The Intelligence Engine

Sunday, August 11, 2013

Decision Modeling and Optimization in Game Design, Part 5: Class Assignment Problems

This article is the fifth in a continuing series on the use of decision modeling and optimization techniques for game design. The full list of articles includes:

- *Part 1: Introduction (Blogspot) (Gamasutra)*
- *Part 2: Optimization Basics and Unrolling a Simulation (Blogspot) (Gamasutra)*
- *Part 3: Allocation and Facility Location Problems (Blogspot) (Gamasutra)*
- *Part 4: Competitive Balancing Problems (Blogspot) (Gamasutra)*
- *Part 5: Class Assignment Problems (Blogspot) (Gamasutra)*
- *Part 6: Parametric Design Techniques (Blogspot) (Gamasutra)*
- *Part 7: Production Queues (Blogspot) (Gamasutra)*
- *Part 8: Graph Search (Blogspot) (Gamasutra)*
- *Part 9: Modular Level Design (Blogspot) (Gamasutra)*

The spreadsheet for this article can be downloaded here: link

Introduction

In the previous articles in this series, we introduced the concepts behind decision modeling and optimization – a set of tools that can be helpful in framing and optimizing a surprising number of design decisions. We also discussed their limitations and showed a number of examples of how they can be used to find optimal player strategies in certain cases and to help optimize design decisions directly (see links above).

In this week's update, we'll introduce assignment problems and show some potential ways to solve this class of problems. Then we'll reveal the stunning plot twist, that this entire article has been an elaborate setup for the next episode, in which we'll discuss some practical applications of these techniques and discuss how we used a similar approach to frame the building design problems in our recent iOS/Android game City Conquest.

The Problem

It's your first day on the job at a company that designs massively-multiplayer role-playing games (MMORPGs). You're very excited to start work on their new fantasy MMORPG, which has been in development for six months.

Upon arriving at work, you receive a long-distance Skype call from your boss, who has left for two months of intensive research in Tahiti.

"We've got a problem," she explains. "We've just refactored our design. Totally changed everything – it wasn't working out, so we smashed it to bits and restarted from scratch."

"That's cool," you reply. "You guys warned me during the interview that you do a lot of design changes."

"Yeah, we do. So the net result is, we've got an issue with the tier 2 spells for the spellcasting classes, and I'd like you to take a look at it. We have five character classes in the game that can cast spells. We also have five spells that aren't assigned right now, and five 'perks' that are in design that seem like a good fit for spell-casting character classes. So I want you to figure out which classes get which spells and which perks."

"Sounds easy enough," you reply.

"See if you can take a shot at this," she replies. "Here are the classes, spells, and perks:"

- Classes: Warlock, Wizard, Mage, Runecaster, Sorcelator
- Spells: Fireball, Iceball, SpellSteal, Drain Life, Smashing Hand
- Perks: Mana Regeneration, Endurance, Haste, Fire Mastery, Ice Mastery

"OK. How do you want me to go about doing these assignments?"

"Here's the feedback I've gotten from the design team:

- It seems pretty obvious that the Fire Mastery and Ice Mastery perks should go with whatever classes get the Fireball and Iceball spells, respectively.
- We'd like the Warlock to get the Mana Regeneration perk.
- We haven't figured out what spell the Wizard should get, but we do know that Iceball would not be a good fit for that class.
- We want the Runecaster to get the Drain Life spell.
- The Haste perk would not be appropriate for the Mage.
- The Mana Regeneration perk isn't a good fit for the class that ends up getting the Spellsteal spell.
- We want to make sure the Mage does not get the Fireball or Iceball spells."

"OK, boss," you reply. "I've got all that written down."

"I honestly don't know if there's a way of assigning these that even works," she continues. "It might not even be possible. Alternatively, there might be more than one way to assign it that makes it work. I just don't know. What do you think?"

Think about this problem for a moment. How would you solve this? And how many solutions are there – zero, one, or more than one?

It's Tougher Than it Looks

This is a difficult problem. Unlike most of the problems in this series, though, it's one that you can, in fact, solve by hand (where by "solve" I mean that you can find a feasible solution – in this case, there is only one).

If you've ever solved a classical logic puzzle, you might recognize this problem. Most of these take the form of figuring out the correct assignments of objects. For example, you have to figure out which of N people wore N hats of N colors to a certain party.

In order to solve these kinds of logic puzzles, you typically make a matrix of adjoining NxN tables, with one table for each permutation of two

Blog Archive

- ► 2015 (5)
- ► 2014 (3)
- ▼ 2013 (9)
  - ► December (1)
  - ► September (2)
  - ▼ August (2)
    - Decision Modeling and Optimization in Game Design,...
    - Decision Modeling and Optimization in Game Design,...
  - ► July (4)

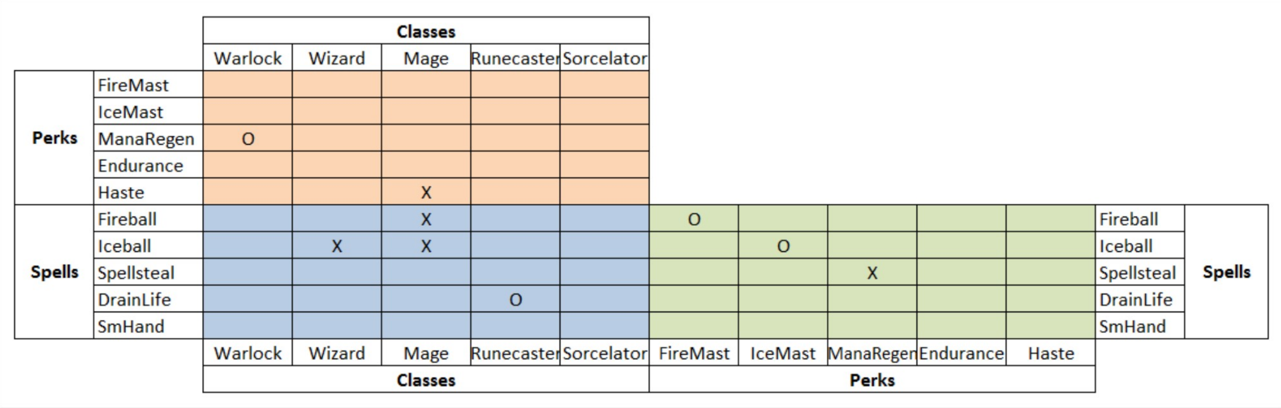
Contributors

- Paul Tozour
- Sensai Pose
- Unknown

possible assignment. Similarly, if any given row or column of an NxN subtable is filled with 'x' values except for one cell, you can fill that empty cell with an 'o,' since by process of elimination, the one remaining cell must be the valid assignment.

(There are some good examples online at http://www.printable-puzzles.com/printable-logic-puzzles.php, and you can see tips on solving them at http://www.puzzlersparadise.com/article1021.html and http://www.youtube.com/watch?v=CNxfZwvaQ-k.)

Since the problem we're trying to solve is assigning N classes to N spells to N perks, where N=5, we'll need a set of 3 adjoining 5x5 tables, one each for all of the possible [classes, spells], [classes, perks], and [perks, spells] assignments. Here's what that would look like in our case: we have a [classes, perks] table (in peach, at the top), a [classes, spells] table (bottom left), and a [perks, spells] table (bottom right).



