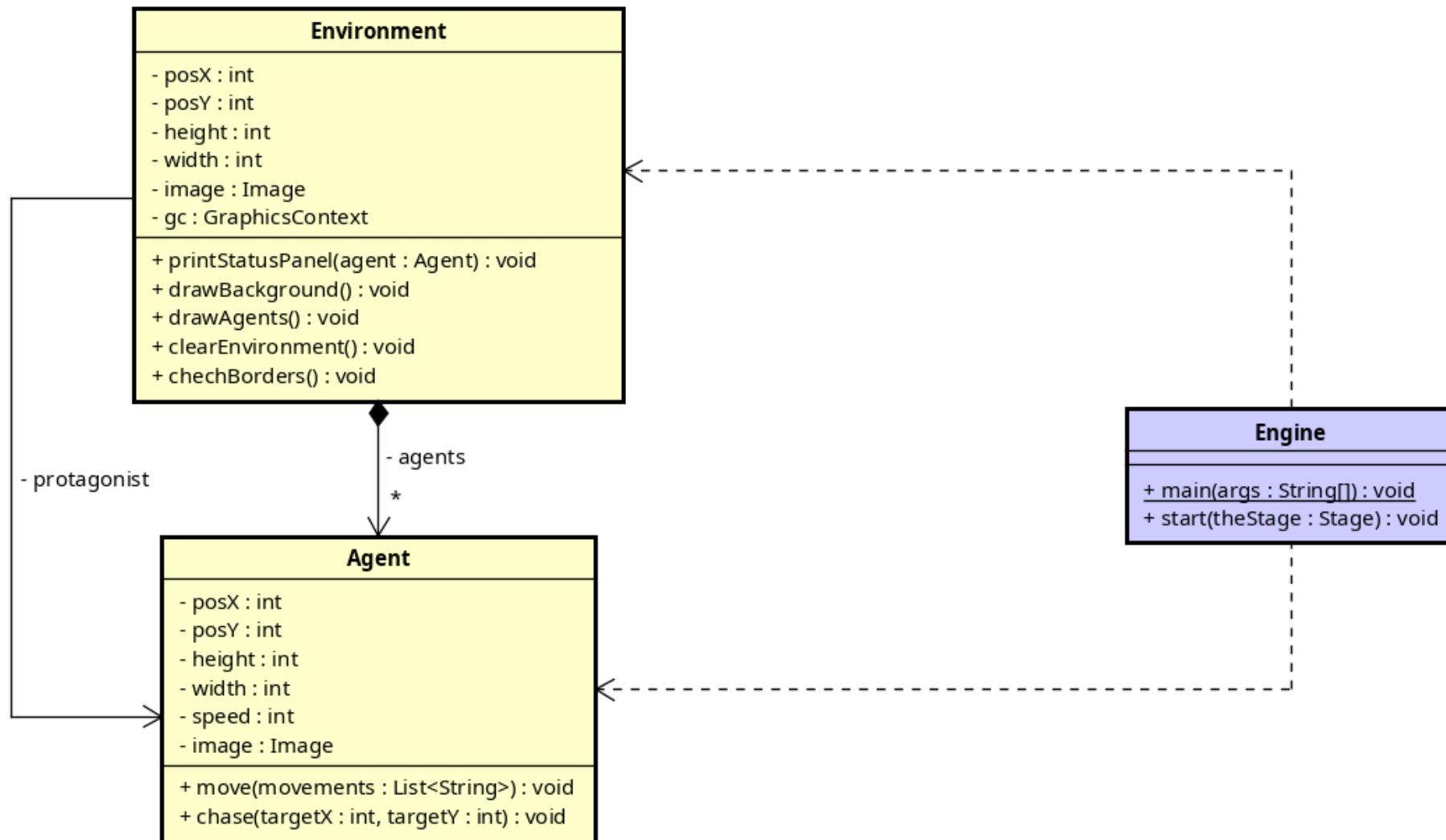
# The Class Diagram



# The Class Diagram



# The Class Diagram



# The Class Diagram

