Angry Birds Basic Game Playing Software

version1.2

By

XiaoYu (Gary) Ge, Stephen Gould, Jochen Renz

Sahan Abeyasinghe, Jim Keys, Kar-Wai Lim, Zain ul Hassan, Andrew Wang, Peng Zhang

Research School of Computer Science
The Australian National University

2013-07-20

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Installation

Install JAVA environment

The framework has been tested with the JAVA6 and above. To check your java version, you can enter java -version in your command window.

You can download java environment (JRE) from this link
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Install Chrome

Chrome can be obtained from http://www.google.com/intl/en/chrome/browser/

Install the Framework

Unzip the software package (e.g. versionX.zip), assuming the directory of the software folder is ./angrybirds/

You will have the following files/folders:

- The external folder contains all necessary libraries.
- The src folder contains all source files
- The Proxy folder: extension for interfacing with Chrome.
- The setup.ini file: the file used to specify the focus_point. (discussed in section <Other Issues>)

Install Chrome Plugin

1. Open Chrome
2. Go to: chrome://chrome/extensions/
3. Click the 'Developer mode' check box
4. Click the 'Load unpacked extension...' button
5. Browse for the 'GIExtension' folder (contains the 'manifest.json' file)

NOTE: if the screenshots are coming out as transparent PNG files do this:

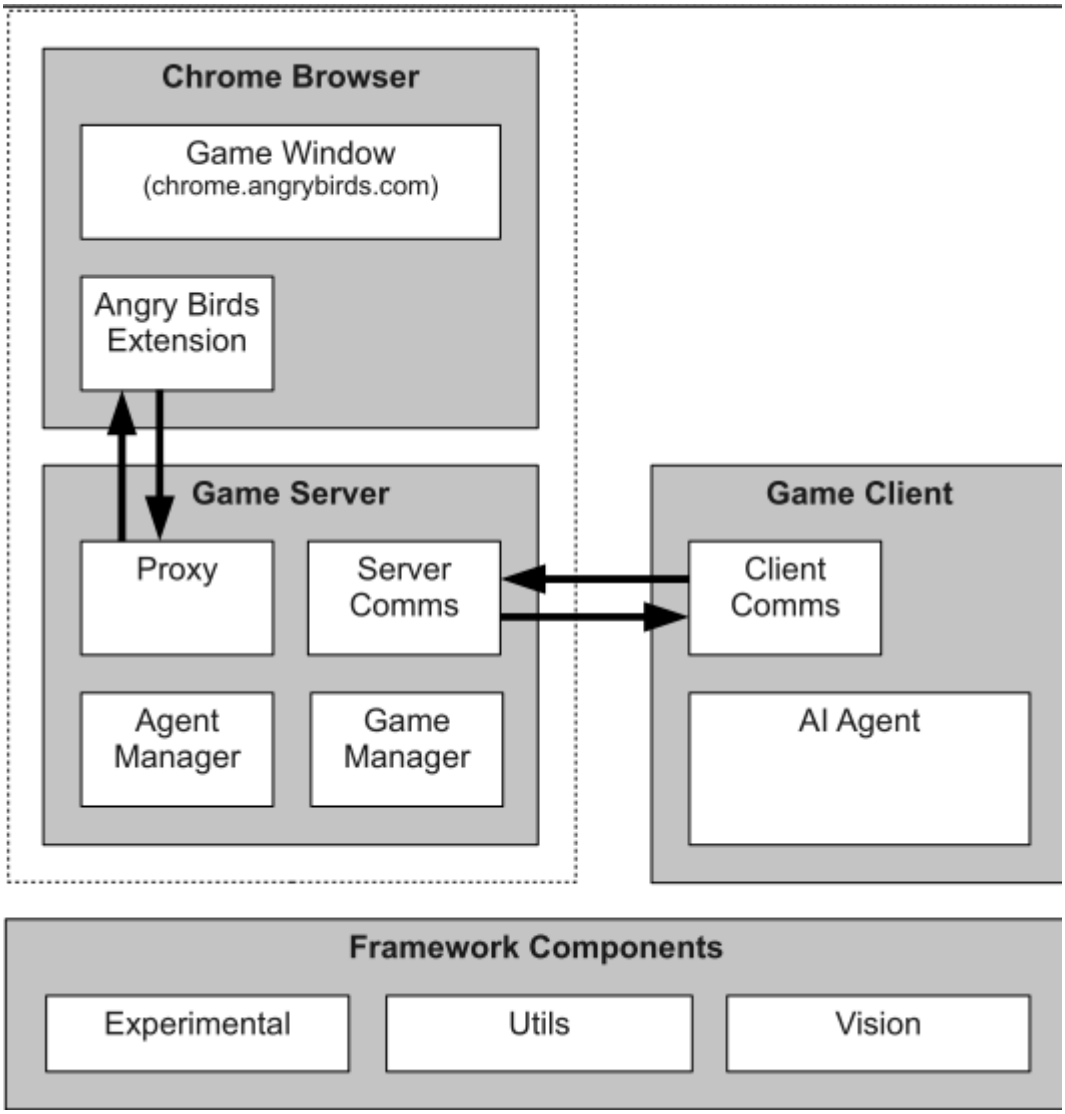
1. Open Chrome
2. Go to: chrome://flags/
3. Find the option 'Disable WebGL' and enable it (i.e. enable the disabling of WebGL)

