Deceptive Angry Birds: Towards Smarter Game-Playing Agents

Matthew Stephenson
Research School of Computer Science
Australian National University
Canberra, Australia
matthew.stephenson@anu.edu.au

Jochen Renz
Research School of Computer Science
Australian National University
Canberra, Australia
jochen.renz@anu.edu.au

ABSTRACT

Over the past few years the Angry Birds AI competition has been held in an attempt to develop intelligent agents that can successfully and efficiently solve levels for the video game Angry Birds. Many different agents and strategies have been proposed to solve the complex and challenging physical reasoning problems associated with such a game. The performance of these agents has increased significantly over the competition's lifetime thanks to the different approaches and improved techniques employed. However, there still exist key flaws within the designs of these agents that can often lead them to make illogical or very poor choices. Most of the current approaches try to identify the best or a good next shot, but do not attempt to plan an effective sequence of shots. While this might be due to the difficulty in predicting the exact outcome of a shot, this capability is precisely what is needed to succeed, both in games like Angry Birds, but also in the real world where physical reasoning capabilities are essential. In order to encourage development of such techniques, we can create levels where selecting a seemingly good next shot will lead to a worse outcome. In this paper we present several categories of deception to fool the current state-of-the-art agents. By evaluating the performance of the most recent Angry Birds agents on specific level examples that contain these deceptive elements, we can show how certain AI techniques can be tricked or exploited. We also propose some ways that future agents could help deal with these deceptive levels to increase their overall performance and generality.

CCS CONCEPTS

• **Computing methodologies** → *Artificial intelligence*; • **Applied computing** → Computer games;

KEYWORDS

Angry Birds, Agents, Physics-Based games, Video games, Deception

ACM Reference Format:
Matthew Stephenson and Jochen Renz. 2018. Deceptive Angry Birds: Towards Smarter Game-Playing Agents. In *Foundations of Digital Games 2018 (FDG18), August 7–10, 2018, Malmö, Sweden.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3235765.3235775

1 INTRODUCTION

The creation of an intelligent agent that can reason and predict the outcome of actions in a physical simulation environment, typically with inaccurate information, is a key subject of investigation in the field of AI. It is particularly important for the development of such agents to integrate the areas of computer vision, machine learning, knowledge representation and reasoning, planning, and reasoning under uncertainty. The Angry Birds AI (AIBirds) competition was created as a means to promote the research and creation of these agents through the use of the physics-based simulation game Angry Birds [9]. This type of physical reasoning problem is very different to traditional games as the attributes and parameters of various objects are often imprecise or unknown, meaning that it is very difficult to accurately predict the outcome of any action taken [11]. Many of the previous agents that have participated in this competition employ a variety of techniques, including qualitative reasoning [16], internal simulation analysis [8, 12], logic programming [5], heuristics [6], Bayesian inferences [7, 15], and structural analysis [17]. Some of these approaches are faster, whilst others may be more consistent or adapt better to new scenarios.

However, even with all these advancements in the development of Angry Birds agents there are still key weaknesses with the approaches and designs used. As Angry Birds is an incredibly complex puzzle game, it is impossible to hand code solutions for every possible level that an agent could be given. As a result of this, agents will often make assumptions or generalisations about how levels are solved which may prove to be incorrect. By creating levels that exploit an agent's pre-defined strategies we can deceive it into making poor shot decisions. Understanding why certain agents can be fooled by certain types of deception will allow future agents to perform better and avoid these deception pitfalls. Physical simulation games such as Angry Birds provide a large and varied range of challenging levels [10], and as such we attempt to classify common categories where the solution requires creative reasoning in order to solve it. This is by no means an exhaustive set, but we believe it encompasses the main types of deception that an Angry Birds level could pose to an agent. To prevent re-treading already covered ground, we do not consider levels that only require what we would term intuitive approaches to solve them, such as aiming directly at the most pigs or targeting structure weak points, but instead focus on solution approaches that the majority of current Angry Birds agents are not capable of achieving. The reasoning required to solve levels with these deceptive elements should be difficult for agents but simple and understandable to human players.

The remainder of this paper is organized as follows: Section 2 describes the Angry Birds game and the AIBirds competition framework; Section 3 discusses the agents that will be examined

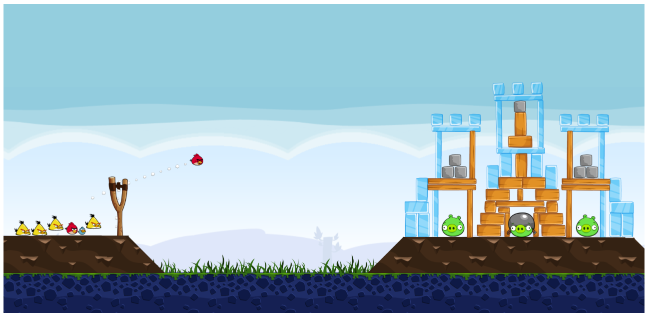


Figure 1: Screenshot of a level from the Angry Birds game.

and analysed; Section 4 categorises the types of deception that Angry Birds levels could contain; Section 5 details the experimental process and provides a summative description of the results; Section 6 discusses why certain agents and their approaches performed the way they did for each type of deception, and proposes several future possibilities.

2 BACKGROUND

2.1 Angry Birds Game

Angry Birds is a popular physics-based puzzle game in which the player uses a slingshot to shoot birds at pigs, with structures composed of blocks and other physical objects protecting them, see Figure 1. The goal of each level is to kill all the pigs using a set number of birds provided. All objects within the level have properties such as location, size, mass, friction, density, etc., and obey simplified physics principles defined within the game's engine. Blocks are also made of one of three materials, wood, stone or ice. Different bird types are available with different properties, and pigs are killed once they take enough damage from either the birds directly or by being hit with another object. The player can choose the angle and speed with which to fire a bird from the slingshot, as well as a tap time for when to activate the bird's special ability if it has one, but cannot alter the ordering of the birds or affect the level in any other way. The difficulty of this game comes from predicting the physical consequences of actions taken, and accurately planning a sequence of shots that will result in success. Points are awarded to the player once the level is solved based on the number of birds remaining and the total amount of damage caused.