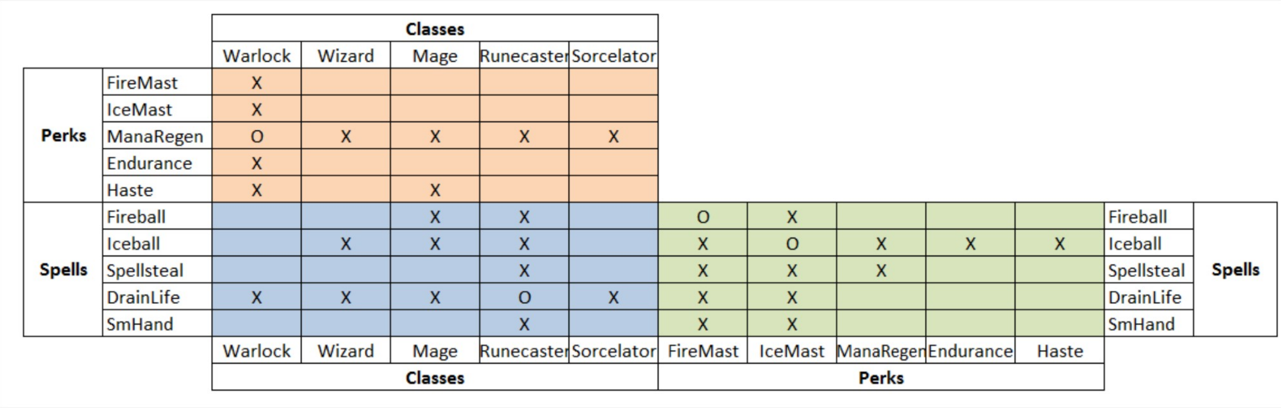
		Classes					Perks						
		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	ManaRegen	Endurance	Haste		
Perks	FireMast												
	IceMast												
	ManaRegen	O											
	Endurance												
	Haste			X									
Spells	Fireball			X			O					Fireball	Spells
	Iceball		X	X				O				Iceball	
	Spellsteal								X			Spellsteal	
	DrainLife				O							DrainLife	
	SmHand											SmHand	
		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	ManaRegen	Endurance	Haste		
		Classes					Perks						

In order to get the ball rolling, we've already started filling in the table with the constraints from the problem description. We fill in an 'O' in any cell where we know that the assignment must hold, and an 'X' in any cell where we know an assignment is forbidden.

In this case, the boss' insistence that the Fire Mastery and Ice Mastery perks should go with the Fireball and Iceball spells made us put the two 'O' cells in the green perks/spells table in the lower right, one correlating Fire Mastery with Fireball and the other correlating Ice Mastery with Iceball. The 'X' just below and to the right of those, denying a correlation between Mana Regeneration and SpellSteal, encapsulates the boss' insistence that the Mana Regeneration perk isn't a good fit for the class that ends up getting the Spellsteal spell. If you read through all of the other constraints listed, you'll see that the table above encapsulates them precisely.

The next step is to make as many inferences as we can from that point. For every cell with an 'O,' we know that no other cell in that row or column in that sub-table can have an 'O.' Therefore, we can fill them in with 'X' marks.

For example, we've filled in the 'O' in the upper left (peach) sub-table indicating that the Warlock gets the Mana Regeneration perk. Therefore, no other class can get Mana Regeneration, and the Warlock cannot have any other perk. Once we fill in all of those 'X' marks, we get a table that looks like this:



		Classes					Perks						
		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	ManaRegen	Endurance	Haste		
Perks	FireMast	X											
	IceMast	X											
	ManaRegen	O	X	X	X	X							
	Endurance	X											
	Haste	X		X									
Spells	Fireball			X	X		O	X				Fireball	Spells
	Iceball		X	X	X		X	O	X	X	X	Iceball	
	Spellsteal				X		X	X	X			Spellsteal	
	DrainLife	X	X	X	O	X	X	X				DrainLife	
	SmHand				X		X	X				SmHand	
		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	ManaRegen	Endurance	Haste		
		Classes					Perks						

That's a lot better! Now we can instantly see what assignments can't be possible, and this can give us a much better handle on what can be possible. We can also use a few other basic rules of logic to complete this puzzle – for example, if there are four 'X' marks in any row or column, and the fifth cell in that row or column is blank, then we know by process of elimination that there must be an 'O' there. Luckily, the step above nearly filled in the "Iceball" row in the blue spells/classes assignment table in the lower left, so if we can rule out either the Warlock or the Sorcelator for Iceball, we can fill in the other class.

We can also use basic logical inference to help us fill in more. For example, we can see that the Warlock has Mana Regeneration, and the spells/perks table in the lower right (green) shows that we have 'X' marks in the "Mana Regen" column for Iceball and SpellSteal. This means that we can go to the spells/classes table in the lower left (blue) and fill in 'X' marks in the Warlock column for Iceball and SpellSteal.

Which, of course, from our inference in the prior paragraph, means that the Sorcelator must be the one who gets Iceball …

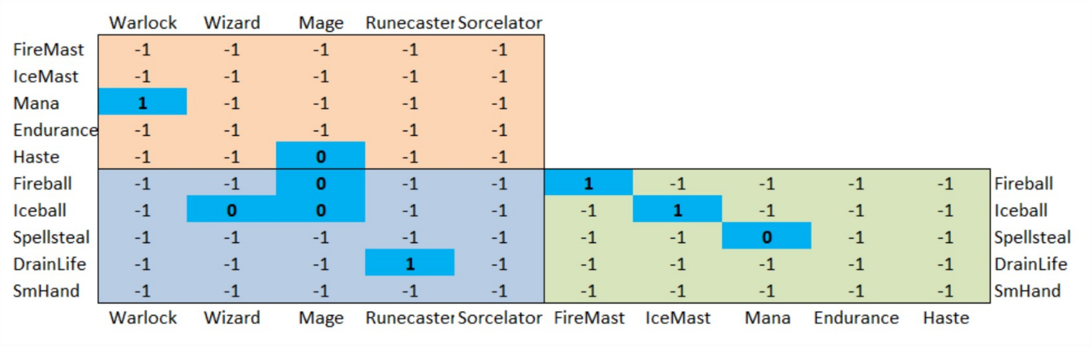
But instead of completing the exercise in that way, we're going to stop there and change course. Curious readers are welcome to try to solve this puzzle by hand – it's not too difficult from this point – but taking the manual exercise any further would be a waste of time. After all, this series is all about automated decision modeling. We're all about getting computers to do the heavy lifting for us, not filling in big tables by hand.

So instead of trying to find the solution manually, let's see if there's a way we can frame this as a decision modeling problem in Excel. Even though this is obviously a toy problem, there's surely something we can learn from this.

Design Wizard Casts "Summon Decision Model"

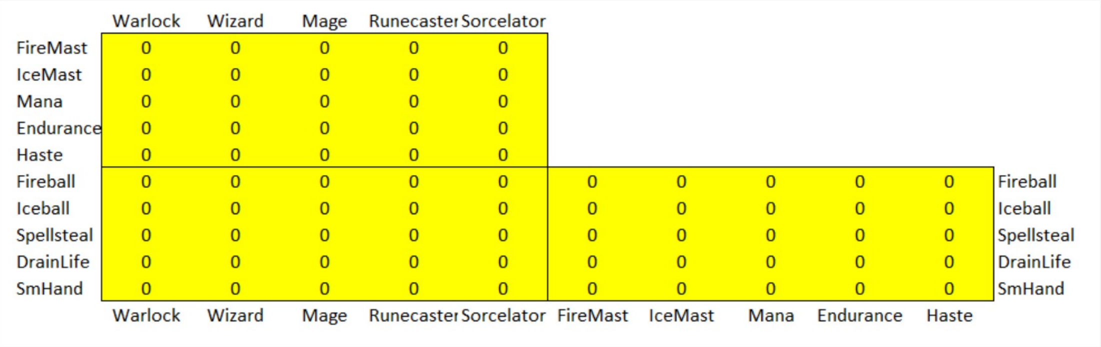
Ideally, we could somehow take this table and import it whole-hog into Excel, and then let Excel figure out whether to fill in each cell with an 'X' or an 'O.' Unfortunately, trying to do it in a single unified table is a bit messy – Solver is going to want us to specify our decision cells (and set the

First, we make the “constraint table,” as shown below. This table is exactly the same shape as our manual logic puzzle, but we fill in each cell with either ‘1’ to indicate an ‘O’ value, ‘0’ (zero) to indicate an ‘X’ value, or ‘-1’ to indicate that the value of the cell isn’t stated in the problem definition.



