Adaptive Game Difficulty

By

Jon Vegard Jansen, Robin Tollisen

Supervisor: Sondre Glimsdal

Project report for IKT411 in Spring 2013

Faculty of Engineering and Science
University of Agder
Grimstad, 7th of June

Status: Final

Keywords: E-learning, gamification, unsupervised online learning, stochastic hill climbing, maximize fun

Abstract:
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Table of contents

1 Table of figures

2 Table of tables

[1]

3 Introduction

Today, there exists more commercial video and computer games (henceforth: games) than ever before, and the number of persons owning a smart-phone or personal computer is increasing. Many students and pupils neglect their studies in favor of watching television or playing games, maybe because they consider it more fun. We believe that through the use of E-learning and gamification, which enhances traditional learning methods with elements from games, it is possible to make any learning process more fun, resulting in more interested students and better grades.

As each student is unique, we believe that they also have different learning methods that work best for them, which is why we have chosen to look into adaptive games. In this project we research whether a game can be made, such that it adapts to each player on the fly.

3.1 Background

A lot of research has already been done on the area of E-learning, and there exists multiple definitions. We have chosen to use the following one, because it emphasizes the individual student, which goes hand in hand with individual adaption of application.

"We will call e-Learning all forms of electronic supported learning and teaching, which ... aim to effect the construction of knowledge with reference to individual experience, practive and knowledge of the learner..." [2].

Gamification is a concept that has been around for a long time, and has increased with popularity since 2010. [3] [4]. We present the two following definitions, as we think both of them show that gamification can be used well in conjunction with E-learning.

"The process of game-thinking and game mechanics to engage users and solve problems." [4].

"A process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation." [5].

The latter definition is directed towards service marketing, but if one reads 'service' as 'learning process' and 'user's overall value creation' as 'learning outcomes', we believe that gamification is something that could be used together with both tradition learning methods and E-learning.

If one is to create a game that is meant for learning purposes, it is important to note that the game needs to be fun. [4]. If a game is not fun, it will not be able to educate either, because players lose interest in games the same way as students lose interest lectures, should they be boring. This is why it is important to figure out how it is possible to maximize the fun in any kind of game, thus keeping the interest of players. Since games appeal to a wide variety of players, the perceived level of fun may often vary among its players, due to personal preferences. In order to accomodote for such varying preferences, traditional games have often implemented several levels of difficulty the player may choose from, as well as providing the player with several settings or options for how they want the game to behave. When it comes to difficulty, other work has already been done on how to scale and adapt this to the level of player, in order to make even games, but in order to make a game fun, we believe that more than difficulty needs to adapt to individual preferences, thus more aspects of a game should be able to change on the fly. [6].

3.2 Problem Statement

Our project is to research whether it is possible to create a game, that uses an unsupervised online learning algorithm, which will adapt the game to the user's preferences based on limited user feedback.

3.3 Problem Solution

We have chosen to create a game using the tower defence genre, as this requires somewhat less work and graphics than many other genres, as well as having a lot of possible parameters that can be tweaked in order to adapt the contents of the game (henceforth: gameplay). The game uses a modified stochastic hill climbing algorithm together with some user feedback after each game, to adapt itself towards the individual player.

4 Stochastic hill climbing

A hill climbing algorithm is an iterative improvement algorithm, searching for a local maximum. It tries to maximize a function f(X), where X could be a node, state, position or, in our case, a vector of parameters. It compares neighboring nodes to the one it is standing on, and checks if they are an improvement on f(X). If one of the other nodes is improving the situation, then the algorithm moves to the new node, commonly the best node, and repeats the process. The process is repeated until no more improvements can be found, thus the algorithm has found a local maximum. [7]