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Run the Software

There are two commands available through the command line. You can run the naive agent (see section Run the Naive Agent) , view the segmentation of the game images in the real-time (see section Vision Module) and view the real-time trajectory output (see section Trajectory Module)

Architecture



This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Angry Birds Extension

The chrome plugin interacts with the chrome browser. It offers functionalities such as capturing game window, executing actions (e.g. move the mouse, click, and wheel to zoom). To install it, please see Install Chrome Plugin

AI Agent

An agent will use the vision module to analyze game scenarios and the trajectory module to plan on shots. Although the framework is written in java, the agent can be done in other languages such as c++, phython, etc.

Proxy

The server interacts with the angry birds extension via the proxy module. The following are the proxy messages:

- CLICK: left click of the mouse
- DRAG: drag the cursor from one place to another
- MOUSEWHEEL: scroll the mouse wheel
- SCREENSHOT: capture the current game window

The agent does not need to access the Proxy component.

S/C Communicating Port

Server/Client Communicating Port receives messages from agents and sends back feedbacks after the server executed the required actions indicated by the messages. The messages fall into the following three categories

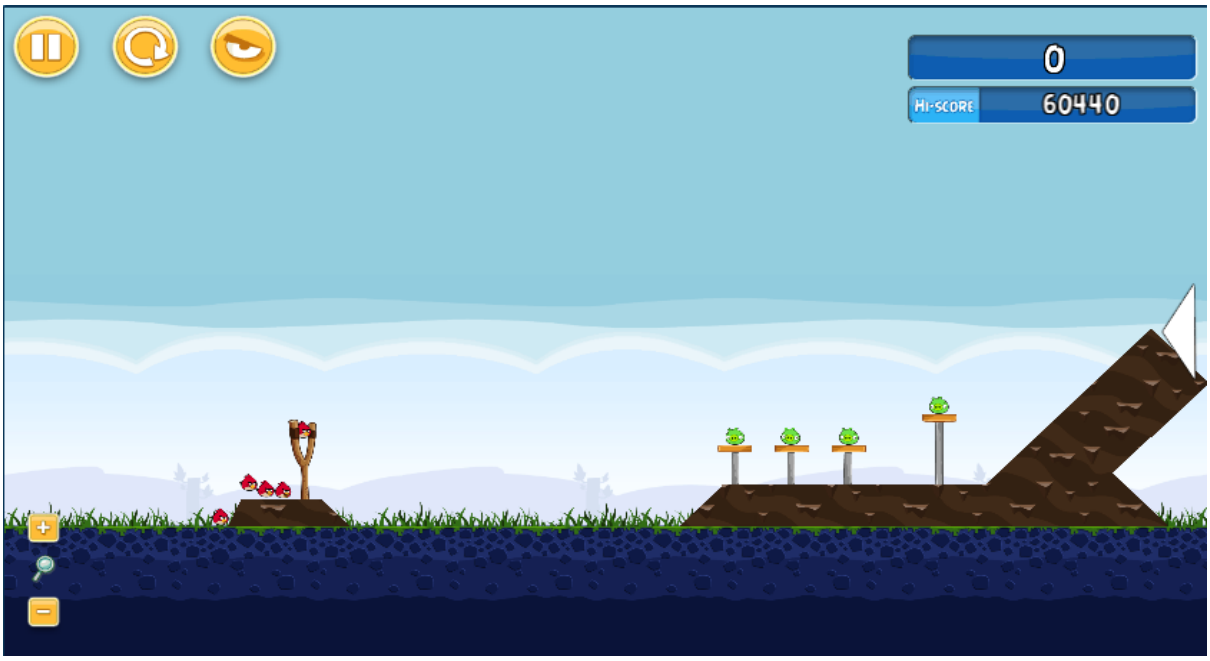
- Configuration messages
- Query messages
- In-Game action messages
- Level selection messages