	Warlock	Wizard	Mage	Runecaster	Sorcelator						
FireMast	-1	-1	-1	-1	-1						
IceMast	-1	-1	-1	-1	-1						
Mana	1	-1	-1	-1	-1						
Endurance	-1	-1	-1	-1	-1						
Haste	-1	-1	0	-1	-1						
Fireball	-1	-1	0	-1	-1	1	-1	-1	-1	-1	Fireball
Iceball	-1	0	0	-1	-1	-1	1	-1	-1	-1	Iceball
Spellsteal	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	Spellsteal
DrainLife	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	DrainLife
SmHand	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	SmHand
	Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	Mana	Endurance	Haste	

Next, we’ll make a separate table for the decision variables. This table will be the one we optimize with Solver. We’ll fill in each cell with 0 to start out, and we’ll ensure that Solver sets each cell to either 0 or 1.

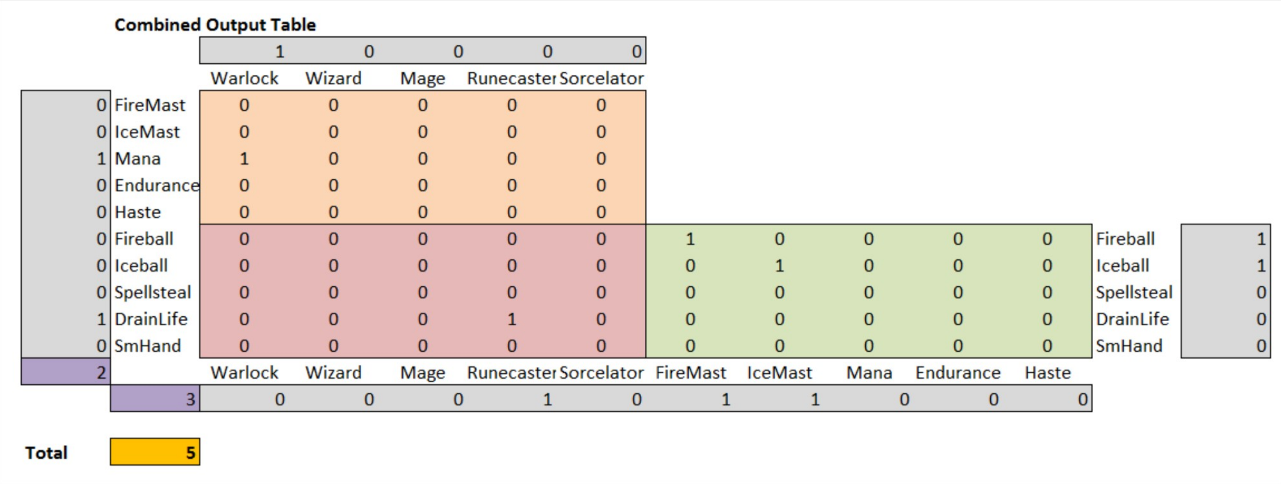


	Warlock	Wizard	Mage	Runecaster	Sorcelator						
FireMast	0	0	0	0	0						
IceMast	0	0	0	0	0						
Mana	0	0	0	0	0						
Endurance	0	0	0	0	0						
Haste	0	0	0	0	0						
Fireball	0	0	0	0	0	0	0	0	0	0	Fireball
Iceball	0	0	0	0	0	0	0	0	0	0	Iceball
Spellsteal	0	0	0	0	0	0	0	0	0	0	Spellsteal
DrainLife	0	0	0	0	0	0	0	0	0	0	DrainLife
SmHand	0	0	0	0	0	0	0	0	0	0	SmHand
	Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	Mana	Endurance	Haste	

Finally, we’ll make a third table that combines both of these two: the “combined output table.” This table will combine the values of the “constraint table” and the decision cells together. Each cell in this third table will use the value in the “constraint table” if it is 0 or 1, but if it’s -1 (meaning it’s not explicitly defined in the problem statement), it will use the value from the (yellow) decision variable table instead. Each cell in this combined output table will serve as the actual, final value.

We will also use special counter cells around the outer edges to determine how many cells in each row or column of each sub-table are set to a value of ‘1.’ We know that the final solution should have exactly a single assignment for each combination, so it should have exactly one ‘1’ value in each row and column of each of the three sub-tables, so we know that these summation cells should have a final value of 1.

Here’s what the combined output table looks like:



Combined Output Table

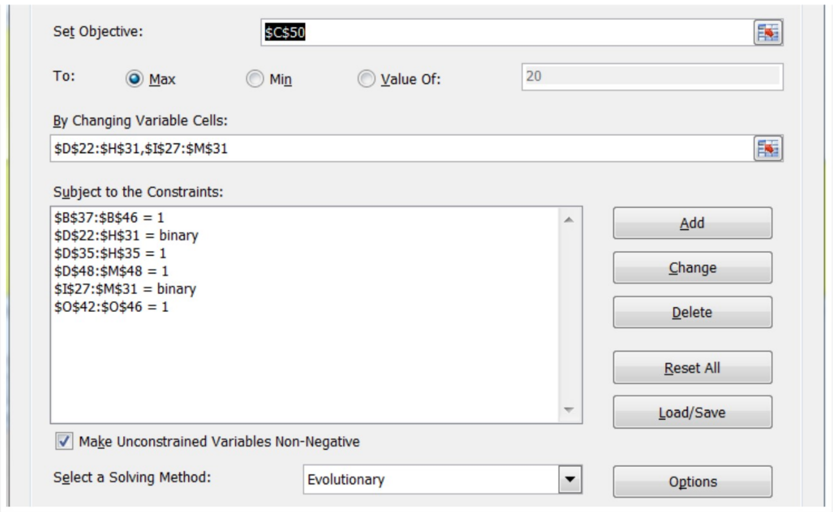
		1	0	0	0	0							
		Warlock	Wizard	Mage	Runecaster	Sorcelator							
0	FireMast	0	0	0	0	0							
0	IceMast	0	0	0	0	0							
1	Mana	1	0	0	0	0							
0	Endurance	0	0	0	0	0							
0	Haste	0	0	0	0	0							
0	Fireball	0	0	0	0	0	1	0	0	0	0	Fireball	1
0	Iceball	0	0	0	0	0	0	1	0	0	0	Iceball	1
0	Spellsteal	0	0	0	0	0	0	0	0	0	0	Spellsteal	0
1	DrainLife	0	0	0	1	0	0	0	0	0	0	DrainLife	0
0	SmHand	0	0	0	0	0	0	0	0	0	0	SmHand	0
2		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	Mana	Endurance	Haste		
	3	0	0	0	1	0	1	1	0	0	0		

Total 5

The grey cells around the left, right, top, and bottom are the summation cells – as you can see, they each count the total of the cells in the nearest row or column of the nearest sub-table. We use these to count the number of times each element is used. Once we’ve found the proper assignments, all of these grey cells should equal 1, to indicate that there’s been exactly five non-conflicting [spell, class] assignment, exactly five non-conflicting [class, perk] assignments, and exactly five non-conflicting [spell, perk] assignments.