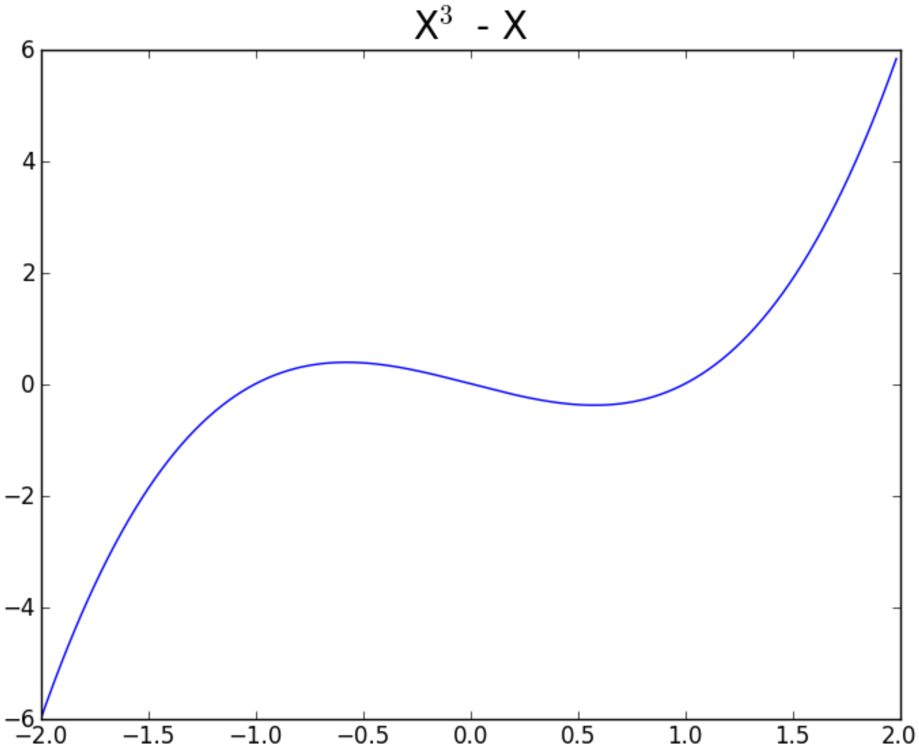

Figure 1: A function that has a local maximum

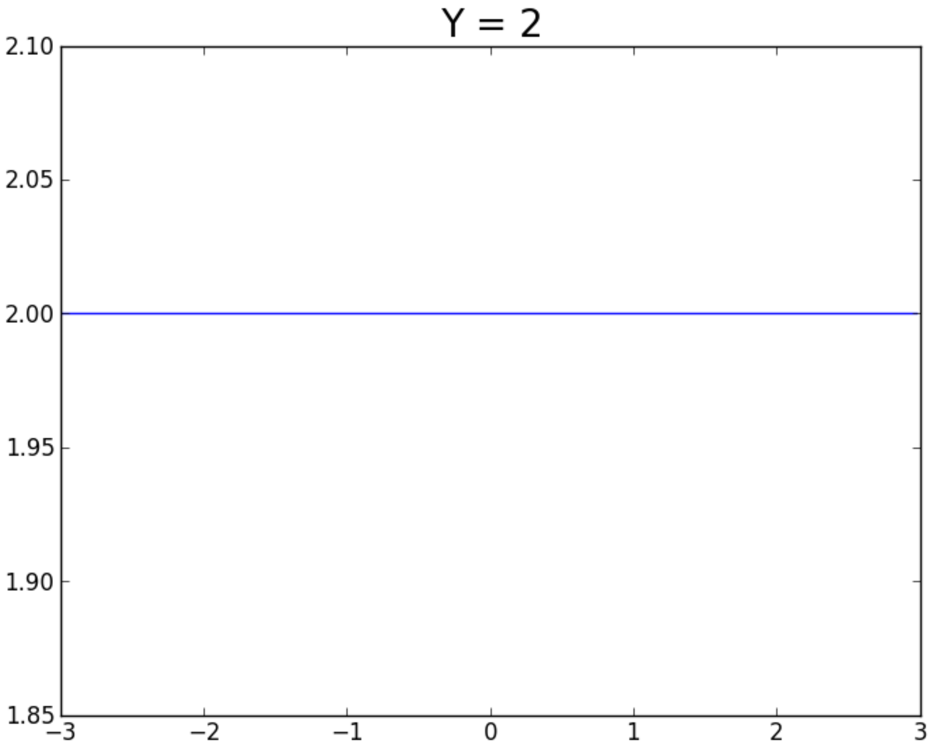

Figure 2: A function that is too flat to search on

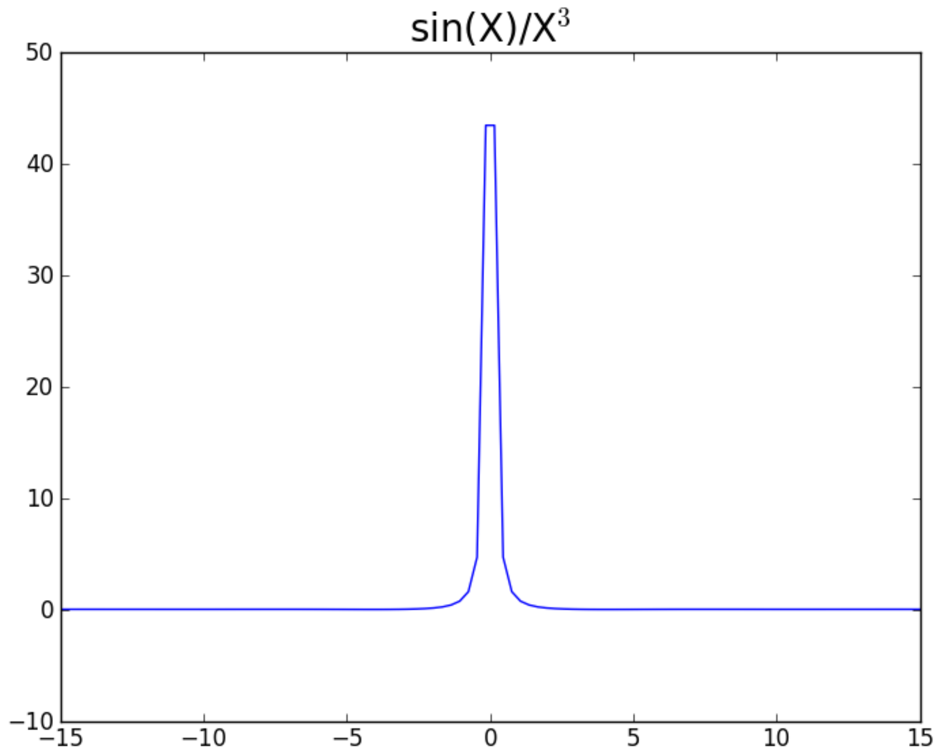

Figure 3: A function that is very steep

One issue with this algorithm, is that it can become stuck on a local maximum, which may not be the global maximum, as in *Figure 1*. One way to reduce the probability of this to happen, is to allow the algorithm to jump larger distances each jump, or to make sure that the landscape is convex. Another issue shown by *Figure 2* is that the search area may be flat, that is, each node is as good as all its neighbors. In this case, any jump will not bring any improvement. A third issue in *Figure 3* is that a node may be lying on a very steep ridge, that is, the 'area' of parameters that give a good node is very small, and can thus be hard to find. [7]

There are different variants of hill climbing algorithms, with different ways of deciding where and when to jump. The stochastic hill climbing algorithm, closely resembling simulated annealing, chooses a position at random, then evaluates whether it is an improvement, and if it is, jumps to the new position. If the new position is not an improvement, it will go back to the last one. With stochastic hill climbing, it may be wise to set a specific maximum jump distance, and let this decrease over time. This allows the algorithm to converge or settle on a position after some time. [7]

4.1 Stochastic hill climbing example

Here is an example for how to use stochastic hill climbing for how to find the maximum value of the function f(X) = 2 - X^2. which has a maximum value of 2. We initialize the maximum jump distance (in x-direction) to a value of 1.0, and let this decrease by 0.1 with each jump.

