

Stand firm

Game Development Diary



last update: MAY.3, 2022

University of utah eae 2100

u1351288@umail.com

Github: https://github.com/DefenderJohn/StandStill-Game-Project

## Background

Before I entered the project, I do have some experience in designing games, but I don’t really have sufficient experience in making those games, thus I need to take this class and learn more about game design.

The background helps me to get the basic ability to design digital games, but more importantly, it helps me to have the braveness to step out of my comfort area and force myself to learn new things in the class.

## Game Idea

The game idea is coming from the traditional tower defense game and the RTS games. I really like those two classes of games, but they also have their disadvantages.

The tower defense game is easy and interesting to play, but it’s relatively hard to make the game more interesting because the tower defense game itself is based on the fact that the tower is not going to move. If the tower is not going to move, there will be tones of good features not applicable to it, which is a loss of players since they should have an even better game to play.

The RTS game is in an extreme of another direction, the gameplay of RTS is absolutely interesting, no need to say, we see so many fantastic RTS games in STEAM and other gaming platforms. However, RTS gave have a huge weakness, which is the difficulty of learning and operating. Players sometimes need a few weeks or even longer time to learn all the tricks and features of the game, after that, they can start to enjoy the fun of the game. However, not all the players have the time, and more critically speaking, the player is relatively hard to get the fun of RTS game the very first time they touched them.

There should be many ways to overcome the weakness of both tower defense and RTS, this is not doubtful. And we also believe that there are some people who had already generated better ideas, but it’s still worth to start considering that question from the very beginning and figuring out what we should do to overcome those weaknesses.

My idea in the very beginning is that, if the limitation of a tower defense game is that the tower can not move, then why not just let the tower move? If the tower is moving, then we can start to add more and more interesting features to it and get rid of some old features that exist only because it’s required by the definition of tower defense but contributes no help to the fun of the game. Also, because the RTS game is too complicated to play, then why not simplify it in some parts, get rid of all the buildings, some of the resources mining, and huge amounts of soldiers?

So my first idea is generated for a very simple reason, if we want to make the tower defense game interesting, then it’s better to let the tower move. If we want to make the RTS game interesting, then it’s better to simplify the soldiers into towers—which are less in amounts, and easier to control. Thus the idea becomes that, we want a game with a tower defense with towers that have been able to move, also we want to somehow limit the tower’s moveability so that it will not have the ability to move unlimitedly.

## During the development of game:

The development of the game is tough in the very beginning because the idea seems clear, but it just “seems clear”, actually is unclear at all. I only decided what the game's basic features will be like, but how about all the gameplay? The story? The challenges? I don’t know that yet. So the first version of my game is a tower defense game with an actual “tower”, the gameplay is that the player controls a group of soldiers who pushes the tower in some direction, while the tower is firing during moving, there is some supply line system that restores the moveability and hit points of the tower.

Looks not bad, however, the first attempt to make the game has already given me a huge failure. I don’t know how to reason why the tower should be pushed, how to write code to let a group of soldiers push the tower (practically difficult), and if I need to let those soldiers be able to absorb damage and died. Those problems are difficult to consider and even harder in practice.

The difficulty drives me to change my mind on the game idea selection, I’m considering first having a good and clear game topic before proceeding any further.

After the idea gets revised, there are more challenges coming to me, and most of them are programming problems. Since this class is not about some specific programming detail, I'll briefly discuss the problem in a most easy-to-understand way without involving any specific code implementation, as a reference to others if they want to practice a similar problem.

图片包含 草, 户外, 男人, 骑

描述已自动生成 The first challenge I meet is the movement of the tank. Since the tank is not a rock, it has multiple moving parts, which can be classified as turret, body, and gun, those three parts are moved and controlled by different functionalities. The body needs to be controlled by moving command, the turret needs to be controlled by aiming, and the gun needs to be controlled by fire. At the same time, the turret must move with the tank body, which means they need to have some relation with each other. That makes me struggle for a while with lots of bugs like this:

It’s funny, but we definitely don’t want to see the same thing in the released game.