The two purple cells in the lower left calculate the sum of the left-hand column and the bottom row of the summation cells. We know that once all of the tables are filled in, each of these should have a value of 10, since each of the 10 cells of the left-hand column and the bottom row should have a value of 1.

The orange objective cell at the bottom simply adds the values of the two purple cells.



Set Objective: $C$50

To: ● Max ○ Min ○ Value Of: 20

By Changing Variable Cells:

$D$22:$H$31,$I$27:$M$31

Subject to the Constraints:

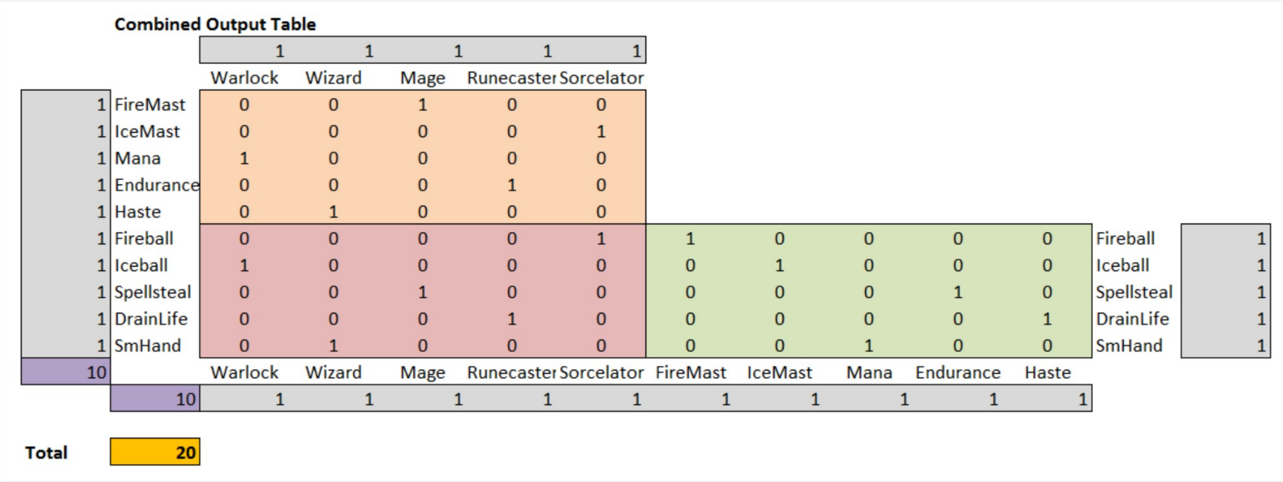
$B$37:$B$46 = 1
$D$22:$H$31 = binary
$D$35:$H$35 = 1
$D$48:$M$48 = 1
$I$27:$M$31 = binary
$O$42:$O$46 = 1

☑ Make Unconstrained Variables Non-Negative

Select a Solving Method: Evolutionary

- The cell in the “Set Objective” category is the orange objective cell, and we try to maximize its value (we could set it to “Value Of”=20 knowing that its value cannot exceed 20, but in practice, this is less reliable than simply maximizing it, since it can’t physically go over 20 anyway).
- The cell ranges in the “By Changing Variable Cells” category are our yellow decision cells in the binary decision variable table.
- The two “…=binary” constraints are constraints on our decision cells. Setting a “binary” constraint ensures that the Solver will force each cell in the range to a value of either 0 or 1, which is exactly what we want.
- All the remaining constraints are “…=1” constraints, and these are constraints on the four greyed summation categories above, below, left, and right of our final combined output table. This will ensure that we have exactly one assignment for each class to each spell, each class to each perk, and each spell to each perk.
- We set to the “Evolutionary” solving method, since the other solvers cannot handle a problem like this.

Now run it! Within 5-10 seconds, we should get a table that looks like this:



Combined Output Table

		1	1	1	1	1							
		Warlock	Wizard	Mage	Runecaster	Sorcelator							
1	FireMast	0	0	1	0	0							
1	IceMast	0	0	0	0	1							
1	Mana	1	0	0	0	0							
1	Endurance	0	0	0	1	0							
1	Haste	0	1	0	0	0							
1	Fireball	0	0	0	0	1	1	0	0	0	0	Fireball	1
1	Iceball	1	0	0	0	0	0	1	0	0	0	Iceball	1
1	Spellsteal	0	0	1	0	0	0	0	0	1	0	Spellsteal	1
1	DrainLife	0	0	0	1	0	0	0	0	0	1	DrainLife	1
1	SmHand	0	1	0	0	0	0	0	1	0	0	SmHand	1
10		Warlock	Wizard	Mage	Runecaster	Sorcelator	FireMast	IceMast	Mana	Endurance	Haste		
	10	1	1	1	1	1	1	1	1	1	1		
Total	20												

Problem solved, right?

Well … no. Not exactly.

The three tables are all “solved,” but they’ve been solved independently, so they don’t necessarily make any sense as a whole. The left side of the table shows that the Warlock has the Mana Regeneration perk and the Iceball spell, but the spells/perks table on the right shows that whoever has the Iceball spell should actually have Ice Mastery instead. Similarly, the Wizard has the Haste Perk and the Smashing Hand spell, but the spells/perks table on the right shows that whoever has Smashing Hand should have Mana Regeneration.

In other words, we’ve only implemented a partial solution, because we’ve solved the three sub-tables ([class,perk], [perk,spell], and [class, spell]) separately, and they need to be solved together.

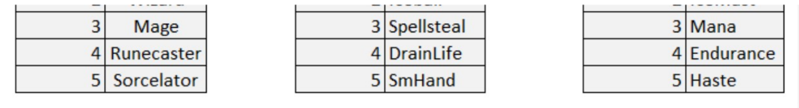
This is a problem! There’s surely some way to fix it, but it would require us to do seriously complex Excel mojo to try to add constraints that would properly correlate across all three tables.

Instead of doing that, let’s see if we can’t find a simpler way to represent the entire problem.

Design Sorcelator Targets Decision Model and Casts “Simplify”

Instead of trying to build a huge table and optimize it, let’s see if we can’t find a much simpler representation. After all, each class should have only one spell and one perk, so we should be able to just set up a “spell” entry and a “perk” entry for each of the 5 classes. Then, we can hold the classes fixed, and try to optimize the 1 spell and 1 perk that belong to each the 5 classes, so that we end up optimizing a total of only 10 decision cells, instead of the 75 yellow decision cells from the previous decision model.

We’ll start by making three tables, listing the classes, spells, and perks.



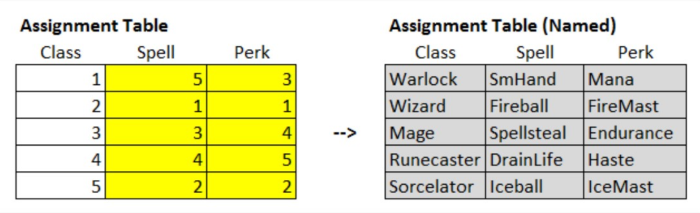
3	Mage
4	Runecaster
5	Sorcelator

3	Spellsteal
4	DrainLife
5	SmHand

3	Mana
4	Endurance
5	Haste

These five tables indicate numerical assignments: we'll use the digits 1 through 5 to represent each of the 5 classes, spells, and perks. These tables indicate which class, spell, or perk a given number stands for, depending on where it's used.