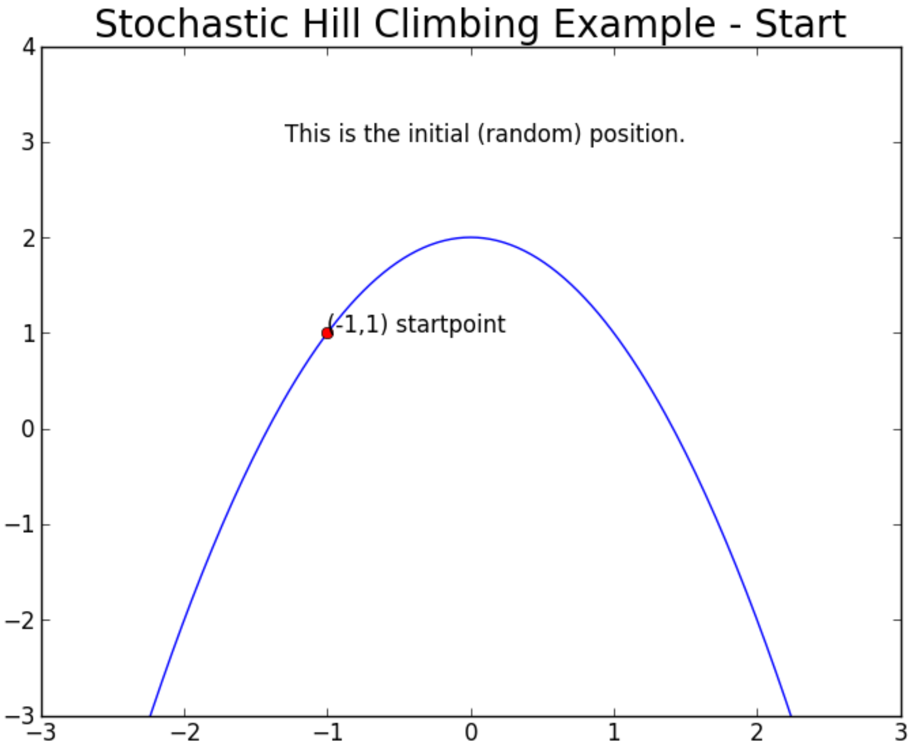

Figure 4: The startpoint for the search

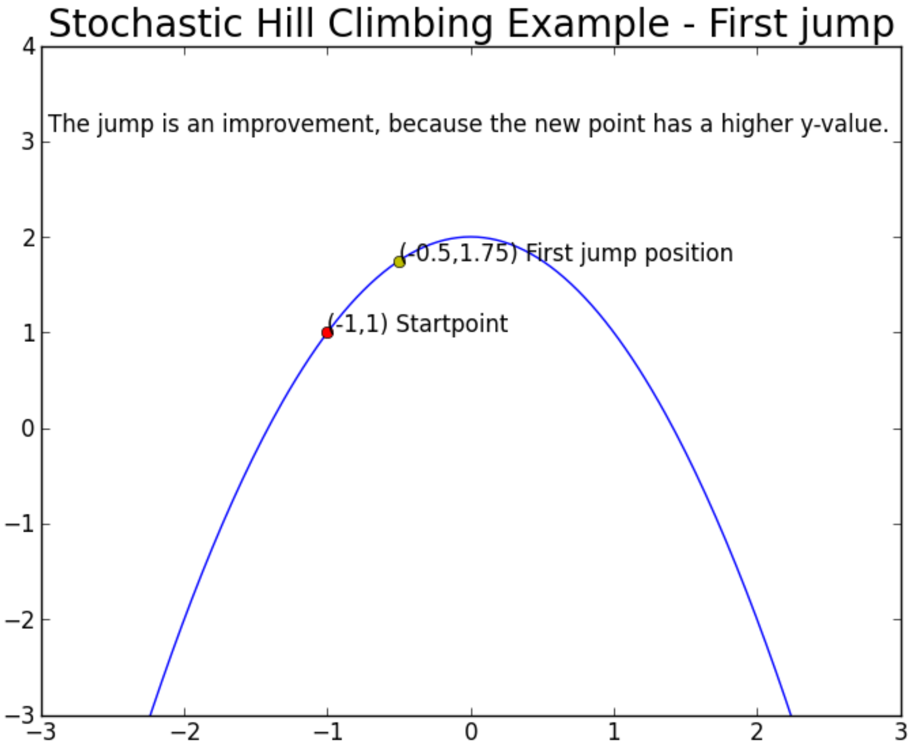

Figure 5: The first jump

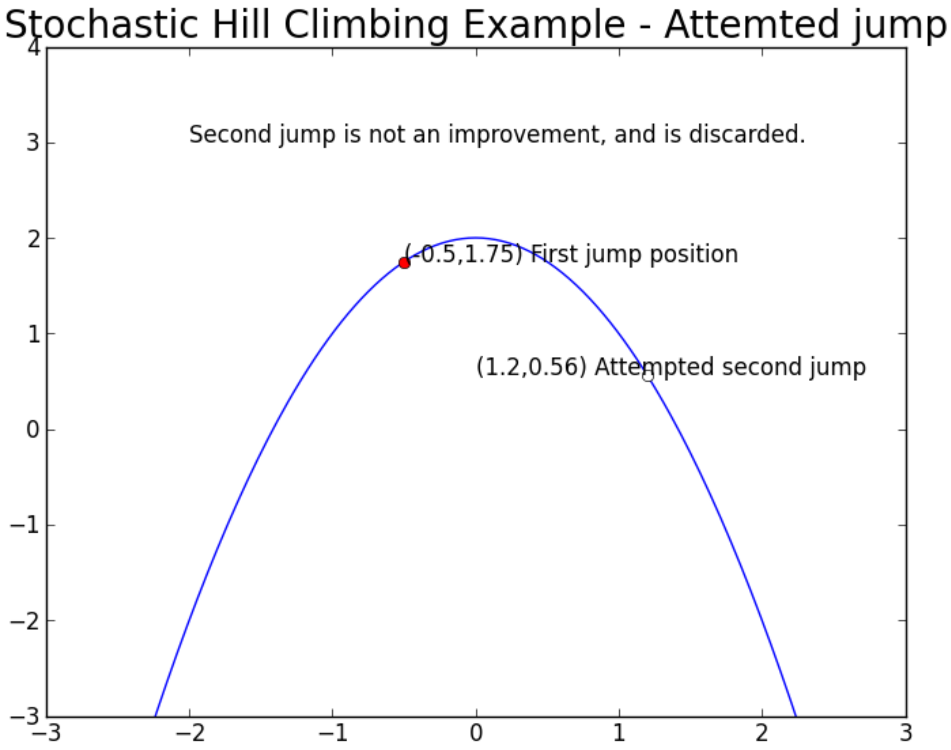

Figure 6: An attempted jump

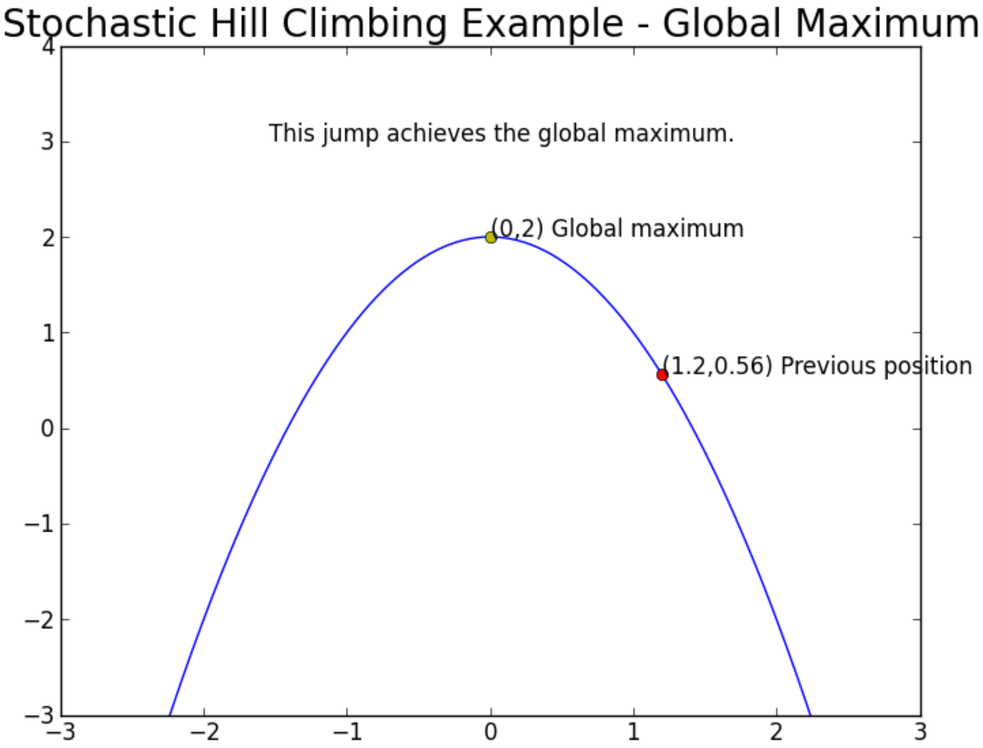

Figure 7: Achieving global maximum

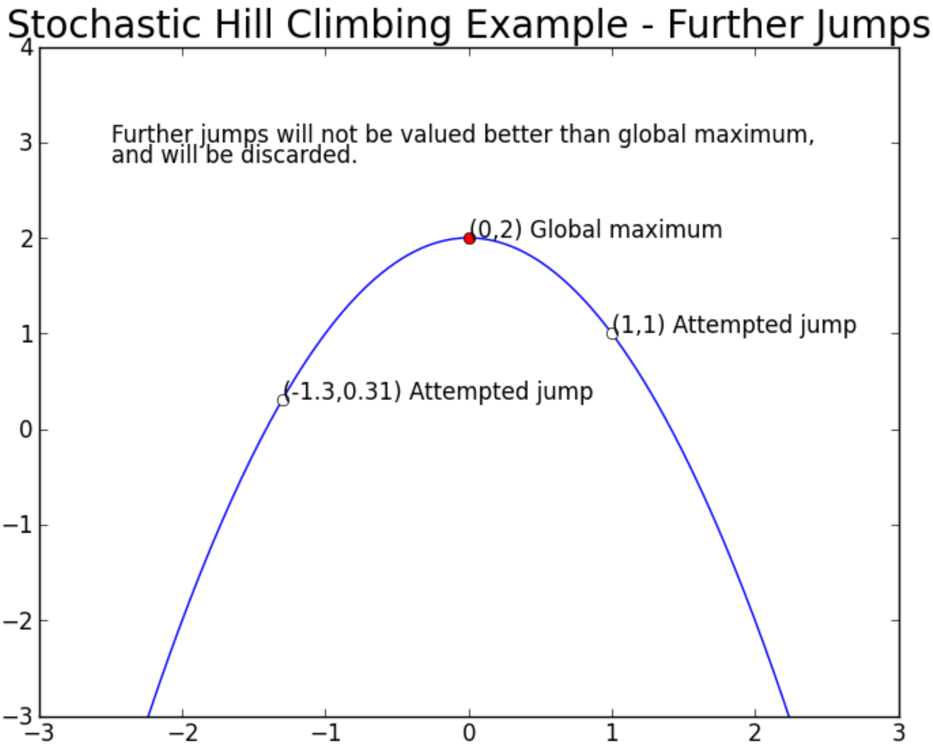

Figure 8: Further jumps are discarded

This is a short example of how stochastic hill climbing will eventually find the local, or global, maximum on a graph. This example is very short, compared to how many jumps it would normally take, but this is actually possible.

With a starting maximum jump distance of 1.0, decreasing by 0.1 with each jump, the algorithm will have converged after ten jumps, hopefully at an acceptable position. In a game setting, it may be wise to let the maximum jump distance never decrease to zero, because it can make the game feel more lively.

5 Our algorithm

Our algorithm works with a set of parameters which affects the game, these parameters are listed in [8]. We have done some testing and found some values for these which we think make an ok game for most people. After each game there is a questionnaire which