The syntax of the messages can be found in /doc/ServerClientProtocols.pdf

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Vision Module

The vision module will segment an image and captures the minimum bounding rectangles of essential objects in the image. (The *essential objects includes {"Red Bird", "Wood", "Trajectory", "Tap", "Pig", "Ice", "Stone", "Slingshot", "Blue Bird", "Yellow Bird"};)* The following is an illustration of the output from the vision module.



To show the real-time segmentation, you can type and execute the command in the command window under the software directory.

java -jar Client.jar -showSeg

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Trajectory Module

The Trajectory Module estimates the trajectory that a bird will follow given a particular release point (relative to the slingshot). We use a constant velocity for every shot within the same level and use the release point and slingshot location to determine the launch angle *a*. From these we can compute the initial horizontal and vertical velocities as *v cos(a)* and *v sin(a)*, respectively.

With the initial velocities in hand, we use Netwon's classical laws of physics to estimate the parabolic path that the bird will take. The predictTrajectory function in the Trajectory Module returns a list of points that the bird will follow given a certain release point. To facilitate planning shots, the Trajectory Module provides a function (called estimateLaunchPoint) to estimate the release point given a desired target (for example, the centre of one of the detected pigs). The function uses the following equation to calculate the launch angle

$$\theta = \arctan\left(\frac{v^2 \pm \sqrt{v^4 - g(gx^2 + 2yv^2)}}{gx}\right)$$

Where (x, y) are the normalised coordinates of the target point relative to the sling (using sling size as the scale), and the gravity g is assumed to be 1 unit. Two angles are obtained from this equation and the function returns two corresponding launch points in an ArrayList of Points.

Note that when a shot is made, the actual launch angle *a* is always slightly different from the arctangent of the launch vector and the difference varies from level to level. Also, the normalised velocities are slight different between levels. The variation in angle and velocity are taken into account when predicting trajectories and estimating launch points. The trajectory module provides a method (adjustTrajectory) to adjust these two changes using information from a previous shot, so that the accuracy of the following shots is improved.