2.2 AIBirds Competition

In this competition, agents are tasked with playing a set number of unknown Angry Birds levels within a given time, attempting to score as many points as possible in each level. The exact location and parameters of certain objects, as well as the current internal state of the game, are not directly accessible. Instead, information about the level is provided using a computer vision module, effectively meaning that an agent gets exactly that same input as a human player. Agents are required to solve these levels in real-time and can attempt levels in any order and as many times as they like. Once the time limit has expired the maximum scores that an agent achieved for each solved level are summed up to give its final score. Agents are then ranked based on this value and after several rounds of elimination a winner is declared. The eventual goal of this competition is to design AI agents that can play new levels as well as or better than human players.

2.3 Deception

The idea of deceptive problems for agents in video games is a relatively new area of research, although some prior work has been carried out. Most notably in a recent paper exploring the effect of deceptive games on general video game AI (GVGAI) agents [4]. These agents are designed to play previously unknown games, whilst attempting to maximise their total score in a set time period. Agents have full access to both the current game state and a forward model for determining the result of any action taken. These agents are usually heavily reliant on each game's scoring system to help guide their expected reward function towards desirable actions. This allows for the creation of levels that can exploit this forward model to lead agents towards making sub-optimal decisions (i.e. high short-term reward but small long-term reward).

Agents designed for playing Angry Birds do not have the luxury of a forward model, and so the types of deception offered in this environment are substantially different from those of GVGAI. Whilst Angry Birds does contain a scoring system, points are only awarded to an agent after it completes a level, making it difficult for agents to judge the success of specific actions. Most agents simply treat killing pigs as a positive outcome and using birds as a negative, something that we will exploit later on in some of our deceptive levels. Due to the fact that points are only awarded upon completion of a level, we only consider whether an agent was able to solve a level or not, rather than the points it achieved for doing so. Within this paper, the concept of deception can simply be taken to mean a particular feature or quality of a level that can cause agents (or players) to make poor actions (shots) by exploiting their specific biases or limitations.

It is important to make clear the distinction between deception and difficulty. Increasing the size or complexity of a level, such as having more pigs, birds or structures, may make the level more difficult and require more time to solve it, but does not necessarily make the level any more deceptive. However, changing the underlying strategies and approaches that are needed to solve a level could be considered a different form of deception. Levels that contain deceptive elements are designed to deliberately exploit pre-defined agent strategies, which prevents or highly impairs their ability to solve the level.

3 AGENT DISCUSSION

Our analysis will involve investigating the twelve agents that participated in the 2017 and/or 2016 AIBirds competitions. Whilst there have been over 30 different agents that have participated in the AIBirds competition over the years, the agents from the most recent competitions represent the best that are currently available. A brief description of each of these agents is given below, with full details available on the AIBirds website [3] and in the following papers [13, 14].

3.0.1 Naive Agent. The Naive agent is provided to all competition entrants as a useful starting point upon which to create their own agent. It fires the currently selected bird at a randomly chosen pig using either a low or high trajectory (also chosen at random). No other objects apart from the current bird and pigs are used when determining a suitable shot, and tap times are fixed for each bird based on the total length of its trajectory. It can therefore make shot calculations quickly and accurately but is by far the least sophisticated of the agents.

3.0.2 Datalab Agent. The Datalab agent uses a combination of four different strategies when attempting to solve a level. These can be described as the destroy pigs (kill most pigs), building (destroy blocks protecting or supporting pigs), dynamite (target TNT boxes) and round blocks (target round blocks or blocks which support them) strategies. The decision of which strategy to use is based on the environment, possible trajectories, currently selected bird and remaining birds.

3.0.3 IHSEV Agent. The IHSEV agent creates an internal Box2D simulation of the level, within which it tries out many shot angles and tap times. These mental simulations are carried out in parallel to identify the shot that destroys the most pigs. However, the simulation is not a perfect representation of the real Angry Birds environment and there are often many discrepancies between the two. The vision module has also been slightly improved from the base code provided so that objects are more robustly identified.

3.0.4 Angry-HEX Agent. The Angry-HEX agent uses HEX programs to deal with decisions and reasoning, while the computations are performed by traditional programming. HEX programs are an extension of answer set programming (ASP) which use declarative knowledge bases for information representation and reasoning. The Reasoner module of this agent determines several possible shots based on different strategies, each of which is then simulated using an internal Box2D simulation.

3.0.5 Eagle's Wing Agent. The Eagle's Wing agent chooses from five different strategies when deciding what shot to perform. These are defined as the pigshooter, TNT, most blocks, high round objects and bottom building blocks strategies. The decision of which strategy to use is based on the estimated utility of each approach with the currently selected bird. This utility is calculated based on the level's features and how these compare to a small collection of practice levels that are used to train the agent.

3.0.6 SEABirds Agent. The SEABirds agent uses an Analytic Hierarchy Process (AHP) for deciding which shots to make, and determines the best object or structure to hit based on five different criteria. This includes the Y-axis position, surrounding objects/structures, breakability (for currently selected bird type), relative distance to pigs and whether the object is a TNT box. The relative importance of each criteria compared to the other alternative options is calculated based on a collection of prior training levels.

3.0.7 s-birds Agent. The s-birds agent has two different approaches for determining the most effective shot to perform. The first strategy is called the bottom-up approach and identifies a set of candidate target blocks based on the potential number of affected pigs. The second strategy is called the top-down approach and utilizes the crushing/rolling effect of a bird or round block onto pigs, as well as the toppling effect of thinner blocks. Suitable target blocks are identified for each method and are then ranked based on the expected number of pigs killed and the likelihood of the shot's success.

3.0.8 Bambirds Agent. The Bambirds agent creates a qualitative representation of the level and then chooses one of nine different strategies based on its current state. This includes approaches such as utilizing blocks within the level to create a domino effect, targeting blocks that support heavy objects, maximum structure penetration and prioritizing protective blocks, as well as simpler options such as targeting pigs/TNT or utilizing certain bird's special abilities. These strategies are each given a score based on their estimated damage potential for the current bird type.

