About angry birds AI competition:

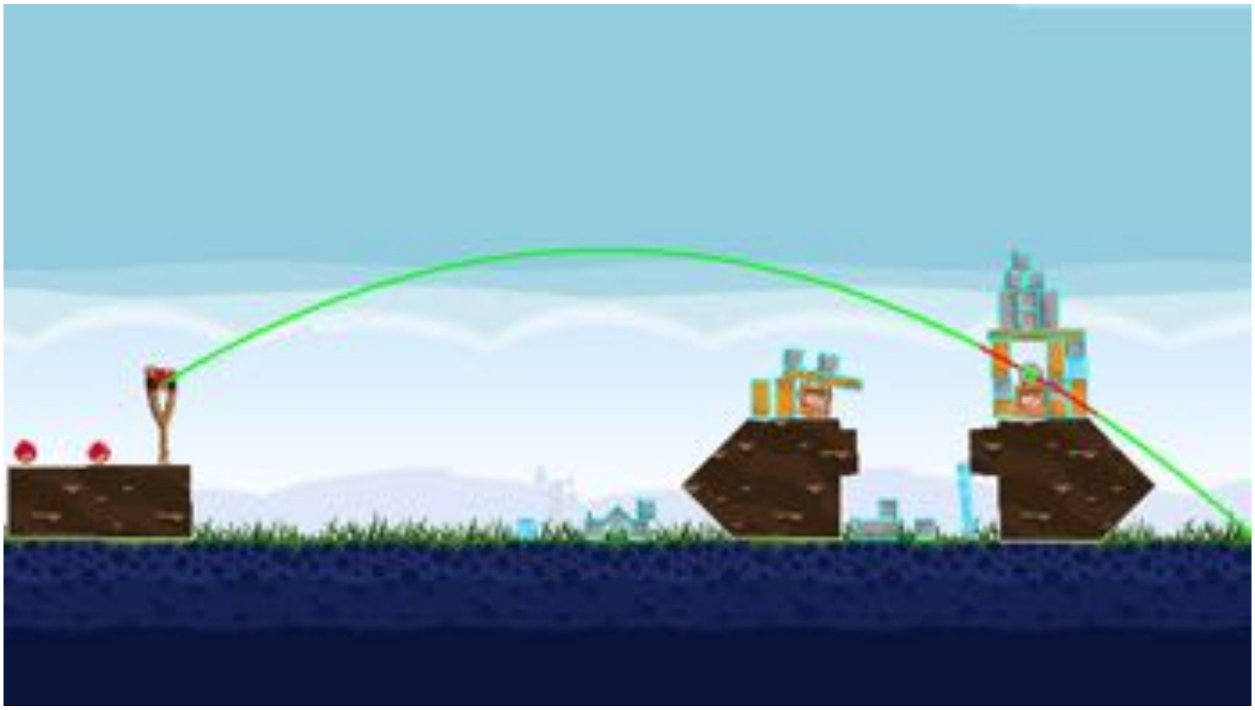
Angry Birds is a popular video game where the task is to shoot birds with different properties from a slingshot at a structure that houses pigs, with an aim to destroy all the pigs. Thought it may sound simple, the structure can be very complicated and can involve a number of different object categories, each with different properties. However, the game and the structure largely observes the laws of physics, hence to a large extent it is possible to infer how the structure will act or change when hit at a certain position.

The task of the **Angry Birds AI Competition** was to develop an **intelligent Angry Birds playing agent** that is able to successfully play the game autonomously, without human intervention. The long term goal however, is to build AI agents that can play new levels better than the best human players. This may require analyzing the structure of the objects and to infer how to shoot the birds in order to destroy the pigs and to score most points. In order to successfully solve this challenge, participants can benefit from combining different areas of AI such as computer vision, knowledge representation and reasoning, planning, heuristic search and machine learning. Successfully integrating methods from these areas is one of the great challenges of AI.