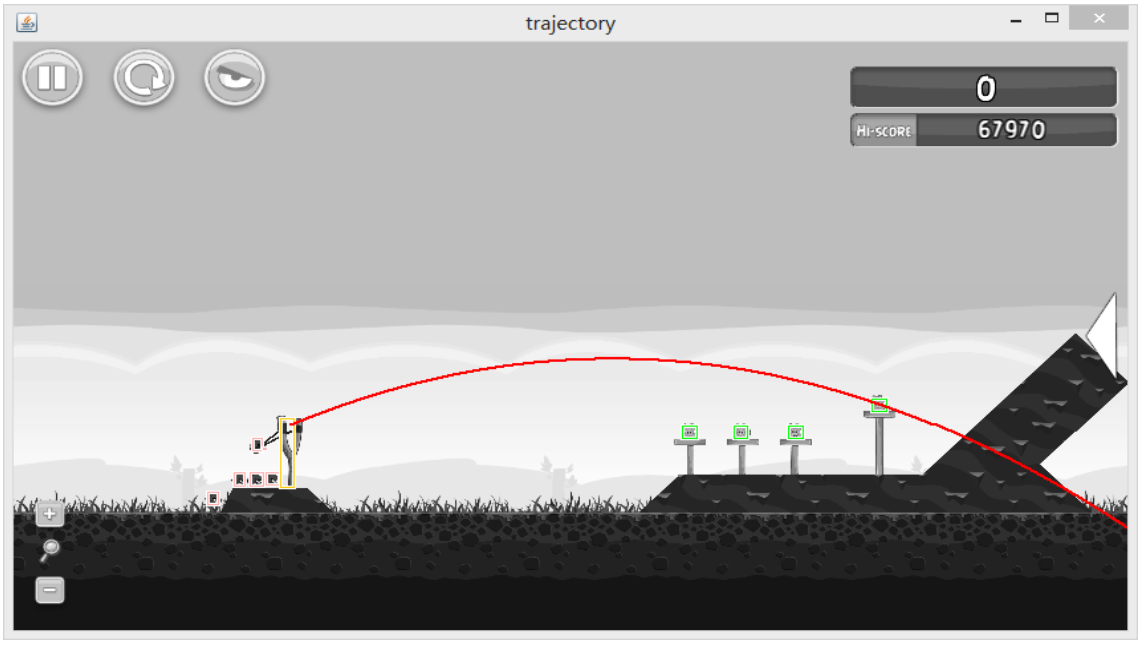
Also note that due to pixel level variations in the location of the slingshot and bird and effects due to scaling, the paths estimated by the Trajectory Module are only approximate. Agents should take this into account when planning their shots.

To show the real-time segmentation, you can type and execute the command in the command window under the software directory.

java -jar Client.jar -showTraj

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

The following is an example output



This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

The Naive Agent (standalone) – a demo

Naive Agent

The naive agent is created to demonstrate how to construct an agent on the basis of the provided modules, namely, the chrome plugin, the vision module, and the trajectory module. The naive agent is called “naive” because it shoots bird directly to the pig without reasoning.

The code can be found in /src/ab/demo/NaiveAgent.java

Play the Naive Agent

Load the Game Window

To load the game window

1. Open Chrome
2. Maximize the browser window
3. enter “http://chrome.angrybirds.com/” in the address bar, then press enter
4. Select **SD** version, then click Play
5. Select the episode POACHED EGGS and stay at the level selection page.

(the level selection page)

Please keep the tab of the game activated. You do not need to keep the browser window in front of the screen. i.e. you can minimize the window after loading the game, but ensure the tab remains selected.

Run the Naive Agent

Assuming the directory of the software is /angrybirds/versionX, to run the naive agent

1. Load the game in Chrome (see section Load the Game Window)
2. Open the shell.
3. Go to the software directory.

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

4. Execute the command **java -jar Client.jar -na [1-21]** to invoke the naive agent, the naive agent will start from the level you specified. The initial level is 1 by default.

   E.g. java -jar Client.jar -na 2 will start the naive agent in the level 2.