Next, we'll set up an "assignment table" listing a spell and a perk for each class, with each one represented by a number between 1 and 5 corresponding to the appropriate entry in the tables above. This is shown below.



Assignment Table

Class	Spell	Perk
1	5	3
2	1	1
3	3	4
4	4	5
5	2	2

-->

Assignment Table (Named)

Class	Spell	Perk
Warlock	SmHand	Mana
Wizard	Fireball	FireMast
Mage	Spellsteal	Endurance
Runecaster	DrainLife	Haste
Sorcelator	Iceball	IceMast

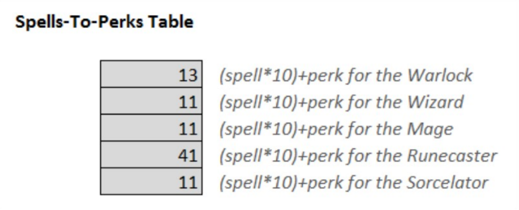
This table shows the current assignments. Each row represents one class (these five are held fixed), and the corresponding "Spell" and "Perk" columns represent the spell and perk currently assigned to that class. Once we find the optimal values for these ten yellow decision cells, we'll have found the five [class, spell, perk] assignments that correctly satisfy our problem criteria, and each of the "Spell" and "Perk" columns should contain each of the digits '1' through '5' in some order.

As you can see, the cells on the left (in the "Class" column of the Assignment Table) are uncolored, since we don't need to change the ordering of the classes; we just need to determine the values of the other two relative to the classes.

The table just to the right ("Assignment Table (Named)") just shows the names of the classes, spells, and perks of the table on the left. It contains Excel functions that translate the numbers in the Assignment Table into text descriptions using the lookup tables at the beginning of this section (so that, for example, the first row shows we are giving the Warlock the Fireball spell and Mana Regeneration perk with this initial setup).

This table neatly encapsulates the [class, spell] and [class, perk] subsections of the logic matrix we prepared at the beginning of this article (the red and blue sub-tables along the left side of the matrix). However, it doesn't encapsulate the [spell, perk] mappings at all (the green sub-table at the lower right corner of the logic matrix). This is a problem, since several of the elements in our problem statement (our constraints) are in the form of mappings between perks and spells, with no mention of classes. So with the way the Assignment Table is currently set up, we can't set up constraints relating to [spell, perk] mappings without testing every pair in the list – and that seems like it could get messy.

Instead, we'll set up a simple table like the one below, that maps each [spell, perk] combination to a number, by treating the spell as the "tens" digit and the perk as the "ones" digit.



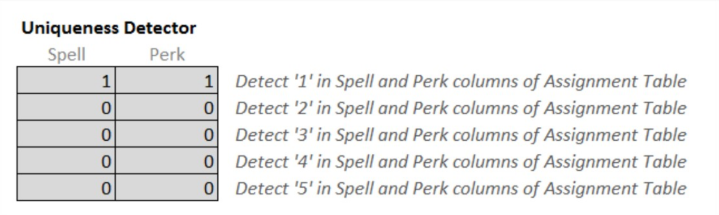
Spells-To-Perks Table

13	*(spell*10)+perk for the Warlock*
11	*(spell*10)+perk for the Wizard*
11	*(spell*10)+perk for the Mage*
41	*(spell*10)+perk for the Runecaster*
11	*(spell*10)+perk for the Sorcelator*

In this way, we can easily set up constraints that search through this table to see if any given [spell, perk] combination does or does not exist.

We also need to ensure that when we optimize our decision cells, we end up with exactly one of each of the values '1,' '2,' '3,' '4,' and '5' in the "Spell" and "Perk" columns of the Assignment Table. This would be easy to do if Excel had a good function for ensuring that all the cells in a given range had unique values, but unfortunately, it does not. Solver also has a special "dif" constraint type that is supposed to ensure this uniqueness, but it unfortunately comes with its own set of idiosyncrasies that makes it more or less useless for our purposes.

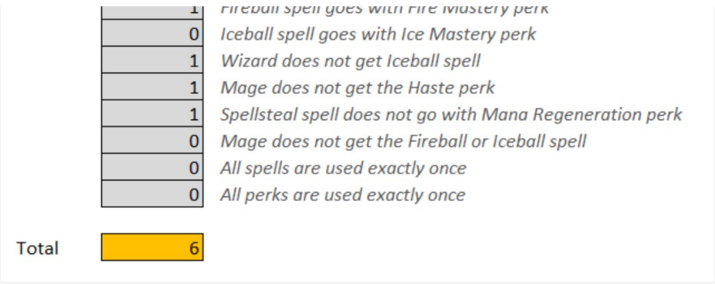
Therefore, in order to ensure that we end up with exactly those five values (in any order), we'll set up a special "Uniqueness Detector" table that detects the presence of '1,' '2,' '3,' etc values in each of the "Spell" and "Perk" columns of the Assignment table. They do this by calling the MATCH() Excel function (and then wrapping it in an ISNA() function since MATCH() returns the special value "#N/A" when the value is not found).



Uniqueness Detector

Spell	Perk	
1	1	*Detect '1' in Spell and Perk columns of Assignment Table*
0	0	*Detect '2' in Spell and Perk columns of Assignment Table*
0	0	*Detect '3' in Spell and Perk columns of Assignment Table*
0	0	*Detect '4' in Spell and Perk columns of Assignment Table*
0	0	*Detect '5' in Spell and Perk columns of Assignment Table*

Finally, we set up a table to test our constraints. Our approach in this case is rather unique; we explicitly set up a cell for every constraint as its own separate entry that will have a value of '1' when the constraint is satisfied and '0' otherwise. Each of these cells (the grey cells in the table below) tests one of the conditions in the problem description.



1	Fireball spell goes with Fire Mastery perk
0	Iceball spell goes with Ice Mastery perk
1	Wizard does not get Iceball spell
1	Mage does not get the Haste perk
1	Spellsteal spell does not go with Mana Regeneration perk
0	Mage does not get the Fireball or Iceball spell
0	All spells are used exactly once
0	All perks are used exactly once
Total 6	

A few quick notes on the setup of these constraint cells:

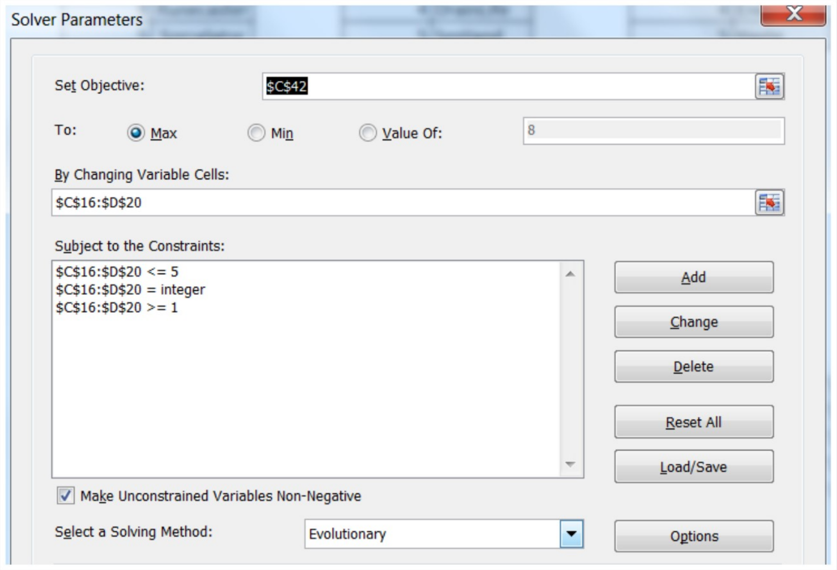
- The constraints related to [class, spell] and [class, perk] mappings are trivial; they just use the Excel "IF()" function to test a certain condition in the Assignment Table (the decision cells).