3.0.9 PlanA+. The PlantA+ agent alternates between two different strategies each time it attempts a level. The first strategy involves identifying two possible trajectories to every pig and TNT within the level, and then counting the number of blocks (for each material) that are blocking each trajectory from being successful. This is then compared against the type of bird that is currently available, to calculate a heuristic for each possible shot. The second strategy is similar to the first, except that the number of pixels crossing the trajectory is used rather than the number of blocks.

3.0.10 Vale Fina 007. The Vale Fina 007 agent uses reinforcement learning (specifically Q-learning) to identify suitable shots for unknown levels based on past experience. The current state of a level is defined using a list that contains information about every object within it. Each object is described based on several features, including the object angle, object area, nearest pig distance, nearest round stone distance, the weight that the object supports, the impact that the current bird type has on the object, and several others. Q-learning is then used to associate the features of the objects within a level to certain actions (shots) that result in success.

3.0.11 Condor. The Condor agent chooses from five different strategies when deciding what shot to perform. These are defined as the structure, boulder, TNT, bird and alone pig strategies. Each strategy has corresponding level requirements to decide whether it's considered or discarded for the current shot. Each strategy also has a numerical weighting based on human analysis of their potential impact for the current level.

3.0.12 AngryBNU. The AngryBNU agent uses deep reinforcement learning, more specifically it uses deep deterministic policy gradients (DDPG), to build a model for predicting suitable shots in unknown levels. The model trained with DDPG can be used to predict optimal shot angles and tap times, based on the features within a level. The level features that are considered when training and utilising this model are the current bird type, the distance to the target points, and a 128x128 pixel matrix around each target (nearby objects). Continuous Q-learning (SARSA) is used as the critic model and policy gradient is used as the actor model. By following this process, a deep learning model is trained to predict the best target point for a shot based on the level's features.

4 TYPES OF DECEPTION

Based on our analysis of the strategies and techniques utilised by our selection of agents, we have come up with six common types of deception that an Angry Birds level could contain, and that we believe have a strong possibility of causing agents to make poor shot decisions. Within some of these categories there also exist sub-categories based on more exact specifics. It is important to note that each type of deception described here is unlikely to be deceptive to all agents, as whether a level is considered deceptive or not is very specific to the agent and strategy that is used.

4.1 Material analysis

This type of deception requires the agent to analyse the material of certain blocks within structures to identify which bird types should be used on them. This involves more straight forward levels where the agent must simply use each bird against the material it is best against, but also more challenging levels in which blindly targeting the material best suited for the current bird will result in failure. The agent must understand that certain bird types are good against certain materials, but also that always choosing targets this way may not lead to the best outcome. The material that each bird type is strong/weak against is as follows:

- Red bird: Neither strong nor weak against any material.
- Blue bird: Strong against ice, weak against stone.
- Yellow bird: Strong against wood, weak against ice.
- Black bird: Strong against stone.
- White bird: Neither strong nor weak against any material.

4.2 Non-greedy actions

This type of deception requires the agent to take actions that may initially seem poor, but pay off in the long term (i.e. kill less pigs or deal less damage now, to kill more pigs later on). The agent must look ahead to the future birds that are going to be available later, and then make a decision with the current bird using this knowledge (agent must use forward planning). The result of the first shot(s) will likely not be the best possible for that bird on its own, but will allow the agent to either make a better shot with a subsequent bird or accomplish something that later birds cannot do.

4.3 Non-fixed tap time

This type of deception requires the agent to use a non-fixed tap time for bird abilities. Most of the agents we examined used a fixed tap time, either based on the trajectory distance to the object targeted or the first object hit, towards the end of the bird's flight path with a small amount of stochasticity. We therefore designed levels that required the agent to make either very early or very precise tap times, relative to the length of the bird's trajectory. The agent will have to understand the effect that taping a particular bird type has, and that this effect can be used in more ways that simply being stronger against certain materials. The abilities activated by each bird when tapped are as follows:

- Red bird: No special ability.
- Blue bird: Splits into three birds.
- Yellow bird: Shoots forward in a straight line.
- Black bird: Explodes and damages nearby objects.
- White bird: Drops an egg directly downwards that explodes on contact with another object.

No.	Description / Solution
01	Use yellow bird on unprotected pig and black bird on pig within stone structure
02	Same as previous level but now stone structure also has some wood blocks within it
03	Use black and yellow birds on correct structures
04	Use blue and yellow birds on correct structures
05	Make non-greedy shot with yellow bird
06	Make non-greedy shot with yellow bird (v2.0)
07	Make non-greedy shot with yellow bird (v3.0)
08	Make non-greedy shot with blue bird
09	Make non-greedy shot with black bird
10	Must “waste” first bird in order to solve level with second bird
11	Use blue bird tap time correctly (precise)
12	Use black bird tap time correctly (precise)
13	Use white bird tap time correctly (precise)
14	Use yellow bird tap time correctly (precise)
15	Use yellow bird tap time correctly (early / within normal range)
16	Use yellow bird tap time correctly (early / out of normal range)
17	Knock round wood block so that it rolls down slope onto pig
18	Destroy ice blocks supporting round stone blocks which roll onto pigs (indirect rolling)
19	Destroy ice blocks supporting round small stone blocks which roll onto pigs (indirect rolling)
20	Knock round stone block so that it falls on top of pig
21	Knock round small ice blocks so that they fall on top of pig
22	Target structure which collapses and falls on top of pig
23	Use falling red bird after shot collision to hit pig
24	Use falling red bird after shot collision to hit pig (v2.0)
25	Hit TNT to destroy structure and kill pigs
26	Hit TNT to push round stone block on top of pig
27	Target pig directly and ignore structures / TNT
28	Use first bird to clear path for second
29	Use first two birds to clear path for third
30	Use first three birds to clear path for fourth

Table 1: Level number and description / solution

4.4 Rolling / falling objects

This type of deception uses the fact that objects can roll or fall after they have been hit. Round blocks in particular can be easily pushed off terrain platforms or rolled down slopes. Other blocks and even the birds themselves can also do this. Because of this, we have come up with three sub-categories for this deception. The first involves rolling round blocks down slopes (by pushing them or destroying the objects supporting them) into pigs. The second involves pushing or rolling blocks off edges or steep drops onto pigs. The third uses the fact that a bird will fall downwards after its initial impact, and so could be used to hit pigs not normally reachable with its basic trajectory.