asks "How fun was this game?" and "How difficult was this game?", which is then saved and used to create the next game. We save the value we receive in "How fun was this game?" as a metric, and use it to determine wether to jump from the current or the last game.

We scale the difficulty based on certain paramteres, and "fun" based on other parameters. The parameters which fall into the category of scaling fun are related in pairs of a "positive" and a "negative parameter such that if one jumps, the other one jumps equally as much, such as to keep difficulty even. By positive parameter we mean one which when scaled up would make the game easier for the player, and a negative would make it harder for the player when scaled up.

6 Game Description

This chapter gives some pictures and a short description of our game, as well as some general information about the game genre, namely tower defence.

6.1 Genre

The game we created to demonstrate our algorithm is a real-time strategy game, more precisely it would be refered to as a Tower Defense game. The goal of the game is to build tower tactically accross a map to prevent computer controlled enemies from reaching the other side.

6.2 Our game

Our game is very similar to classic tower defence games, in that the player has to stop different kinds of enemies from reaching the destination on their path, by using different kinds of towers. Our game differs from other tower defence games on the following:

- There are no scripted or static levels, since each game is a step closer towards the player's 'ideal game', that is, the game that is perceived as the most fun.
- Other than that the game learns after each level, the player has nothing that is carried over to the next game, such as tower access, special items or game progress in general. The player simply plays successive games until he or she quits.
- Modular towers and enemies. This is not necessarily new to the tower defence genre, as not all tower defence games may have pre-defined enemies and towers, but this is something that can be used in order to adapt the game towards personal preferences.
- (Guided) Randomly generated maps. Not necessarily new to the genre either, but is another useful technology that can be used to adapt the game to the player. Note that our game did not use this for game adaption, only to make the game seem less static.