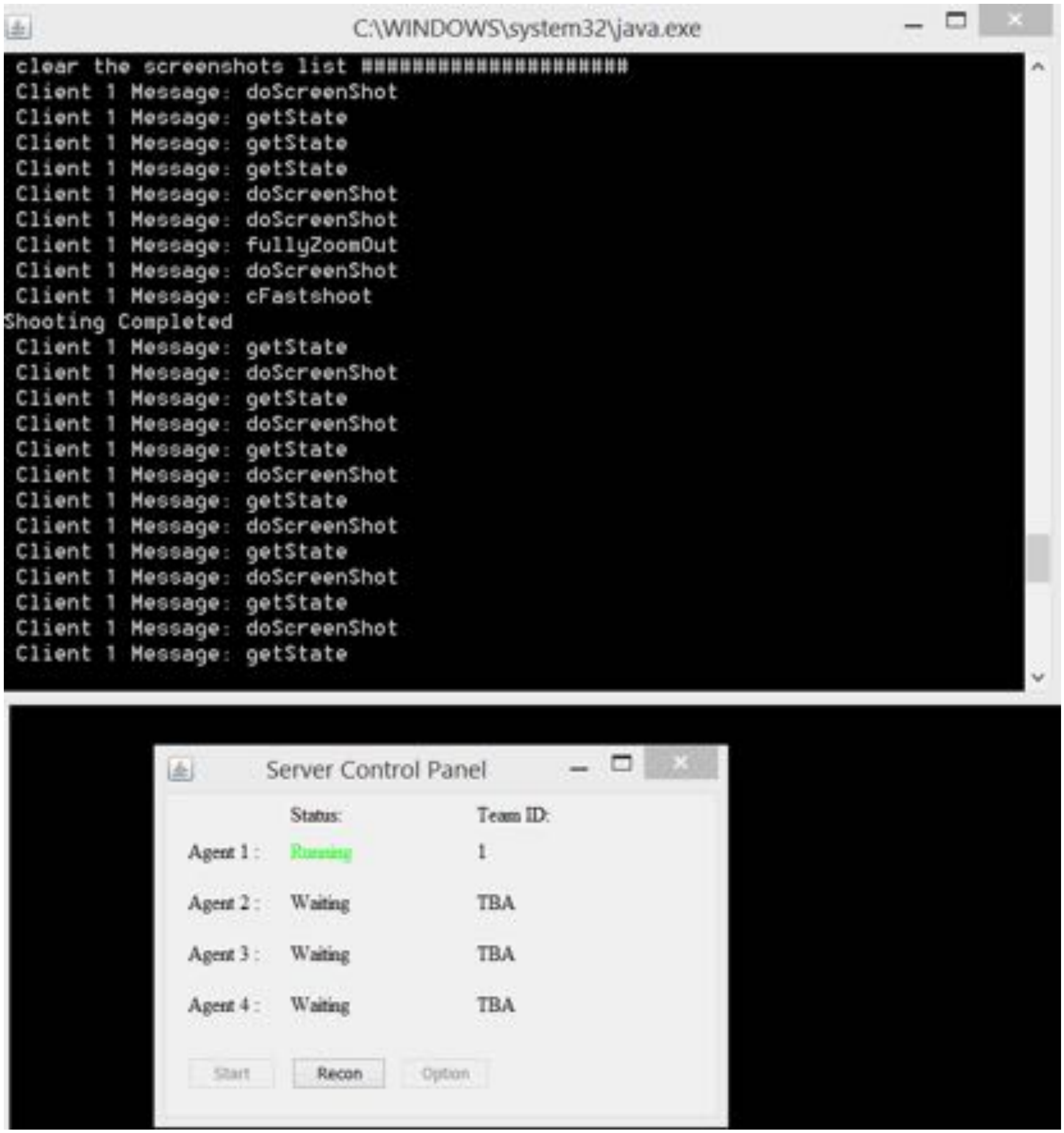
Since it cannot be expected that all participants can develop all these capabilities themselves, the organizers provide a **basic game playing software** that is implemented using Java and includes the following components:

- A **computer vision component** that can analyse a video game frame and identifies the location, category and bounding box of all relevant objects plus the game score.

- A **trajectory component** that calculates trajectories of birds and computes where to shoot from in order to hit a given location.



- A **game playing component** that executes actions and captures screen shots.



Participants are free to use these components or can develop their own components. One must however note that there is a small amount of uncertainty in the output that the supplied components produce and must take this into account when developing their programs.