4.5 TNT

This type of deception involves the use of TNT boxes. These boxes explode when hit, damaging and/or pushing any objects that are nearby. Like the previous category, we have devised three possible cases for the use of TNT in deceptive levels. The first requires the agent to hit the TNT to cause direct damage to pigs or structures. The second requires the agent to hit the TNT to cause indirect damage to pigs by pushing other objects onto them. The third uses the TNT as a distraction from the real objective of killing pigs, the agent can solve the level by simply targeting the pigs and hitting the TNT will not help solve the level.

(a) (Material analysis) (b) (Non-greedy actions) (c) (Non-fixed tap time)

(d) (Rolling / falling objects) (e) (TNT) (f) (Clearing path)

Figure 2: Six example deceptive levels (a:02 b:05 c:13 d:18 e:26 f:28).

4.6 Clearing path

The final type of deception requires the agent to first clear a path to a pig before it can be killed. This pig will have obstacles preventing the agent from killing it immediately, and the agent must use the first bird(s) to destroy or move blocks that are protecting the pig. This might be done by directly destroying the block preventing a successful shot or moving these protective blocks by destroying their supports. The agent must often plan out a sequence of multiple shots in order to successfully clear a path to the pig.

5 EXPERIMENTS AND RESULTS

Using our six types of deception as a basis for creating challenging levels for agents, we designed 30 levels that we believe may deceive some agents into making poor shot decisions. A brief description of each level is given in Table 1, as well as six example levels shown in Figure 2, with full screenshots of all the other levels available in the appendix. To summarise the type of deception that each level focuses on, levels 01-04 focus on material analysis, levels 05-10 focus on non-greedy actions/shots, levels 11-16 focus on non-fixed (precise or early) tap times, levels 17-24 focus on rolling or falling objects (more specifically 17-19 are on rolling blocks, 20-22 are on falling blocks, and 23-24 are on falling birds), levels 25-27 focus on TNT (more specifically 25 is on direct TNT damage, 26 is on indirect TNT damage, and 27 uses TNT as a red-herring), and levels 28-30 focus on clearing paths.

5.1 Methodology

Each of our selected agents was given three sets of five minutes to solve each of our deceptive levels. Agents can attempt the level as many times as they like within each of these five-minute sets. Agents also had their memory wiped between each of these sets. To prevent agents which rely heavily on randomness in their decisions solving levels by lucky shots, each agent needed to solve a level in at least two out of these three sets to be counted. This experiment was carried out using an Ubuntu (14.04) 64-bit laptop PC, with an i5-2520M CPU and 8GB of RAM. While these specs may seem low, this is the same exact hardware that is used in the AIBirds competition setting to evaluate and run agents.

5.2 Agent Performance

After fully evaluating each agent's performance on our deceptive levels we can consolidate our results, see Table 2. This table shows which agents were able to consistently solve a particular level (solved in at least two out of three five-minute sets). We also include the total number of deceptive levels each agent was able to solve, the 2016/2017 AIBirds competition rankings, and the benchmark scores achieved by each agent on the first 42 levels of the "Poached Eggs" episode from the original Angry Birds game [1]. Figure 3 provides a more visual representation of the total number of levels containing each type of deception that each agent could solve.

The agent that managed to solve the most levels was Angry-HEX with 19 levels, while the agent that solved the least levels was PlanA+ with only five levels. None of the levels were able to be solved by all agents, and two of the levels could not be solved by any agent (levels 16 and 23). The hardest levels for most agents seemed to be those that required, non-greedy shots, precise tap-times, using the falling bird after first impact, and clearing paths to the pig. While some agents certainly performed better than others, no agent was able to successfully dominate across all types of deception.

5.3 Human Performance

We also recruited ten human participants to play our deceptive levels, again with a five-minute time limit on each level. These participants were allowed to play the first 21 levels from the Poached Eggs episode beforehand, to help those who had never played Angry Birds before learn the mechanics of the game. These levels are

Level Number	Naive	Datalab	IHSEV	Angry-HEX	Eagle's Wing	SEABirds	s-birds	Bambirds	PlanA+	Vale Fina 007	Condor	AngryBNU
01	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED		
02	SOLVED		SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED		
03	SOLVED	SOLVED			SOLVED		SOLVED			SOLVED	SOLVED	
04	SOLVED	SOLVED		SOLVED	SOLVED	SOLVED	SOLVED			SOLVED	SOLVED	
05	SOLVED	SOLVED		SOLVED			SOLVED	SOLVED		SOLVED	SOLVED	
06	SOLVED	SOLVED		SOLVED			SOLVED	SOLVED		SOLVED	SOLVED	
07	SOLVED	SOLVED		SOLVED			SOLVED	SOLVED		SOLVED	SOLVED	
08	SOLVED			SOLVED			SOLVED	SOLVED		SOLVED	SOLVED	
09	SOLVED				SOLVED					SOLVED	SOLVED	
10			SOLVED	SOLVED								
11			SOLVED									
12			SOLVED		SOLVED							SOLVED
13		SOLVED			SOLVED							SOLVED
14			SOLVED			SOLVED						
15		SOLVED	SOLVED	SOLVED	SOLVED	SOLVED			SOLVED			
16												
17		SOLVED	SOLVED	SOLVED	SOLVED	SOLVED						SOLVED
18		SOLVED	SOLVED		SOLVED							SOLVED
19		SOLVED	SOLVED		SOLVED							SOLVED
20		SOLVED	SOLVED	SOLVED	SOLVED	SOLVED						SOLVED
21			SOLVED	SOLVED	SOLVED	SOLVED						SOLVED
22		SOLVED		SOLVED								SOLVED
23												
24			SOLVED									
25		SOLVED		SOLVED	SOLVED			SOLVED				SOLVED
26			SOLVED	SOLVED	SOLVED			SOLVED	SOLVED		SOLVED	
27	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED	SOLVED			SOLVED		
28				SOLVED								
29				SOLVED		SOLVED						
30		SOLVED		SOLVED	SOLVED	SOLVED			SOLVED		SOLVED	
# solved	10	16	15	19	17	11	9	8	5	10	9	9
2016 rank	6th	3rd	2nd	7th	5th	4th	8th	1st	-	-	-	-
2017 rank	-	7th	2nd	3rd	1st	-	5th	9th	4th	6th	8th	10th
Benchmark	1,439,660	2,007,850	1,429,280	1,534,160	1,838,470	1,608,406	955,790	1,016,880	1,576,200	953,930	956,730	1,382,540