- The constraints related to [spell, perk] mappings instead query the special "Spells-to-Perks Table" we set up earlier. Just as in our "Uniqueness counter" table, they use the Excel "MATCH()" function to try to test for the presence or absence of a certain [spell, perk] combination anywhere in that table, and then wrap this inside Excel's ISNA() function to test for the special return value "#N/A" if the value is not found.

- Finally, the last two cells ensure that each perk and each spell is used exactly once. They do this by simply summing up each respective column of the "Uniqueness counter" table we set up earlier; if the "Spells" column of the Assignment Table contains the values '1' through '5' in any order, then the "Spells" column of the Uniqueness counter should all have '1' values and should have a total of 5 (and similarly for the "Perks" column of each of those two tables).

Finally, the orange objective cell at the bottom adds up the total number of satisfied constraints. This is different from the objective functions we've used so far; we've generally been trying to minimize or maximize some value, but in this case, we're simply trying to ensure that all the constraints are satisfied. *Any* solution that satisfies all the constraints is a "correct" answer, so we can just count the number of properly satisfied constraints and use *that* as the objective value.

Now that we've got all that out of the way, we're ready to run Solver:



Solver Parameters

Set Objective: $C$42

To: ● Max ○ Min ○ Value Of: 8

By Changing Variable Cells: $C$16:$D$20

Subject to the Constraints:

$C$16:$D$20 <= 5
$C$16:$D$20 = integer
$C$16:$D$20 >= 1

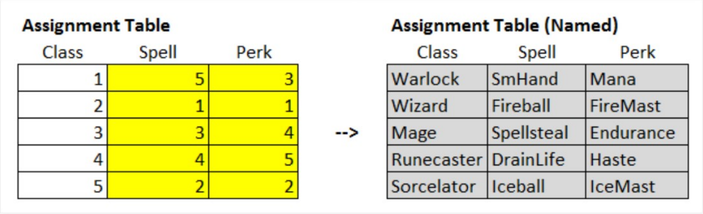
☑ Make Unconstrained Variables Non-Negative

Select a Solving Method: Evolutionary

We target the orange objective cell, as always, and try to maximize its value (knowing it can never go beyond 10). In "By Changing Variable Cells," we enter the cell range for our Assignment Table (including both the blue and yellow cells). These are our decision cells. Finally, we add three Solver constraints to tell it that all the cells in the Assignment Table must be integers between 1 and 5, inclusive.

Finally, we ensure that "Solving Method" is set to "Evolutionary," and hit Solve. It can be time-consuming, as this is a non-trivial problem; it can take 30-60 seconds for the objective cell to get up to a value of 10 indicating that all 10 constraints have been met.

Here's the final, correct solution:



Assignment Table

Class	Spell	Perk
1	5	3
2	1	1
3	3	4
4	4	5
5	2	2

-->

Assignment Table (Named)

Class	Spell	Perk
Warlock	SmHand	Mana
Wizard	Fireball	FireMast
Mage	Spellsteal	Endurance
Runecaster	DrainLife	Haste
Sorcelator	Iceball	IceMast

And there you have it! These tables show the numeric and text versions of the final solution, respectively. We could pretty much take the table on the right, e-mail it to the boss, and call it a day.

disguised logic puzzle. That situation would obviously never happen in real life. Then you went through a whole lot of complicated spreadsheet wrangling to finally solve it when you could have just done the whole thing by hand in ten minutes!"

All true. Guilty as charged!

But the point, of course, is not about what classes get what spells, it's about how we can apply this kind of decision model in all kinds of useful ways you might never have imagined. The journey is far more important than the destination, and this particular kind of journey is one we face as designers in many different forms.

In the next episode, I'll delve into a much more practical example and show how we used a similar approach to design the towers and dropship pads in our iOS/Android game, *City Conquest*.

-Paul Tozour

Part 6 in this series is now available here.

Please note that we have switched to a bi-weekly posting schedule due to time constraints.

This article was edited by Professor Christopher Thomas Ryan, Assistant Professor of Operations Management at the University of Chicago Booth School of Business

Posted by Paul Tozour at 10:19 AM

Labels: Part 5

46 comments:

Joe Osborn August 12, 2013 at 10:38 AM

I've been following this series closely! It's really interesting to see Excel used for stuff like this.

My introduction to game-design-as-search was through the work of Mark Nelson and Adam Smith under Michael Mateas ([1]is a representative assortment); that's part of why I'm at their former lab at UCSC right now[2].

The tool of choice around here for complex optimization problems is answer-set programming, a formalism that's derived from logic and constraint programming. There are fast, free off-the-shelf solvers (we tend to use the Potassco collection from Potsdam[3]), which are purpose-built for doing exactly this kind of search.

I wrote up and commented a quick encoding[4] of this post's problem; six lines of answer-set Prolog for general stuff like "every class has a unique spell and perk", and then eleven lines for problem-specific stuff like the set of classes and spells and the given design constraints. Lots of comments there too.

I think the point-and-clickiness of Excel is useful and interesting, but I wonder if the workarounds needed to get it to work with symbolic problems like this might be a bit confusing. I'd be interested in hearing your opinions on answer-set programming!

[1]. http://tinyurl.com/mzdng48
[2]. http://eis.ucsc.edu
[3]. http://potassco.sourceforge.net
[4]. https://gist.github.com/JoeOsborn/6213131

Reply

Replies

Paul Tozour August 12, 2013 at 11:06 AM

Thanks so much, Joe! I will check this out right away!

Joe Osborn August 12, 2013 at 11:39 AM

You might be interested in its optimization statements as well, where you can specify e.g. weighted sums to minimize or maximize.

ASP can attack problems in complexity classes up to NP^NP, for example "there is some puzzle I can generate such that all solutions require that these five pieces are used". So it's not totally general purpose, but many useful optimization problems fall into NP or NP^NP.

Paul Tozour August 13, 2013 at 11:41 AM

This is incredibly useful stuff; I cross-linked to this post from the version of the article on my Gamasutra blog so people could see your comment.

Very impressive to see how clasp can instantly solve this and tell me that there's only one correct answer and give me that answer.

I've read up on answer-set programming a bit thanks to some of Michael Mateas' articles but I'm not very adept with it yet. My main sense is that it clearly has huge potential. However, I'm not yet sure which of the kinds of problems that it can address, based on the articles in the series so far and the 4-5 additional articles I have planned.

So if you'd like to fill in your thoughts on which of these are solvable with answer-set programming and how you'd go about doing that, and/or give your own solutions as you've done here with this example (thanks so much for that!), that would be great!

But my bigger concern is scaring off designers from using these approaches, since I feel like the level of complexity of the solutions I'm giving them in Excel and Solver is already close to the limit and overwhelm some people. Most designers don't have the patience or the intellectual curiosity to learn answer-set programming.

So I think it's best to stick with Excel Solver for now, but I'd definitely like to see if there would be a way to some answer-set programming into the series in the future without making it too complicated or cumbersome.

I have at least 4-5 more articles in the series that will be based on Excel and Solver, but there might be room for something more after that.

In any event, I'd certainly like to learn more about it just for my own use and expanding my own understanding.

