Adaptive game engine

#### Introduction

By Volkovyy Pavlo

# Part 1: The problem

Let’s imagine you want to create a game. You have a brilliant idea and you’re sure it will find resonance in millions of people after you release it to the light. You read something about start-ups and decided to proceed to action. Here are the steps you would probably take.

creating the game:

1. **Talk to other people.**

The first thing you do when you have a great idea is testing it on potential customers, right? So you go outside, grab the first encountered human being and ask one what does one think about your genius thought. In best case scenario the answer would be: “What on earth you are talking about”. This is where this phase actually starts.

The best thing you can do is to give your evaluator the quick intro about what the goal is and how great will be the result. 2 minutes later you will proudly take a breath after finishing the last sentence. Then you may even get the positive answer on your “is it all clear?” kind of question. Congrats, you made it to the interesting part of the process. The part that is as inevitable as the existence of the word “individual”.

The key to this step is to eliminate all the differences in your brain and the brain of your interlocutor. Unfortunately, if the only way to do this was to shoot that parts out of one’s skull, the only efficient way would be to cut one`s head off straight from the beginning. That’s why what you will want is to continue this process until you make that person to answer correctly to least 50% of your questions (you can put other specifications as well. For example, 7/15 questions asked within 2.5 minutes, etc.). The good thing about this process is that you do not have to actually rewrite his mind (which is in between 10 and 2,500 terabytes). Instead of drawing the picture, you just point the canvas with a brush, see the flow of your paint and correct the ‘errors’.

OK, now you are sure that human finally understood your idea. Great! It’s time you can get meaningful review. By running from one person to another you receive more and more feedback on your future project until it’s late or you realized that this is a working day.

During dinner/supper you are looking through those several papers you wrote, thinking that if it is fairly easy to make so much people understand your thoughts, how hard can it be to implement them in an actual product using modern high-level programming languages and powerful game-engines? You quit your current job and plan to start processing your dreams to reality next morning.

1. **CHOOSE YOUR WEAPONS**

It is 7 a.m. and you are ready to start your new project. But before that, like it or not, you have to configure the workflow. Here is where the struggle begins. What are the key differences between those latest versions of 3-5 most popular game engines? Is it possible to test all of those before the breakfast? And no, it isn’t. Let’s write down the usual game engine testing cycle:

* See some examples in action
* Watch several tutorials
* Convince yourself to download and install those gigabytes of data
* After finally understanding that you are on the beginning of your learning curve, watch several (thousands) more tutorials
* Make something more or less resembling your idea and vocalize the verdict

And that leads us to obvious conclusion: you will just choose the first game engine you find with enough likes and ‘essential’ terms in the feature list.

1. **CLIMB The MOUNTAIN**

Your breakfast time is long gone, but now you have your workflow all set and ready to go! You dive into the development process and after several hours/days/months you dare to ask yourself an easy to spot question: “Why Instead of conveying the meaning of my idea I have to watch hours of tutorials about how do I make this square bounce of another square and then waste my life on duplicating this squares all over the level of my game to create something nearly resembling the surface when even the dumbest of all people I asked understood me when I said “meadow full of flowers”? “.

Yes, you can be partially saved by asset database(store), where you can even find sample scenes that does look more or less nice, but let’s count the steps:

1. (Optional) Navigate to the browser and search the internet for a store that may have needed assets.
2. Search the store and find what you need (skins, objects, animations, etc.)
3. Download chosen items
4. Import to the project
5. Integrate it into your game (position, connect events, etc.)

And that algorithm should be basically applied for each single word you said when explaining your game to the human beings. The idea of explaining your game verbally individually to every single being on this planet instead of making the actual game becomes more and more alluring…

1. **Don’t lose YOUR way**

The first week went by, then the second one. Finally, you made it to the alpha version of your app. It features one 3\*3km map and 5 levels that cover not more than 10 percent of your genius plot. To celebrate this, you finally go outside and order a coffee in the nearby restaurant. While waiting for your rewarding beverage you start realizing that the beautiful picture you had at the beginning of your work is already faded and some of the details became almost impossible to remember. On top of that you receive one more message from your previous employer that tomorrow you have one last chance to come back or you will lose your work there for good. And with the creamy drink you are also being served a question: “Does It worth all that time?”. What if you spend even more of it on this idea and in the end it will turn out to be not something you expected it to be?

As it probably was expected, there are two roads starting from this point:

1. Give up. Just walk away and then remind it only when your resume is lacking several lines.
2. Continue. (and risk your time, money and sanity to put the period at the end of this sentence of your autobiography)
3. **Finish**

And realize it was only the beginning. You have your game made. You played it several hundred times and this time you’re satisfied with the result. You even posted it in Steam/Play Market/Nearby CD shop. But somehow you still found no people willing to shake hands with the glorious creator of the unique and ridiculously time-consuming game. Then you realize why those silly developers start revealing their games in alpha/beta stages when there are still so many bugs and limitations.

You managed to spend your last money on the best advertising you could find and ready to receive your first 5-star comment, and here it goes, **“Looks cool, but the gameplay/animations/design could be better, and can you please remove that feature? I don’t understand why it is there anyway and I think it is slowing the game down”.** What? After all this time spent on this masterpiece somebody dares to ask you to dump the most complex and innovative mechanics you made for your creation? Should you silently accept your time was lost without a reason or say to the commentator everything you think one needs to know and probably get your first 1-star review as well? If only the game creational process was as easy as its first stage…

analyzing the process

As it is clearly seen, the most valuable resource spent in game development is time. While all of us have a lot of experience communicating with people, very few can efficiently deliver their ideas to the highly formalized computer language. Also developers must be ready to spend most of their time on routine processes that have nothing in common with creativity. And even when it comes to creation it is not as fun as interacting with another person. When you explain your thoughts to a human, human can understand them not as you do, but one can also refine them by applying one`s knowledge and explaining one`s vision of your idea. When you have Computer it is far more possible that idea transfer would be one way. Want to know how to make it better? It is your choice, go search the internet!

So here are the key negative points:

1. Game creation takes a lot of time
2. Game creation involves a lot of non-creative work.
3. Game creation software does not usually guide you to the best solution.

Let’s see how this problems are handled in human-human communication.

1. When received a part of the information, human can complement it using one`s world-view. If both humans share almost similar world-view the communication process is incredibly easy.
2. Human mind can position all that 1 million flowers while hearing me talking about the mountain valley without explicitly being told where each of the flowers is and how does it look like. That makes my story deprived of long lists of boring data without making it less stunning.
3. When telling your story to the human, one may spot controversial moments as well as propose to implement some corrections to make your idea even better. Distributed intelligence was never a bad thing, wasn’t it? (If there are managers reading this paper I hope you won’t stop reading after this statement).

And here we came to the point when the main goal can be finally described more or less formally:

Create the game development software capable of mimicking features of human-human interaction. For example, idea-completion, creativity and personal world-view. The goal is to free the developer of uncreative and long work and provide the simple and intuitive way to get the result.

# Basic concept

Here you can find authors vision of the game creational software described above. Vision, that is going to shape the idea. Later in this article you may find the results of an attempt to recreate this idea in real life, so it is important to understand the underlying principles listed below.