Table 2: Agent performance on deceptive levels (black square indicates solved in at least two out of three sets)

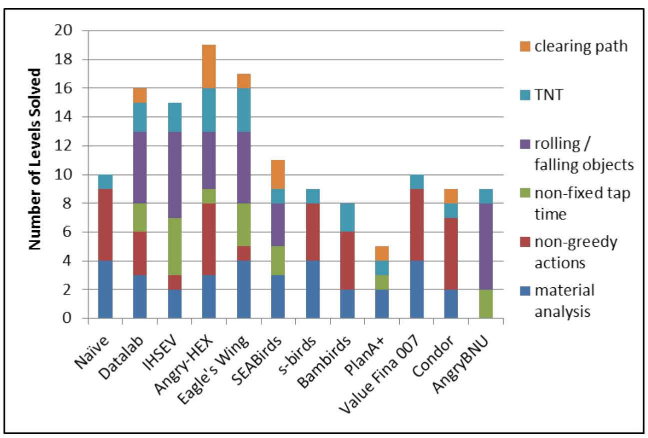


Figure 3: Number of levels that each agent could solve for each specific type of deception.

also available to all entrants in the AIBirds competition to help with designing and testing their agents. All participants were able to solve all 30 of our deceptive levels within the given time limit, showing that most humans and even newcomers to the game can solve these creative reasoning problems with relative ease.

6 DISCUSSION AND FUTURE WORK

Based on these results, we can now attempt to identify why certain agents and the techniques they use are more successful at dealing with certain types of deception than others.

6.1 Material analysis (levels 01-04)

Most agents were able to solve at least some of the material analysis levels. Agents that couldn't solve more than two levels tended to fail either levels 01 and 02, or 03 and 04. Levels 01/02 required the agent to target an unprotected pig first followed by a protected pig, whilst levels 03/04 required the agent to use the correct bird types on the correct structure materials. There doesn't appear to be much reason why specific AI techniques would struggle on these levels, suggesting that poorly defined heuristics or inaccurate simulations are likely to be the cause for the observed failures. Agents that couldn't solve levels 01/02 were likely coded to target the protected pig first, as this was perceived to cause more collateral damage or score additional points. Agents that couldn't solve levels 03/04 typically targeted the closest structure first, which always resulted in failure. Any agent with a stochastic target selection policy (such as the Naive agent) would be able to solve all material analysis levels given enough time, as it would eventually select the correct pigs to target by random chance. AngryBNU was the only agent that didn't solve any material analysis levels, as it kept making unusual shots without any real identifiable target. This habit of AngryBNU to make shots at nothing in particular continued into other deception categories as well. Datalab was tricked into making a poor shot by adding some wooden blocks to the stone structure in level 02, drawing its first shot away from the unprotected pig. Angry-HEX and SEABirds were able to solve level 04 but not 03, suggesting that the greater damage potential of the black bird lured them into making a poor initial shot.

6.2 Non-greedy actions (levels 05-10)

The non-greedy levels proved challenging for a lot of the agents, with some of the typically high-performing agent's such as IHSEV, SEABirds and Eagle's wing struggling far more than other generally worse agents. This is likely due to them always attempting to kill the maximum number of pigs possible with the current bird (as is certainly the case for IHSEV). While this is usually a wise course of action, failing to correctly plan an effective sequence of shots can sometimes lead to a poor final outcome. However, the fact that certain agents were able solve these levels does not automatically imply that they can plan multiple shots ahead. Some agent strategies are designed to target certain materials with specific bird types, which can cause them to inadvertently solve some of these non-greedy levels. This is backed up by the fact that most agents were able to solve non-greedy levels with certain bird types but not with others (e.g. Datalab was able to make non-greedy shots with the yellow bird but not with the blue or black birds). It is also the case that, similar to the material analysis levels, agents which select targets randomly would also be able to solve most of these levels by chance after multiple attempts. From our own observations of the agents playing these levels it is currently unclear whether any of them even consider which birds are still available, which is essential in planning out a sequence of shots. Interestingly the only agents that were able to solve level 10 were those that use an internal simulation to estimate shot outcomes (IHSEV and Angry-HEX). This level was unique in that it required the agent to essentially waste the first (blue) bird, in order to be able to solve the level with the second (yellow) bird (i.e. targeting the pig with the first bird makes the level unsolvable). It is likely that the simulations run by these successful agents determined that the pig could not be killed with the first bird, resulting in them making a random shot, but could find a valid solution using the second bird.

6.3 Non-fixed tap time (levels 11-16)

Most of the historically better performing agents with higher benchmark scores were able to solve at least some of the levels that required precise or early tap times for different bird types. This result might indicate that this is also a useful skill to have when attempting to solve more traditionally designed Angry Birds levels. Each of the levels requiring precise tap times (levels 11-14) could be solved by at least one agent, but no agent was able to solve them all. Most of these successful agents appeared to be proficient with estimating how the trajectory or properties of certain bird types changed when tapped, but bad at doing so for other bird types. IHSEV performed best on these levels and was the only agent to solve level 11. This success is likely due to its heavy reliance on internal simulations to evaluate many possible angles and tap times. This approach was a severe downside when tackling non-greedy levels but appears to have been far more successful here. Most good agents were able to solve level 15, where the yellow bird must be tapped before hitting the wooden block, but no agent was able to solve level 16 with a pig placed outside the regular range of a shot (must use yellow bird's ability to travel further than usual). This was likely due to the trajectory module for the game competition's framework being unable to find a valid release point, and is not specifically the fault of any particular AI technique. Another minor noteworthy point is that AngryBNU finally decided to stop firing at nothing and managed to solve some levels at last. It managed to solve levels 12 and 13 by bouncing the bird off the ceiling rather than using its ability; an unorthodox approach but successful nonetheless.