Create your own Intelligent Agent

Screen Shot and Segment an image

It is very easy to make a screen shot and segment it. The following is a code fragment showing the process:

```
//capture Image
 BufferedImage screenshot = doScreenShot();
// process image
 Vision vision = new Vision(screenshot);
 List<Rectangle> red_birds = vision.findRedBirds();
  List<Rectangle> blue_birds = vision.findBlueBirds();
  List<Rectangle> yellow_birds = vision.findYellowBirds();
```

Please refer to /src/ab/vision/Vision.java for details

Access the Game State

There are four main game states

1. "Won": a page shows the final grade and the stars you won.
2. "LOST" : a big pig tells you lost
3. "PLAYING": the game is not finished
4. "LEVEL SELECTION"： the level selection page.
5. "LOADING": the game is currently loading and a progress bar is showing on the page

Please refer to /src/ab/vision/GameStateExtractor.java for details

Trajectory Module

Once you know your target, you need the trajectory module to plan on the shots. To get the set of estimated release points given a target, you can call the estimateLauchPoint(Rectangle, Point) method. This method requires a bounding box of the slingshot and the target point

```
//Initial a trajectory planner
TrajectoryPlanner  tp = new TrajectoryPlanner();
//Get the estimated points
ArrayList<Point> pts = tp.estimateLaunchPoint(sling, _tpt);
```

Please refer to /src/ab/planner/TrajectoryPlanner.java for details

Execute Shooting Actions

You have two ways to shoot a sequence of birds. A SHOT contains six fields (x,y,dx,dy,t_shot,t_tap) The first four parameters specify a move from (x,y) to (dx,dy). T_shot is the time at which a bird is

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

supposed to be launched. T_tap specifies the gap between the releasing time and the corresponding tapping time. Time are counted in milliseconds

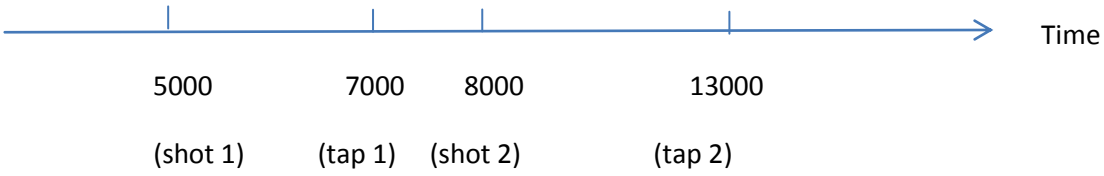
You can submit a list of shots to the server and the server will execute shooting accordingly.

Self-Manage mode

Let's say you want to shoot 2 birds and the following are corresponding shot instances:

1) Shot (x,y,dx,dy, 5000,2000);
2) Shot (x,y,dx,dy, 8000,5000);

You can pass the two shots via ClientShootCmd to the server, and the server will shoot the bird following the timeline below:



Note: t_tap specifies the gap between the releasing the time and the corresponding tapping time. When you specify the shooting time, please beware that:

1) Avoid executing a shoot and tap actions at the same time. E.g. The following are two conflicting shots:

Shot (x,y,dx,dy, 5000,3000); Shot (x,y,dx,dy, 8000,5000);

2) In the game, there is a gap between two shots. You cannot shoot a bird immediately after one shot. The gap is around 5 seconds. So please adjust your t_shot and ensure the gap between two shots more than 5000.

Auto-Arrange mode

We provide another convenient way for shooting, if you do not care about the exact time. We can arrange the shooting time for you. By setting the t_shot field of each shot in the shot sequence to 0, the server will shoot the bird every 5 seconds. For example,

1) Shot (x,y,dx,dy, 0,3000);
2) Shot (x,y,dx,dy, 0,5000);

The first shot will be made at time 0, and second shot will be made at time 5000. Besides, the first tap happens at time 3000 + 0 = 3000 and the second tap occurs at time 5000 + 5000 = 10000.

Note, if you want the server to arrange the shooting for you, you need to ensure **ALL** the t_shot fields are set to 0 in a shooting sequence.