7 Testing

Our testing is split into two parts. The first part contains two case tests of the algorithm, and the second part contains tests from our fellow students and friends.

7.1 Case tests

This is how our algorithm performed.

7.2 Fun tests

After we finished the game, it was mainly sent to friends at the university. They were told to test the game for as long as they wanted, and when they were finished, they sent us a logfile containing some information about their games. They were also asked to rate how fun they thought the game was in total.

As you can see in the table, the game was rated on average. ....gfdlgdfg. Which is not exceptionally fun, but not boring either.

f

8 Discussion

One of the main issues with our process has been that it is hard to measure fun, it is also possible that our game in itself was not a 10/10 in fun, which can influence the results and not tell us much about the algorithm. While our game is entertaining for a short while, it is certainly not an industry grade game. Our main feedback for how well the game worked is the ratings we get from players, and from that we are satisfied with how fun the game was perceived to be.

Another problem with using the algorithm in a game is that the changes may be to subtle for a player to notice when jump distance shortens. The algortithm requires positive feedback when it goes in a direction the player likes, but if the change is not noticable the player may give it the same score and thus have a decent chance of ending up jumping from his old game.

The algorithm needs to be more refined, and would work much better if we could create more specialized sensors, and do more testing as to what different results indicate about the player. If we could do stochastic hillclimbing based on metrics created by how the player is playing, on few parameters for each part of the algorithm, we believe we could get much better results. However, as this has been the first venture as far as we know into creating games which adapt based on fun, we are quite happy with our result.

Gode ting er nettopp at det hele funket.
Dårlige ting er at vi ikke kunne lage et godt nok spill, som kanskje kan ha noe å si, ettersom vår research bit er kun én del av hva som gjør et spill gøy.
TD er kanskje ikke beste sjangeren.

9 Conclusion

With a more refined algorithm we believe this could be highly useful in a game, however there needs to be more sensors to detect which parameters the players were satisfied with, and which he was not. There are also interesting applications for this sort of algorithm in e-learning, if we can detect how to make an e-learning game fun for a player and also scale the difficulty to their ability we may create an effective game for learning.

For this initial test of such an algorithm in games we would also probably have gotten a better result choosing a simpler game than tower defense, when the game uses a stochastic hillclimbing algorithm on too many parameters it is unlikely that all will go in the direction the optimal game resides at once. Which is why as we mentioned that we should create more a more specialized algorithm which can look at very few stats and detect feedback automatically through how a player plays, rather than requiring a questionnaire at the end.

Dette kan bli bra i fremtiden, om vi kan ta det videre. Vi vil nok bruke det i en annen spillsjanger, hvor vi ser mer forskjell.

10 Future Work

The algorithm works ok, but it needs to be more refined. We believe that with some work refining the algorithm, and doing more testing with different sensors to see if certain feedback can be interprented to liking certain things in the game, we can create a great algorithm to maximize fun even in games with many parameters.

We would like to try to implement this sort of algorithm in a game related to e-learning. Currently one of the problems with e-learning games is that they feel to much like learning and to little like games. If we could be able to create a game which can maximize the fun a player has, while also scaling the difficulty appropriately we believe we could create an effective learning tools for many skills.

JVs notes on future
- e-learning with individual adaption
- try it out on another game, in other genre
- spillet: Gjøre noe med diggers, earthquakes.
- Bedre grafikk i spillet, utvide content
- Guide søket til algoritmen

11 References

IEEE, ACM, Springer

[1] Husk class='notoc'
[2] Tavangarian, D., Leypold, M. E., Nölting, K., Röser, M., & Voigt, D. (2004). Is e-learning the Solution for Individual Learning. *Electronic Journal of E-learning*, 2(2), 273-280.
[3] Marczewski, A. (2012). *Gamification: A Simple Introduction.* Andrzej Marczewski.
[4] Zichermann, G., & Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps.* O'Reilly Media.
[5] Huotari, K., & Hamari, J. (2012, October). Defining gamification: a service marketing perspective. In *Proceeding of the 16th International Academic MindTrek Conference* (pp. 17-22). ACM.
[6] Spronck, P., Sprinkhuizen-Kuyper, I., & Postma, E. (2004). Difficulty scaling of game AI. In *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAME-ON 2004)* (pp. 33-37).
[7] Russell, S. J., & Norvig, P. (1995), *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall
[8] Some Chapter