6.4 Rolling / falling objects (levels 17-24)

Much like the previous deception category, the agents with better benchmark scores typically performed much better on levels that required using rolling or falling objects to kill pigs. This would again suggest that this is a commonly required task when playing the original Angry Birds levels. Level 17 required agents to knock a round block (ball) down a slope into a pig to kill it, and could be achieved easily by most of the high-performing agents. Levels 18 and 19 took this to the next step by requiring the agent to instead break some blocks supporting several stone balls, which then roll onto pigs and kill them, with level 18 having large balls and level 19 having smaller balls. A couple of agents that solved level 17 couldn't deal with this additional level of reasoning, but those that did managed to solve both levels 18 and 19 successfully.

Levels 20 and 21 required the agent to knock large and small balls respectively, on top of a pig. Level 22 replaced these balls with a structure made of rectangular blocks. Most agents that solved level 20 also solved level 21, with the only exception being Datalab. By looking at Datalab's strategy description it would appear that it treats large balls as more damaging than small ones, which is likely the reason for this difference. Level 22 was solved by even fewer agents (although ironically Datalab solved it) and is likely due to agents treating round blocks as more likely to fall and do damage than regular structures. Levels 23 and 24 worked on a similar principle but required agents to use the fact that the bird itself falls after it makes contact with an object, and that this falling bird can still kill pigs if it hits them. This was by far the hardest idea for agents to deal with. The only agent that could successfully solve a level with this type of deception was IHSEV, which solved level 24, and was likely due to it stumbling across the successful action by chance when carrying out internal simulations of many shot options.

Amazingly, AngryBNU was able to solve all levels that used rolling or falling blocks, the only agent to do so. The reason for this is unclear, but definitely worth investigating further in the future. AngryBNU is the only agent that currently uses deep reinforcement learning to determine its shots and performed very poorly in most other types of deception, as well as in the most recent AIBirds competition [2]. However, it seems from our results that this approach has some useful benefits in specific situations, particularly those requiring agents to use other objects in the environment to cause indirect damage.

6.5 TNT (levels 25-27)

Due to the way that the TNT levels were designed, it was virtually impossible for each agent to not solve at least one level. It is therefore more important to look at which levels an agent solved in this category, rather than how many they solved. Naive, SEABirds, s-birds and Vale Fina 007 agents didn't target TNT at all in our levels and so were only able to solve level 27, where the agent must shoot at the pig and ignore the TNT boxes. Conversely, Bambirds always targets TNT in our levels even if doesn't help, meaning it was only able to solve levels 25 and 26. This suggests that this behaviour is hard coded and that Bambirds always targets available TNT, without performing any significant reasoning about the consequences of its actions. These issues are clearly caused by a lack of considered target possibilities and very poorly coded heuristics respectively.

A few agents were able to solve level 26 which required an understanding of indirect TNT damage (TNT explosion pushes ball on top of pig), but not level 25 where hitting the TNT directly causes the death of pigs. For IHSEV this could be caused by an internal simulation error (i.e. assumes that pigs will always die to TNT explosion regardless of the shot made), but the reason why the Plan A+ and Condor agents could only solve level 26 is unclear. Both Angry-HEX and Eagle's Wing were the only agents that managed to solve all three TNT levels, suggesting that they can accurately predict the damage and effect that TNT boxes can have on surrounding objects.

6.6 Clearing path (levels 28-30)

The first two clearing path levels (28 and 29) required the agent to initially target objects away from the pig in order to successfully hit it with later birds. This was a challenging concept for most agents, with only Angry-HEX and SEABIrds being able to solve either of these levels. Interestingly, SEABirds was only able to solve level 29 which required the agent to destroy two protective barriers between the slingshot and pig but not level 28 which had only one barrier. This could be due to the fact that the agent believed it could kill the pig in level 28 without destroying the barrier, or because the design of the protection was more complex than in level 29. Angry-HEX was able to solve both levels, suggesting that it currently has the best structural analysis abilities and an understanding of how targeting critical support blocks can make solving a level easier for later birds. Level 30 required the agent to destroy three separate barriers before the pig could be hit, but each of these barriers could be destroyed by simply targeting the pig with a low angle trajectory. This level is actually therefore easier than the previous two, but agents must still be smart enough to target the pig with a low angle shot four times in succession (any high angle shots will make the level unsolvable). Agents that rely on heavily stochastic methods could theoretically solve this level given enough time but would only manage to do so very infrequently.

6.7 Summary

From these results it appears that each of the current state-of-the-art Angry Birds agents is vulnerable to at least some kind of deception, but different approaches have their own strengths and weaknesses. Based on this information it would be possible to design a set of levels that any specific agent would be unable to solve, meaning that the relative difficulty of a particular level is highly dependent on the agent being used. It would also be possible to create levels that contain multiple types of deception, perhaps being able to fool most or all of the current agents. Understanding exactly why each agent and the approach it uses fails at certain types of deception, as well as how to identify these deceptive elements within a given level, is a problem that must be solved if the goal of creating efficient, skilful and adaptable agents that can play as well as human players is to be achieved.

Comparing each agent's deceptive level performance against competition rankings and benchmark scores, allows us to examine how often these deceptive elements appear in more traditional Angry Birds levels. Not every evaluated agent participated in both the 2016 and 2017 competitions, making a formal calculation using this data difficult. However, a moderate positive correlation coefficient of 0.5787 exists between each agent's benchmark score and the number of deceptive levels solved. While agents with higher benchmarks tended to perform better overall, they are still vulnerable to certain types of deception due to their assumptions and pre-set strategies. Datalab, Eagle's Wing and SEABirds all outperformed Angry-hex in benchmark scores and the 2016 competition rankings, but performed worse overall on these deceptive levels. This drop in performance demonstrates how certain levels can be constructed to heavily favour certain agents over others.