The server will treat the following conflicting shooting sequence in the self-manage mode

1) Shot (x, y,dx,dy, 0,3000);

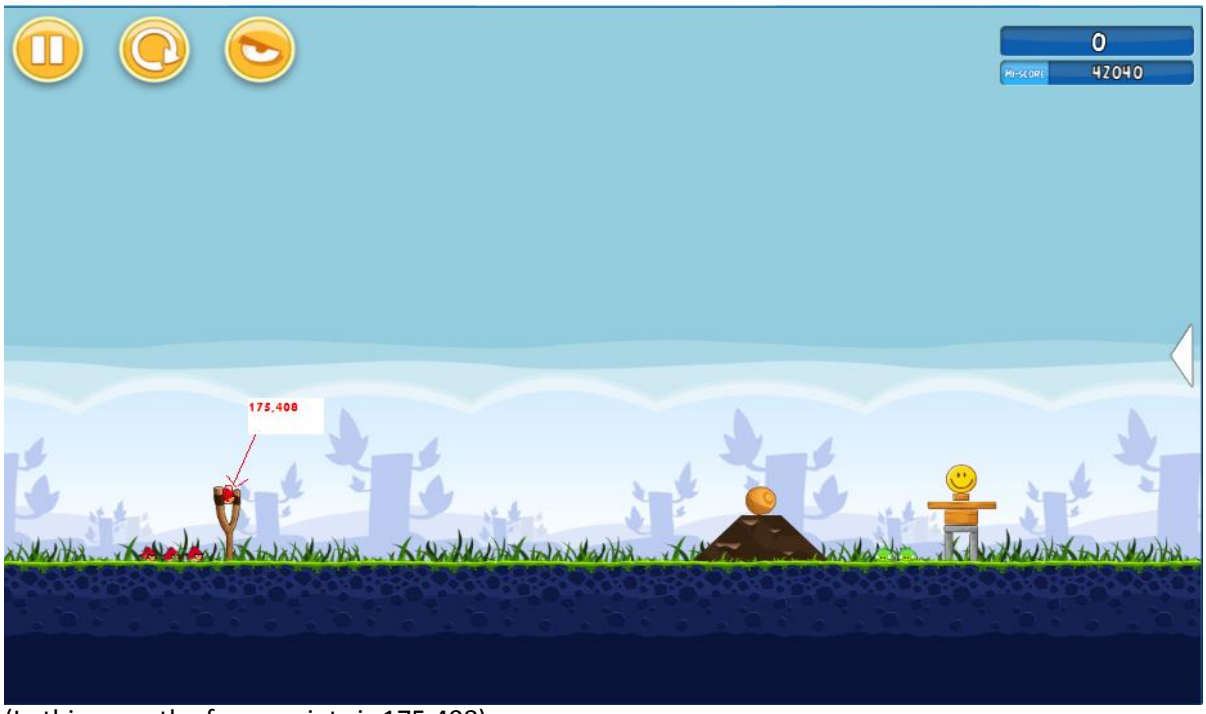
This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

2) Shot (x,y,dx,dy, 0,5000);
3) Shot (x,y,dx,dy, 10000,5000);

Focus Point

The naive agent shoots a bird by first moving the mouse to the focus point (see Collect Trajectories), and then drag the bird to the release point, and release the mouse. A focus point is the middle of the top edge of the slingshot's bounding box. In automatic mode where the agent detects the focus point automatically, the right focus point depends on the correct detection of slingshot. Sometimes, the agent tries to execute shooting action, but the bird is not launched, it is likely due to the wrong setting of the focus point. To solve this, please set the focus point manually in **setup.ini** file by uncommenting # symbol in front of the entry focus_point., and adjust the value manually according to the actual position of the point on the image. E.g.

"focus_pt:274,528,1" means use (274,528) as the focus_point in the level 1.



(In this case, the focus points is 175,408)

Then the agent will use the specified point as focus_point in the corresponding level.

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent

Compile using ANT

You can use ANT to compile the source codes via command line. Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. For more information, please refer to http://ant.apache.org/

You can download ANT from http://ant.apache.org/manualdownload.cgi

ANT 1.7 or above required

To compile the source codes, go to the software directory, and use command

ant compile

To generate an executable file

ant jar

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and can be accessed via https://github.com/aibirds/agent