The **ECAI 2014 Angry Birds AI Competition** is designed to test the abilities of Angry Birds playing agents on a variety of Angry Birds levels. The competition was run using a client/server architecture, where the game server runs an instance of **Angry Birds Chrome** game for each participating Angry Birds agent. The agents run on a client computer and communicate with the server via a given protocol that allows agents to obtain screenshots of their game window from the server at any time. Agents can also obtain the current competition high scores for each level from the server. In return, agents send the server their shooting actions (release coordinate and tap time) which the server will then execute in the corresponding game window.

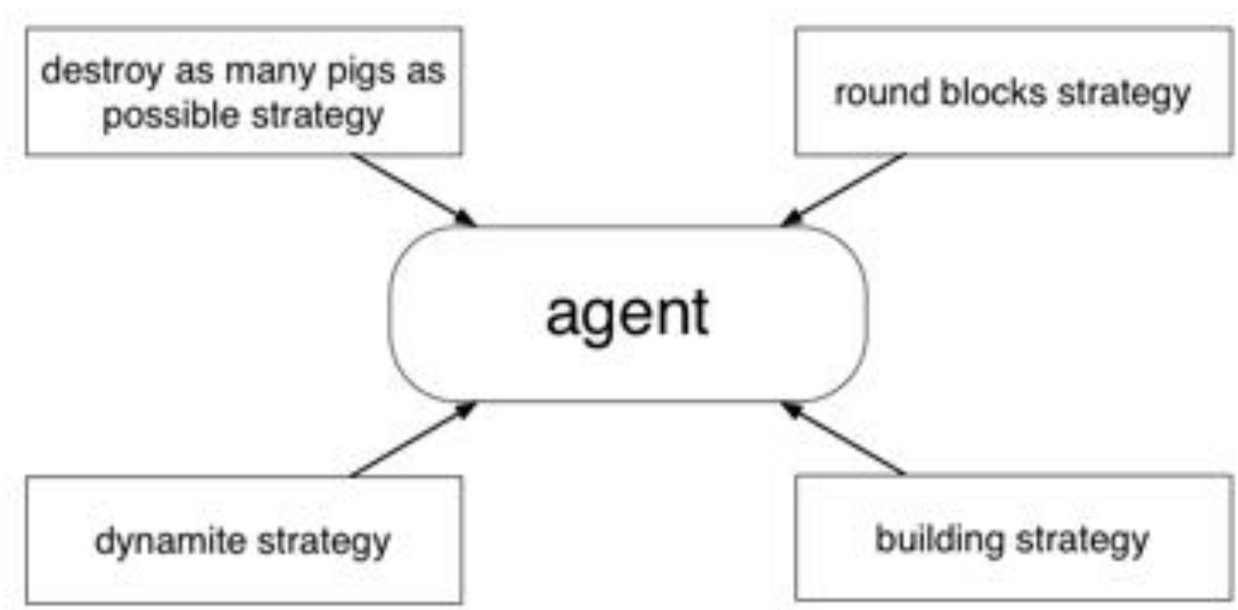
The grand final is time constrained; ie., the qualifying participant's agents are expected to play a given set of Angry Birds levels automatically, without any human intervention, within the given time frame. All levels can be accessed at any time. The agent to achieve the highest combined game score over all levels would be regarded as the **AI Birds Champion 2014**.

Note that the actual game levels used during the competition are not disclosed to the participants in advance. Nonetheless, participants are informed about the birds and the object categories used in the competition game levels, so that their behaviour can be anticipated.

Our solution:

Successful strategies for different levels are based on very different approaches. However, we believe that we are able to cover majority of levels with a relatively small set of strategies. We used a planning agent that decides which strategy to play for the current move considering the environment (blocks configuration, reachable targets), possible trajectories, the bird currently present on the sling and the birds available on the stage.

The agent always plans only one move. It decides about what move to take based on the estimated utility of each strategy. The strategy with maximum utility is then played.




10580
68600


16080
68600

Dynamite strategy

The dynamite strategy tries to aim at the TNT only if there is a pig nearby. We found out that hitting the dynamite is not very useful on its own. Its position must be taken into account. The closer the dynamite is to a pig, the more damage it will inevitably cause. Another important finding is that the dynamite's shock wave will cause the other objects to move around with great force, thus causing greater damage. We count that the dynamite is even more useful when there are a lot of stone objects and other TNTs within a given pixel range. For example, in the picture below, the dynamite has a medium utility as there is one pig in close proximity and two small stone objects.