Is there a good guide to answer-set programming you'd recommend? I'm reading through "A User's Guide to gringo, clasp, clingo, and iclingo" right now but I'm not quite getting how things are defined and which things are built-in keywords vs user-defined, and how the user-defined facts work.

Joe Osborn August 14, 2013 at 12:54 PM

I wonder if it's the syntax or the semantics of ASP that make it seem hard to approach. One metaphor I like is "generate and test", because the process feels to me like producing a state space and then slicing it down; the idea of state spaces seems like it might be intuitive for designers, but I'm so far gone I don't trust my instincts on that.

As for references, [1] is pretty good, and there's also the book [2]. For applying it to games, Adam Smith's TCIAIG paper is a great introduction [3].

[1]. http://potassco.sourceforge.net/teaching.html
[2]. http://potassco.sourceforge.net/book.html
[3]. https://blog.itu.dk/MPGG-E2011/files/2011/09/tciaig-asp4pcg.pdf

Joe Osborn August 15, 2013 at 12:17 PM

Here it is: https://github.com/JoeOsborn/dmogd

In addition to the encoding above, I've done two versions of Supertank. As it turns out, math-heavy problems hit one of stock ASP's weak points, so I did an encoding with "constraint logic answer set programming" via clingcon, another tool in the Potassco collection. The purely symbolic encoding takes about 90 seconds to find the optimal answer, and the constraint logic version takes under half a second.

Both encodings use more code to define the table than they do for their actual logic; it's easy to imagine some preprocessing that gets around that.

I'll be building up more text around these examples in the future, but for now I'll just post them as they are (with some comments).

Paul Tozour August 15, 2013 at 12:20 PM

Awesome!

Reply

planateree August 14, 2013 at 12:28 PM

Thanks for this post, it was an interesting read. Had to try solve this problem in python (that I normally prefer over spreadsheets for almost anything) and came up with this solution (using a library called python-constraint I never heard about before):
https://gist.github.com/lifelike/6234498

Got the correct answer first time I ran it. Was quite straight-forward to figure out how to set up the constraints too, but the code could probably be simplified a bit.

Reply

Replies

Paul Tozour August 14, 2013 at 1:12 PM

Thanks, planateree! That's pretty awesome -- great that you can solve it with such a tiny Python program.

Reply

Dan Kline August 21, 2013 at 9:26 PM

Thanks for writing all this up Paul. Paul, Level 5 Excel Wizard!

Reply

Replies

Paul Tozour August 22, 2013 at 11:01 AM

My pleasure! Glad you're enjoying it, Dan ... there's a lot more on the way! Part 6 should drop on Sunday the 25th.

Reply

Unknown August 22, 2013 at 12:08 PM

I've also been following his series with interest. After reading this post and the previous post, one question comes to mind: Could this "balancing" exercise be applied to a case where the player is a defender and the game is an attacker that attacks in waves with multiple targets?

For example, the classic and simple tower game (only 1 tower allowed to defender and 1 weapon type at a time) where the player must shoot oncoming tanks that attack in waves and in each level each wave increases in size and in variety of tank type.

I was thinking of approaching it from a turn-based perspective, but it's the 1-versus-many perspective that throws me. It seems to me like there is a "balanced" number of tanks but then that balance is purposely upset and it becomes a resource allocation issue for the defender to solve in order to keep up with the increasing threat.

Any hints you can drop to me or maybe a little push in the right direction?

Reply

Replies

Paul Tozour August 22, 2013 at 12:11 PM

Heya Thomas! I'm not sure it's the best fit for *this* particular set of techniques, but it's definitely something you could approach with the decision modeling & optimization techniques I described in parts 2-4.

I can take a look at possibly doing some modeling of this sort of thing in a future article, but I'm not sure I completely understand the game you're referring to.

Is it a tower defense type of game, something like Kingdom Rush http://armorgames.com/play/12141/kingdom-rush? Or something else?

Unknown August 22, 2013 at 3:57 PM

Paul...I don't think that Kingdom Rush is exactly the same, though it is very similar. I think one defining thing about Kingdom Rush that makes it different is that it sends the attackers down a "chute" whereas my problem is dealing with waves of attackers that can be spread out.

Probably a slight differentiation - but maybe not.

https://play.google.com/store/apps/details?id=com.nenoff.ballista

Paul Tozour August 23, 2013 at 6:40 AM

Hi Thomas -- thanks; I checked out Orcs Invasion.

I'm not sure there's a good way to model this. Since it's an action game, I don't think there's any way to realistically model the player's reflexes as the different waves of enemies attack.

What is the question you're trying to answer? I am guessing you're trying to figure out the appropriate number of enemies to throw against the player in each wave, especially toward the end of the game? Is that correct?

If so, it's fundamentally a question of skill. I think you're just going to have to get an alpha version of the game and sit down with some beta testers, and make a bell curve of their skill level with the game so you can get a sense of what the upper and lower bounds are.

Unknown August 23, 2013 at 9:46 AM

Paul,

I agree the the reaction time would be impossible to model - though I look at it as the same as casting time for a mage or the draw time for an archer.

What attracted me to the most current models in your series is that you basically turn it into a turn-based game and I think that that would work to come up with an alpha for mine as you suggest.

The main problem is that every few levels there should be a balance - meaning the current weapon and strength of the walls will be equal to the wave of attackers - in other words, that level will be "easy". I want to create the base or "alpha" version with a weapons tree and attacker tree that has a 1:1 relationship. So the "basic bow and arrow" would be balanced with the "basic attacker wave" and so on.

Each succeeding level is then purposely imbalanced (determined by actual play testing) until a new balanced level is introduced (signified by the introduction of a new and more powerful type of attacker) to build the tension and then relieved.

This is why I said there seems to be a resource allocation problem for the player as well; if he does not upgrade his weaponry then by the 10th level no amount of quick reflexes will help him survive.

In short, if we just look at the game as a balance between defender (walls for HP and weapons for damage, speed and range) and attackers (speed, range, damage, HP)and then use your-turn based approach this might work.

Thoughts?

Paul Tozour August 24, 2013 at 3:09 PM

Hi Thomas. Sounds like an interesting idea.

At this point, it's impossible for me to say how well this would work -- and whether you really could rule out human skill and reaction time -- without knowing more about your game.

Reply

Unknown May 30, 2014 at 4:24 PM

Hi Paul.

This is an awesome series. Extremely well written & designed, feel like I already learned a very valuable design tool.

I've tried this slightly differently then you did in your 2nd example. Interestingly I cannot get my Excel (2013) to find the correct solution. If I copy & paste your solution in then everything works - but Solver doesn't manage on its own.

Any ideas why? I've uploaded the spread sheet here: https://www.dropbox.com/s/makd9o8ywol4jxx/spells.xlsx

Cheers

Alex

Reply

Replies

Paul Tozour May 30, 2014 at 4:39 PM

Hi Alex -- Thanks, and I'm glad you're enjoying the series!

I'll try to take a look at your spreadsheet sometime later this week -- have been crazy busy starting up a new game studio in Austin.

Unknown May 31, 2014 at 1:40 AM

Hi Paul,

thanks for the super quick reply. Don't worry, please take your time. It's not pressing at all, I was just curious about the different behaviour.

Good luck with your studio!

Alex

Reply

Amar Rajput November 6, 2017 at 3:08 AM

Nice Blog! Experience the Best Assignment Writing Services at Assignment Help Sydney Australia

Reply

Olivia Crew August 21, 2018 at 12:02 AM

Excellent information on your blog, thank you for taking the time to share with us. Amazing insight you have on this, it's nice to find a website the details so much information about different artists.
visit here:- my assignment help

Thank You So Much For providing the important information.
I am Max Willor, SEO Expert in a reputed company. All of your information is very useful to me.
I am providing Assignment Help Australia service to students.

