Analysis of generation algorithm for angry birds style physics games

Entertainment, Video Games

This article presents Tanager, a level generator based on a genetic algorithm capable of producing achievable levels for the game Angry Birds. The evaluation of the gameplay in this game requires checking both the stability of stacked blocks and the ability to kill all pigs with the given amount of birds. These two components are managed by the algorithm via a simulation. Three sets of experiments are conducted to evaluate Tanager. The first is the performance of the genetic algorithm under Tanager. The second explores the expressivity of the generated levels taking into account their structural characteristics. The third one measures the design quality of the levels via an online user survey. The results show that Tanager is able to generate a considerable variety of achievable levels as attractive and enjoyable as those designed manually. However, generated levels are less difficult than those created by hand.

Tanager often called as Procedural Level Generation, is the technology used in the gaming world to design gaming levels automatically or semi-automatically via the use of algorithms. The particular technology has been a game changer in the particular domain adversely affecting the video game genre such as platformers, real-time strategy, first person shooters, racing etc. The main aim of the technique/technology used is to create expressive, feasible and engaging levels. In case of most of the games path finding algorithms are used to check for feasibility of the level as to whether the in game character can reach the end point of the level or not. Generating achievable levels is a more complex task in the under-explored kind of physics-based puzzle Games. In this genre, different rigid and soft bodies are used to create puzzles to solve by considering forces such as gravity and other physical constraints. So, producers must decide where to place such bodies a given space to create interesting puzzles. In addition, bodies can be static or dynamic. Static objects do not move and are not affected by forces or collisions. Dynamic objects can move and are affected by the physics of the game. A level is considered feasible if the dynamic bodies are initially stable and if there is a set of valid moves allowing the player to solve it. Levels must be initially stable to ensure that the objects will not be move before any action of the player. Difficulty to evaluate the feasibility of a level is directly related to the type (static or dynamic) and the amount of game objects forming the geometry and challenges of the level. Tanager is built on and extends the work, which introduced a creation-oriented GA Stable blocks of blocks for the game Angry Birds. Tanager improves level representation to evolve more complex structures as well as the number of birds in the level The initialization method previously designed is also extended to improve the stability of the structures during the evolutionary process. A new physical form function is introduced to evaluate the playability of the generated levels.

Angry Birds Overview

Angry Birds is a physics-based puzzle game where the player uses a slingshot to shoot birds at structures made up of blocks, with pigs placed in or around them. The goal of the player is to kill all pigs using the birds provided. The level of atypical angry birds, as shown in Figure 1, contains birds, pigs and a collection of blocks arranged in one or more structures. Each bird is assigned one of five different types (red, blue, yellow, black or white) and each block is assigned to one of three materials (wood, ice or stone). TNT can also be placed in one level and explodes when reached by another object.

Original level generator

The level generator proposed in this paper, takes reference from previous angry birds level generator. It creates levels consisting of a collection of independent units, constructed using the blocks mentioned in the Figure 2. The selected blocks are aligned in different ways to give out different orientation. Local stability requirements are applied on the structure which is to be developed and the units can be added to the structure till the structure reaches a desired height and the stability requirements are fulfilled. The units in the game can be developed from 3 different kinds of materials.

The generator must take care of the weak points throughout the level shielding it as much as possible.

Improved level generator

A block can be either elementary or composed, where elementary blocks are unitary pieces connected to form composed ones.

Block Swapping

The level generator technique discussed above has some limitations which reduces the variety of structures that could be created. In this modified technique of level generator we make use of the “ Block Swapping” approach where an attempt is made to replace some of the blocks in the structure with blocks of some other type having the same height taking into consideration the stability requirements as well.

Terrain Variation

Another minor update to increase the level variety is to use different heights and terrain angles. While platform structures can be suspended in the air at different locations, floor structures were previously always at the same height. Instead, we now allow floor structures to be placed under the floor, resulting in an increase in the range of vertical positions. For each soil structure in a generated level (starting with the leftmost structure and moving to the right), there is a random chance (G) that the current height of the soil increases or decreases.

Conclusion

This article has presented a procedural level generation algorithm for Angry Birds style physics games, which can guarantee that the levels it creates are both stable and soluble. Improvements not only in structures and terrain, but also in their evaluation and optimization, produce levels much higher than those previously generated. We also used Angry Birds agents to check that levels created. The structure generated is evaluated using a high level fitness function, which take into consideration the pig placement potential, type of the blocks to be used and robustness of the structure. There is a wide range of future possibilities for this research. An example might be to develop a structure generation method that can create structures that are no longer segmented into distinct lines, or perhaps inclined structures for sloping traction.