Building strategy

The building strategy finds a connected block structure near the pigs. The building strategy addresses the problem of hidden pigs as they often reside in one of these structures. It is not an easy task to get to the pigs through the structures as sometimes one move will transform the structure into an even better bunker for the pig. Therefore, we had to take only the moves which we know will get us directly to the pigs. We decided to achieve this by choosing only the suitable blocks.

The block inside a given building is considered suitable to be the target object if it satisfies all of the following conditions:

- is reachable,
- is flat or straight,
- has at least two supporters,
- is not high,
- is not a square.

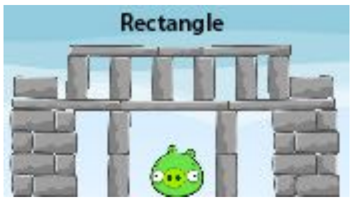
The red-outlined blocks in the picture below fulfill the given conditions.

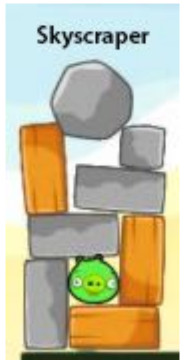
All the selected blocks are then sorted based on their type and relative position in the building. The best block for a given bird on sling is then selected.

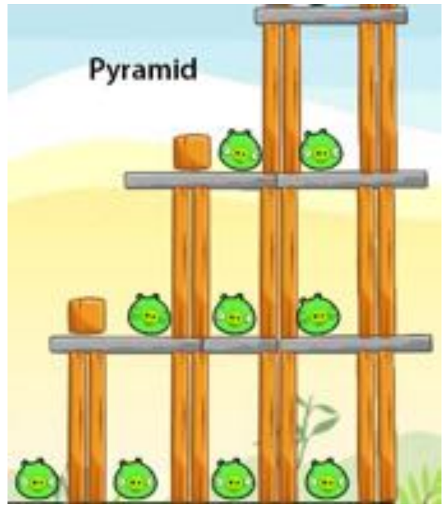
We also distinguish between three types of buildings:

Pyramid,
Rectangle,
Skyscraper.

These three different kinds of structures, also have three different aiming policies. In the pyramid, the goal is to shoot as low as possible with the intention to destroy the target block as then the building collapses. Whereas in the skyscraper policy, the bird tries to move the whole structure and knock it down. Therefore the agent aims at the top. With the rectangle policy, the agent combines the two policies described above. Therefore the bird tries to aim both at the top trying to turn the building over, or at the bottom trying to destroy the whole structure so that it corrupts. The final decision is then made based on the type of the target block.







Round objects strategy

Often, there are levels which are unsolvable without an insight into what will happen after the bird hits the target object. The most prominent example of this is the round block strategy. The round block strategy tries to either hit the block directly or release the block from a shelter. The former one's objective is to push the killer object (most commonly stone) so it kills everything that is in its way until it loses all its energy. However, there is also the delicate strategy of releasing the round object from a shelter. In often cases, there is no need to pass the object the bird's kinetic energy because the killer object has enough potential energy on its own. The only thing that needs to be done is to release it.

Destroy as many pigs as possible strategy

The destroy as many pigs as possible strategy does exactly what the label says. We set this strategy as our default one as it gets right down to what the overall goal of the game is. It tries to find a trajectory with as many pigs in the way as possible. Interestingly, it is also good at going around the obstacles as it tries to find the easiest way to the target.