Reply

Harrison Aiden August 30, 2018 at 1:18 AM

that is very nice post and very informative and knowledgeable post. So thank you for sharing with us.
we have some related posts.
visit here:- coursework help

Reply

Dylan Eales January 31, 2019 at 6:36 AM

Thanks for this informative content. It's really good. Actually, I want to share some thoughts and reviews about an assignment help Australia company in Australia and the brand name is SAMPLE ASSIGNMENT. Here, I am working as an Academic Expert. To look at our online academic assistants who provide reference assignment including Accounting, Management, Finance, IT, Economics, Computer Science, Nursing, Marketing; all Academic subjects to University Students all over the Australia or even worldwide, Here is a big online assignment help providers who help students to get HD grades according to assessment guidelines and instructions. Those who are searching for assignment help Melbourne, Perth, Brisbane, Adelaide etc. get a touch with Sample Assignment - the No.1 Assignment provider. Any student really wants to buy an assignment at the cheapest price goes to our branded website and has a look and opts our amazing and informative services, you can avail of our convenient online assignment help and samples available on our website for free. You can download it if you want. Are you ready to get 100 out of 100 in your university assessment? We assign the best writer according to the subject for your academic problems and provide support in assignment writing services. Our Customer Care Executives are available 24*7 hours to assist you in the best possible manner. Phone calls and emails are the best methods used by online assignment maker.

Reply

SophiaMiller February 24, 2019 at 11:30 PM

Hello
This is sophia miller, Thanks for Sharing.

Get the Best Assignment help in Australia.

Reply

Andy Wilson April 16, 2019 at 8:06 AM

If you are a worry with the thought of 'who can do assignment for me before deadlines' then you are at the right place. We, at Student Assignment Help acts as helping hand to stressed students and provides the best solutions.

Reply

Ronnie louis April 27, 2019 at 12:00 AM

This is Ronnie louis, Thanks for Sharing this.

network information security assignment help

Reply

prashant July 4, 2019 at 3:23 AM

Here I am sharing a very useful resource for students looking for assignment help. ProAssignmentHelp providing all sort of assignment writing in various subjects like as : Electrical engineering assignment help , electronics assignment help , MATLAB assignment help , management assignment help , etc.

Reply

Anonymous July 24, 2019 at 12:23 AM

Thanks for Sharing the useful and informative information, i am also a blogger, writing on different - different topics, if you are interested to reading my website article's please check once as - finance assignment help

Reply

Robin July 26, 2019 at 1:40 AM

mba
bcom
scope after bsc
courses after bcom

bachelor of law

llb academic

post graduate diploma in computer application

blib
pmkk
ba full forrm

Reply

Unknown July 30, 2019 at 7:02 AM

I loved the article, keep updating interesting articles. I will be a regular reader…know about anveshi jain.

Reply

Vehicleyard August 4, 2019 at 11:30 PM

purchase the top bluetooth obd2 scanners
purchase the top obd2 scanner
motorcycle maintenance for beginners
car maintenance for beginners

ISBM University is a new-age university committed to provide innovative learning programs like Diploma in Journalism and Mass Communication, Executive Master of Business Administration, Bachelors in Commerce , Diploma In Business Administration , Bachelors in Business Administration,Bachelor of Science in Computer Science , Diploma in Computer Application, Bachelor in Computer Application and Post Graduation Diploma in Computer Application to meet the requirements of the industry. ISBM University programs are research-driven, skill-based and knowledge-intensive.

Reply

Ideal Assignment Help November 26, 2019 at 11:31 PM

Our writers may be your best guides when it comes to Computer Science Assignment Help or thesis writing on other topics. They need had many years of experience during this field and are subject matter specialists who have the experience and ability to handle every kind of topics. Our assignment writing specialists are cognizant of the different referencing styles and can make sure that your assignment format is keep with the set tips. You can talk to us and write to us any time of the day and night as a result of we promise to be by your side 24×7.

Reply

Max Willor December 3, 2019 at 1:53 AM

I recommend that everyone should seek assistance from Assignment Help Australia for the best academic help. There are professional experts from various esteemed universities in almost every field. They are capable to write the premium quality of assignments according to the requirements of students and university norms.

Reply

john amber December 16, 2019 at 11:51 PM

I suggest all members choose MY Assignment Help Australia for the best guideline in your academia. The perfect and expert assistant your requirement and learning assignment project by the university.

Reply

Olivia Crew December 18, 2019 at 12:11 AM

Excellent information Providing by your Article, thank you for taking the time to share with us such a nice article. Amazing insight you have on this, it's nice to find a website that details so much information about different artists. Kindly visit the LiveWebTutors website we providing the best assignment help services in Australia.

For More Info: Law Assignment Help

Reply

Ashley January 31, 2020 at 3:53 AM

Amazing post. I am so impressed. Could never think of such a thing is possible with it...I think you have a great knowledge especially while dealings with such subjects.

Biotechnology Assignment help
Computer Science Homework Help
Architecture assignment help
Biology Assignment Help
Engineering Assignment help

Reply

John Smith February 3, 2020 at 5:15 AM

Are you looking for write my essay for me justquestionanswer.com is providing write my essay services in the USA from the qualified expert. Our 3000+ experts are available 24x7 to help the USA students and bring forth the peerless Science end of the day without asking an amount that exceeds the budget.

Reply

Sam February 6, 2020 at 11:35 PM

My Assignment Help Australia has pioneered for its quality services, that has brought closer to our customers. We do not only wish to gather vast populace, but also aim to connect with them as their assignment help partners.

Reply

tom554 February 7, 2020 at 4:22 AM

Get best assignment help in Australia from top professionals....For more information do visit our website....

Reply

William Smith February 13, 2020 at 4:31 AM

My self, William Smith Associated with Livewebtutors.com. Which Is Known as A Renowned Assignment Help Australia Company. I Have Been Working as A Full-Time Academic Consultant from Last Ten Years. Livewebtutors.Com Is an Experienced and Highly Reputable All Assignment Help Provider in Australia, Specialized in Academic Writing. We Offer High-Quality Service and Supply Free Plagiarism Report for All Assignments and Best Price.

Reply

Best Travel Portal Development March 17, 2020 at 3:15 AM

Hotelbeds is a world leader in travel and technology and it is known for its API Interfaces. It provides hotel reservation services to B2B and B2C travel features all over the world. Teamindia Webdesign provides the best online services for Hotelbeds API Integration at pocket price.

Reply

Assignment Help Australia April 2, 2020 at 2:20 AM

Use Assignment Help services when you have issues in writing your academic papers for Australian universities. Take the assistance of online tutors to get excellent academic writing for your project submission at my time and make your submission countable for the entire session.
My Assignment Help
Assignment Help Online
Online Assignment Help

Reply

the situation of students to balance the extra curricular activities, their hard work towards their vision and dreams, the expectation of higher and the best grades while staying calm overall. The experts are not just the writers for their subjects rather they are the known industry consultants in their respective fields. Moreover, they are the pillars of Assignment Work help who put their efforts to provide the solution in an writing service domain.

online assignment writing help
assignment help australia
assignment solution australia
homework help australia
online assignment help
best assignment help
assignment work help usa
assignment writing services in USA

Reply

Enter your comment...

Comment as: Ahmad Al-Kashef ((Sign out

Publish Preview □ Notify me

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.