This research and the results presented have many applications beyond Angry Birds, to both other video games and real-world problems. Deceptive categories such as these emphasises the need for agents to utilise multiple different AI techniques when attempting to perform complex and highly varied tasks with imprecise information. Whilst deception categories such as TNT, rolling objects, material analysis and non-fixed tap times are quite specific to Angry Birds, the reason why some agents fail on levels that contain these types of deception can be extended beyond this game. No matter how many heuristics or pre-defined strategies an agent is coded with, it will always be possible to design problems that it cannot solve. The fact that some agents use internal simulations (IHSEV and Angry-HEX) or reinforcement learning techniques (AngryBNU and Vale Fina 007) to help improve their abilities is a good start, but these additions suffer from their own problems and limitations. We have only scratched the surface here in terms of the analysis and discussion that could be performed. The sheer variety of AI techniques and strategies that are employed by the currently available agents make it very difficult to pinpoint exactly why the results are the way they are. Nevertheless, we hope to have provided and accurate and concise summary of where the current state-of-the-art is lacking and where certain teams may want to focus their efforts when attempting to improve their agents.

6.8 Future Work

The most obvious way for future agents to deal with these types of deception would be to expand the range of AI techniques and strategies they can utilise. Even if we combine the performance of just the four best agents (Datalab, IHSEV Angry-HEX and Eagle's Wing), we can theoretically solve 28 of the 30 deceptive levels. However, it is not only important that an agent has more approaches to solve levels, but also that it can accurately identify when to use them. Bambirds has nine potential strategies for selecting shots compared to Datalab's four, but the performance of the latter agent is considerably better. Estimating the outcome of particular shots, even in a more general and qualitative way, is vitally important when attempting to plan out an effective sequence of shots. Until this can be achieved, agents will always fail to equal the performance of human players.

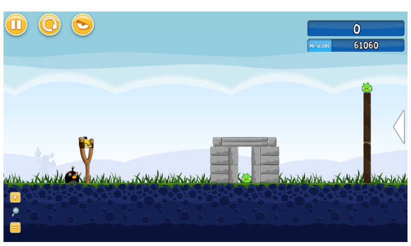
Future research could involve either identifying levels that contain one of these types of deception and determining the AI approach that would be most suitable (e.g. an ensemble or hybrid agent), or by developing more sophisticated AI and machine learning techniques to better solve each deception category (e.g. dynamic programming or simulation training). Further analysis could also be carried out on other video games with different mechanics and challenges. It is clear from our human performance analysis that whilst agents may struggle, humans are very adept at solving these deceptive levels. Investigating how human players are able to think and reason about these types of deception may help design agents that use the same assumptions and generalisations, potentially improving their overall performance. Also worth investigating is whether humans enjoy playing levels with certain types of deception more. Increasing the length of time to solve a level doesn't necessarily increase the difficulty or challenge if the reasoning and actions required to solve it are still relatively simple. It is highly likely that levels which contain deceptive elements require players to think more creatively about the problem, hopefully leading to a greater level of enjoyment. This was confirmed empirically through participant discussions, but further analysis may yield substantial benefits for level designers.

REFERENCES

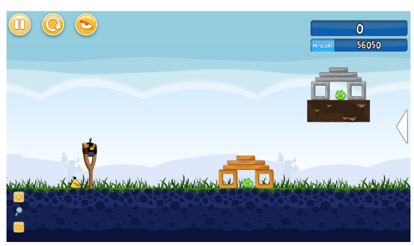
[1] AIBirds. 2017. Agent Benchmarks. https://aibirds.org/benchmarks.html. Accessed: 2017-11-21.
[2] AIBirds. 2017. AIBirds 2017 Competition Results. https://aibirds.org/angry-birds-ai-competition/competition-results.html. Accessed: 2017-11-21.
[3] AIBirds. 2017. AIBirds Homepage. https://aibirds.org. Accessed: 2017-11-21.
[4] Damien Anderson, Matthew Stephenson, Julian Togelius, Christoph Salge, John Levine, and Jochen Renz. 2018. Deceptive Games. In *21st International Conference on the Applications of Evolutionary Computation*.
[5] F. Calimeri, M. Fink, S. Germano, A. Humenberger, G. Ianni, C. Redl, D. Stepanova, A. Tucci, and A. Wimmer. 2016. Angry-HEX: An Artificial Player for Angry Birds Based on Declarative Knowledge Bases. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (2016), 128–139.
[6] S. Dasgupta, S. Vaghela, V. Modi, and H. Kanakia. 2016. s-Birds Avengers: A Dynamic Heuristic Engine-Based Agent for the Angry Birds Problem. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (2016), 140–151.
[7] Anjali Narayan-Chen, Liqi Xu, and Jude Shavlik. 2013. An Empirical Evaluation of Machine Learning Approaches for Angry Birds. In *IJCAI Symposium on AI in Angry Birds*.
[8] Mihai Polceanu and Cedric Buche. 2013. Towards A Theory-Of-Mind-Inspired Generic Decision-Making Framework. In *IJCAI Symposium on AI in Angry Birds*.
[9] Jochen Renz. 2015. AIBIRDS: The Angry Birds Artificial Intelligence Competition. In *AAAI Conference on Artificial Intelligence*. 4326–4327.
[10] Jochen Renz, Xiaoyu Ge, Stephen Gould, and Peng Zhang. 2015. The Angry Birds AI Competition. *AI Magazine* 36, 2 (2015), 85–87.
[11] Jochen Renz, XiaoYu Ge, Rohan Verma, and Peng Zhang. 2016. Angry Birds as a Challenge for Artificial Intelligence. In *AAAI Conference on Artificial Intelligence*. 4338–4339.
[12] S. Schiffer, M. Jourenko, and G. Lakemeyer. 2016. Akbaba: An Agent for the Angry Birds AI Challenge Based on Search and Simulation. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (2016), 116–127.
[13] Matthew Stephenson and Jochen Renz. 2017. Creating a Hyper-Agent for Solving Angry Birds Levels. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.
[14] Matthew Stephenson, Jochen Renz, Xiaoyu Ge, and Peng Zhang. 2018. The 2017 AIBIRDS Competition. arXiv:1803.05156 arXiv:1803.05156v1.
[15] N. Tziortziotis, G. Papagiannis, and K. Blekas. 2016. A Bayesian Ensemble Regression Framework on the Angry Birds Game. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (2016), 104–115.
[16] P. A. Wałęga, M. Zawidzki, and T. Lechowski. 2016. Qualitative Physics in Angry Birds. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (2016), 152–165.
[17] Peng Zhang and Jochen Renz. 2014. Qualitative Spatial Representation and Reasoning in Angry Birds: The Extended Rectangle Algebra. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*. 378–387.