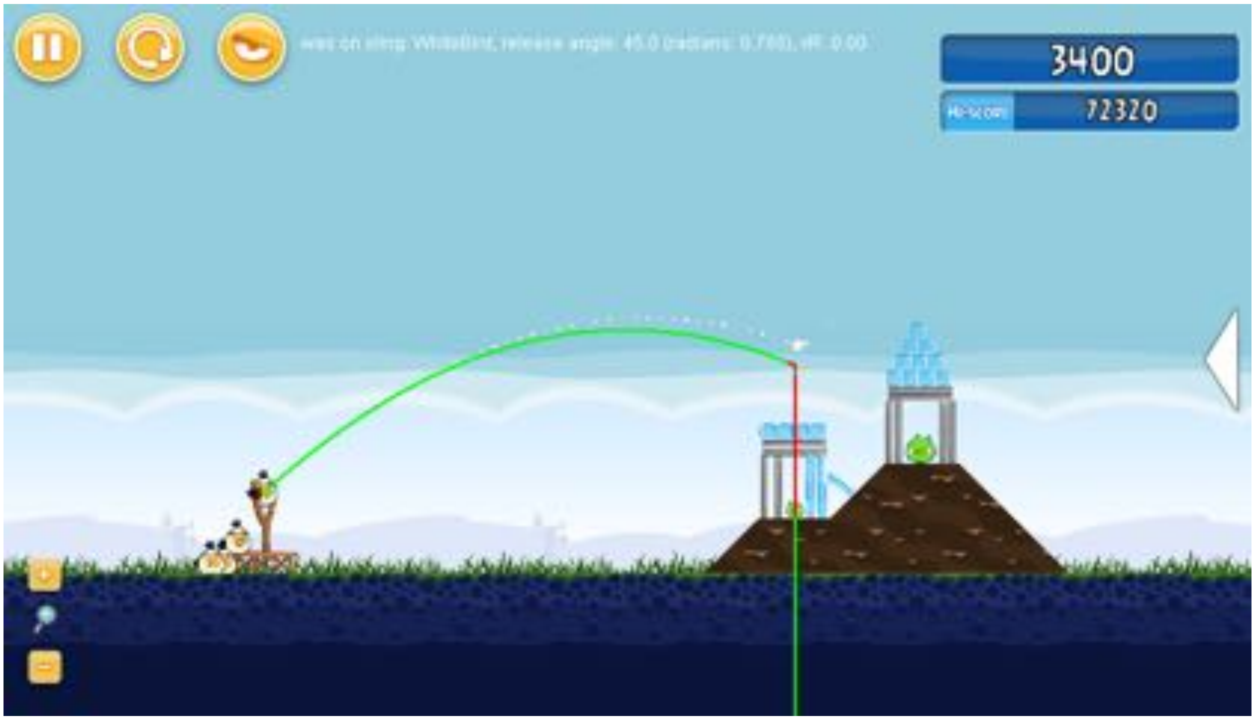
Tapping time

In the game, there are also different types of birds which have different kinds of powers. The powers are released when the bird is tapped in the air. Our approach was to distinguish between these powers and find the right strategy for each bird. We noticed that if the bird hits an object before we even tap it, then the power is lost. The tapping time is therefore computed based on the first point the bird hits an obstacle in the computed trajectory. For example, the blue and yellow birds are tapped on a fixed fifteen per cent of their trajectory before hitting an object. Interestingly, it is better when the black bird is tapped 500 ms after it hits its target object.

White bird

The white bird is completely different from the rest of the birds because upon tapping it releases an egg and immediately flies away without continuing in its trajectory. That’s why, the most important thing with the white bird is to successfully land the egg on the target object as the bird in itself is not so strong.

The white bird is therefore moved above from the target object so long as it hits something in the trajectory. Then the bird's egg is released from above so that it hits the desired target object.



Trajectories & trajectory utilities

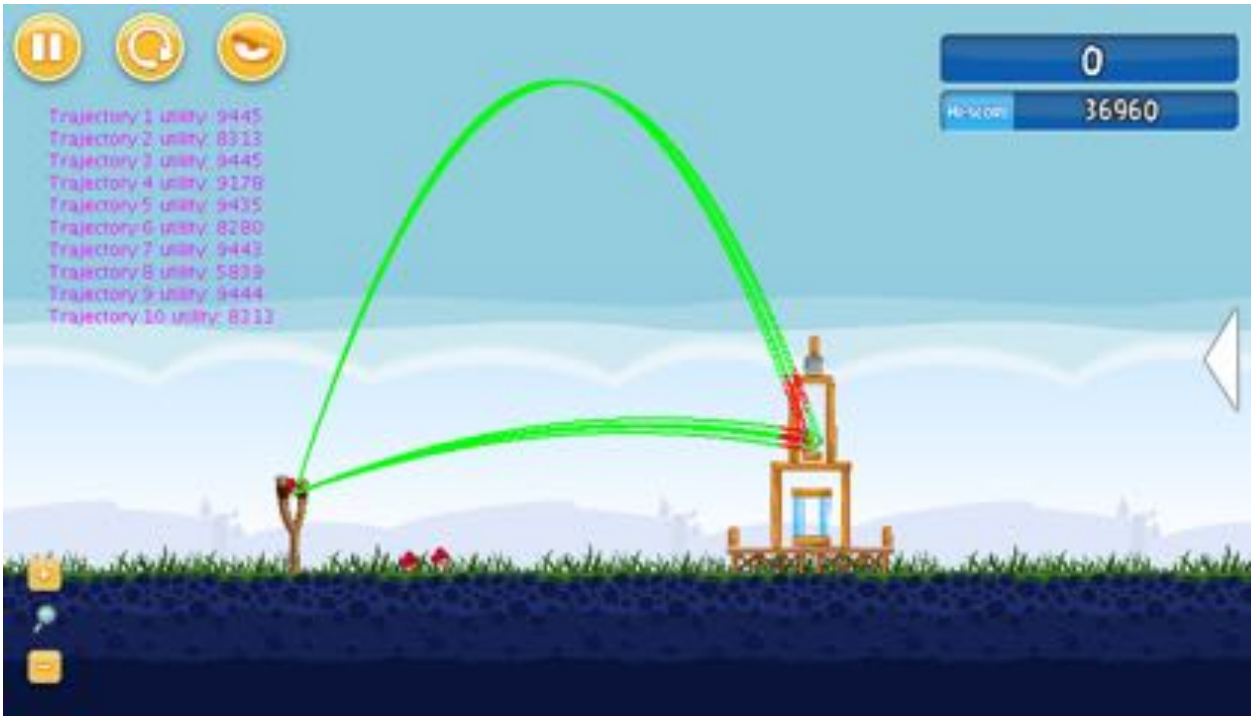
In the game, there are different materials and different birds and not every bird goes well with every material. Having identified this, we decided to distinguish between trajectories by introducing utilities. Each bird has a different utility for each type of block. In a nutshell, the collision utilities of each block in the way with respect to the bird are summed and the trajectory which has the least utility is then chosen. The collision blocks that are further away have also greater utility compared to the closer ones as they don't cause the bird to deviate from the course.

For instance, the blue bird is much better at destroying ice than the yellow or red one.

For the geeks out there, the formal equation of the trajectory utility is as folllows:

$$trajectoryUtility = \sum_{l \in B} \left(\frac{d(t,l)^{1.4}}{\alpha} + 1 \right) * u(r,l), \quad (1)$$

where **B** is a set of game blocks **l** in the planned trajectory, **d(t,l)** is a distance between the target object **t** and a block in the way, **alpha** is a normalization constant, **u(r,l)** is a function, which returns utility based on a bird **r** and a type of a game block in the way.



Game Playing Additions

The organizers provide the participant teams with a prepared, working framework which could be used. However, we decided to make some improvements because some of the components were not flawless.

One of the components we improved was the trajectory planning which is now more precise and works with more release angles and respective velocities than before.

Another problem was that the organizers recommended the teams to use only secure shots which wait 15 seconds after each shot is made no matter what the outcome of the shot is. They thought we would not be able to deduce from the picture when the right time to shoot again is. Our agent uses only fast shoots (which include no waiting) as we are able to deduce from the screenshot if anything moves in the scene. This is done by comparing two consecutive screenshots and pinning out some important block, pig, sling and hills information which is converted into a number. These two consecutive numbers have to be the same.

For the purpose of scene evaluation, we have implemented new methods which can easily infer the relative position of a block towards other blocks and also if two blocks touch. This is also used in the building and round object strategies.
Another important component of the trajectory planning is the collision marker. It goes through every pixel of the trajectory and tells if the particular bird hits an object within its radius. We added the radius for each bird as not all birds are the same size and this improves the precision of the utility evaluation as most of the time there are only pixel differences between hitting an object and not.

Results:

The **Angry Birds AI Competition** attracted a total of 11 participant teams from around the world! However, participating for the first time, team **DataLab Birds** from the **Czech Technical University in Prague** became the **AIBIRDS 2014 Champion** with their grand final score being 406340, almost double of the runner up team, also a new participant, **AngryBER** from the **University of Ioannina in Greece** which scored 243880. Team **PlanA+** from **Sejong University in South Korea**, who already participated last year, stood third scoring 206620.

Since the unmodified winner from last year did not even qualify for the quarter finals, It was evident that this year the agents were much better, and the progress was highly commendable!