A DECEPTIVE LEVELS

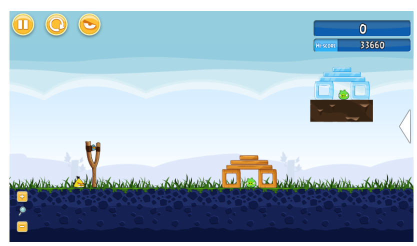
Additional pictures of the deceptive levels used in our evaluation, not including those already shown in Figure 2.



01 (Material analysis)



03 (Material analysis)

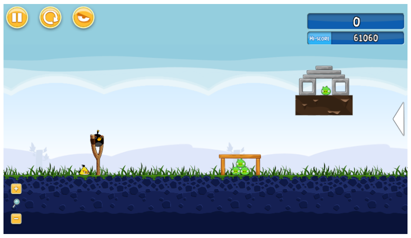


04 (Material analysis)

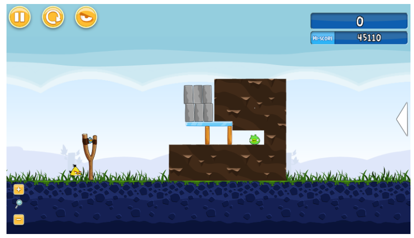
06 (Non-greedy actions)

07 (Non-greedy actions)

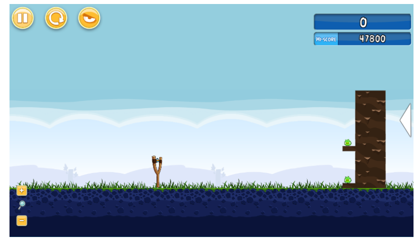
08 (Non-greedy actions)



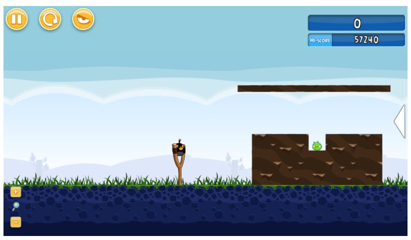
09 (Non-greedy actions)



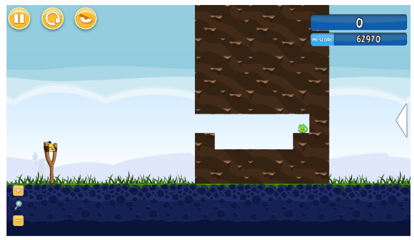
10 (Non-greedy actions)



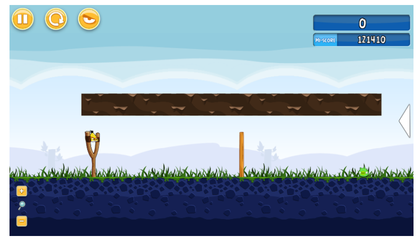
11 (Non-fixed tap time)



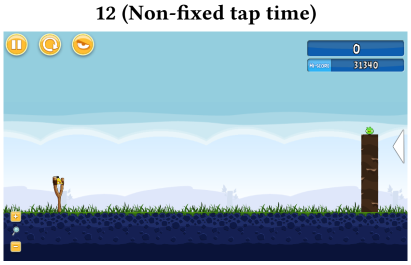
12 (Non-fixed tap time)



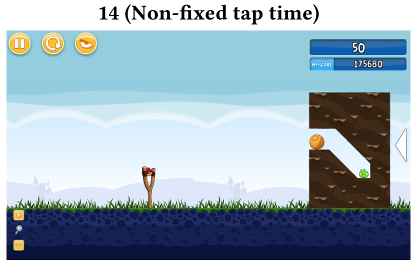
14 (Non-fixed tap time)



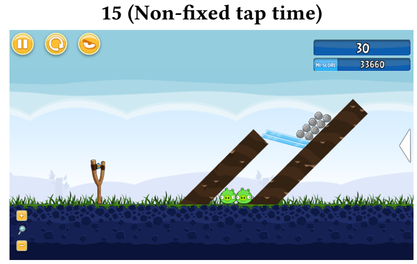
15 (Non-fixed tap time)



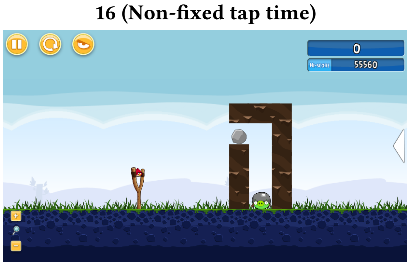
16 (Non-fixed tap time)



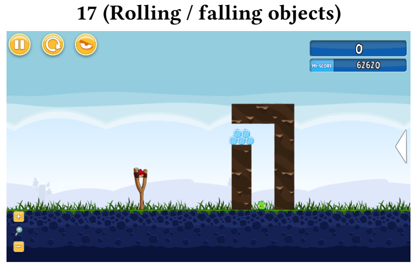
17 (Rolling / falling objects)



19 (Rolling / falling objects)



20 (Rolling / falling objects)



21 (Rolling / falling objects)

22 (Rolling / falling objects)

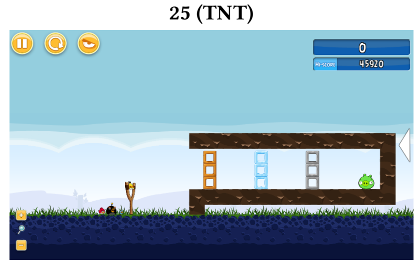
23 (Rolling / falling objects)

24 (Rolling / falling objects)

25 (TNT)

27 (TNT)

29 (Clearing path)



30 (Clearing path)