The qualification day was in itself very exciting. However, this was overshadowed by the first quarter final, when the home team **DataLab Birds** were well ahead of the other three teams. In the final moment when the tension was building up, to everyone's amazement, team **Angry BER** solved another level and moved team **Impact Vector** to third place. Such excitement!

The second quarter final was a brighter assessment, with the two teams **PlanA+** and **IHSEV** clearly ahead of the other two teams. The semi final was dramatic! **Angry BER** and **PlanA+** were within 60 points ahead on the leaderboard when team **DataLab Birds** solved a level shortly before the finishing time and took the lead. Resulting in **PlanA+** standing third with the tiniest possible margin. Interestingly, they solved 4 levels as compared to only 3 levels solved by the runner up team **Angry BER**.

The grand final was a clear decision. Team **DataLab Birds** moved ahead after a short while and never appeared vulnerable of losing it. They managed to solve 6 out of the 8 levels. Given the increased difficulty of the levels, with the most difficult ones in the grand final, this was an impressive achievement.

Here are the scores of the different rounds:

Results after Q2			**Quarter Final 1**	
1. DataLab Birds	423280		1. DataLab Birds	346260
2. PlanA+	372810		2. Angry BER	224860
3. s-Birds Avengers	361770		3. Impact Vactor	173710
4. Angry Dragons	317300		4. s-birds Avengers	167860
(5. Naïve	302710)			
6. Impact Vactor	298390			
7. Angry-Hex	294170		**Quarter Final 2**	
8. IHSEV	292380			
9. Angry BER	253820		1. PlanA+	360920
10. BeauRivage	238080		2. IHSEV	277530
11. RMIT Redbacks	188890		3. Angry-HEX	129610
12. Auto Lilienthal	0		4. Angry Dragons	78970
Semi Final			**Grand Final**	
1. DataLab Birds	232790		1. DataLab Birds	406340

2. Angry BER	206680		2. Angry BER	243880
3. PlanA+	206620			
4. IHSEV	93100			

About team Datalab Birds:

The FIT CTU team got together as a part of the Summer Camp organized by eClub in association with the Data Science Laboratory. The Summer Camp welcomed individuals with an interest in big data, mobile applications, start ups and to participate in the Angry Birds AI Competition. Karel and Radim joined the Summer Camp so as to participate in the Competition. Having completed their course on the Fundamentals of Artificial Intelligence, they thought it would be a great opportunity to apply their acquired knowledge in practice. Tomáš Borovička who ideated the inclusion of the Angry Birds AI competition in the Summer Camp mentored the team and guided them in the right direction during the whole development process. The Summer Camp was supported by both academic experts and industrial partners; Seznam, Samsung, CTU Medialab, FIT, FEL, CIIRC or IAESTE.

Tomas Borovicka is a PhD candidate and researcher with the Faculty of Information Technology at the Czech Technical University in Prague. Tomas obtained his Bachelor degree at the Faculty of Electrical Engineering and Master degree at the Faculty of Information Technology. His scope is Artificial Intelligence and his research explores areas of machine learning and data mining; of late, his main focus has been semi-supervised learning and active learning. Furthermore, Tomas has over seven years experience with software development, following his prior employment as a Software Engineer. Currently he holds the position of Research Scientist in Lely Industries, a Dutch company focused on dairy robotic farming and innovations in agriculture. His passion lies in the application of latest research results and approaches in practice.

Radim Špetlík is a bachelor student of the Faculty of Information Technologies at the Czech Technical University. Moreover, he is in the final year of his study in Liberal Arts and Humanities at Faculty of Humanities, Charles University. His achievements so far stand evidence to his excellence. He won the first place in the Artificial Intelligence Fundamentals course competition in 2013/2014 focused on Genetic Algorithm and he is also a member of the winning team of the ECAI 2014 Angry Birds AI Competition. His interests are Artificial Intelligence & machine learning.

Karel Rymeš is a bachelor student at the Faculty of Information Technology at the Czech Technical University in Prague. Karel will enter his second year of study in September 2014. He is an enthusiastic apprentice and has been a member of the winning team of the ECAI 2014 Angry Birds AI Competition. His main interest lies in the sphere of Artificial Intelligence; his other interests include, programming C/C++, English, cycling and running.