The solution for this is to proceed with a relatively hard-to-understand vector calculation that basically “appends” the turret rotation based on the body’s rotation and limits the rotation angle so that it’ll not show the situations like the image above.

The next challenge is the balance between friendly tanks and enemies. I’ve to play tested the game a lot of times, but it seems hard to keep a balance if we want to let the game act as a real game. The solution for the problem is really no quick solution, the only thing to solve this problem is to spend more time and playtest, but time is something I can’t have in this short semester and with myself.

There are lots of other challenges that I can’t put into the list here by the limit of article length, but one thing I learned from there is that we’ll never know which trouble is waiting for us unless we step onside, with more steps you move further, the more trouble you’ll get in the progress. But we can’t avoid those troubles by not taking the challenge, because there is still have chance to achieve success with the trouble, but there will have no chance to succeed at all when one is not even trying to be successful.

## Idea iterations

Based on the reason listed above, the game idea must have to get revised. I first come up with the idea of modern war, but as soon as I realize that the game will be too large if I follow this idea, I again revised the idea. This time I want to have the idea simplified enough that either practicable and easy to understand, thus I selected the modern tank war as my idea, the reason is that the actual content I’m focused on is not about simulating the real war field, but to construct an interesting and worth-to-play game feature. Thus we don’t want to include all kinds of combat forces inside the game, tank only should be well enough. Also, tank needs to burn fuel to move, which solves the problem that there is no good reason for me to let the tower move and consume supplies.

It also solved the programming difficulty of letting people push the tower because letting a tank move is also not easy, but it’s comparably way easier than controlling a group of people. After persuading myself that this idea is feasible and worth to make attempt, I raise the question to the professor and ask if this idea is good enough to proceed. Professor provides a positive answer and also some following questions about the detailed practice of the idea. With the confirmation and the idea, I’m now sure about having this as my idea for the final assignment.

The next iteration happens during the playtest. At that moment I still have not yet finished the functionality of the basic gameplay, however, the play tester still provides me with a very helpful suggestion. The playtester tells me that I should have a HUD functionality for a global view of the data. Although I’ve never practiced HUD functionalities before, this suggestion makes me to deciding to add more HUD and related features to my game. The HUD functionality is not that beautiful and, to be honest, poorly constructed, as you can see in the game, but this provides me more confidence next time when I am asked to implement a more complex and beautiful HUD view.

I believe those iterations are necessary to be taken because, just as what the professor says in class, the game is designed for others to play, if we don’t get the idea from others, then whatever how much effort we put inside, it’s equivalent to ZERO.

## The change of myself

As my dream is to become a professional game developer, I believe I learned a lot from this class.

From what I learned during the development, one thing I appreciate a lot is that the professor provides us a chance and support to develop a digital game. This provided me a great chance to learn and practice all skills I learned in game design and clearly noticed where I still need to improve and where I’m good. Those self-acknowledge experiences can only get from practicing hard problems and challenges. Here I want to quote from the former president of the US, Kennedy, in his speech we choose to go to the moon. The US chooses to go to the moon, not because it’s easy, but because it’s hard. For me, I choose to go to the United States with such a faraway distance, leaving my family, learn the game design and other subjects, also not because it’s easy, but because it’s hard because only hard challenges make people learn things. Also, I appreciate that this chance makes me finished thousands of lines coding, just like watching a baby growing day by day with my hard work, and finally becomes an interesting game. This satisfying is not able to achieve anywhere else.

In this class, despite the tones of challenge I experienced, there is not even one moment for me to change my idea and say to myself: “Okay I think a board game should be good enough, don’t suffer yourself on programming.”[[1]](#footnote-1)

If I do so, I’ll lose my chance to make attempts and get to know my strength and weakness, also not have the chance to force myself to work until midnight and take a deep review of myself, get acknowledged that where I’m able to improve myself in order to be better.

The second thing I learned is from the playtest and the professor’s advice. That advice is very valuable; I’m very appreciative to have them so that my game designing is always on the right track. However, I should also note that that chance is not always available anytime in the future, after graduating years later, I’ll soon enter the zone where no one is guaranteed to provide me professional advice exactly based on what I’m doing and learning. Thus, I need to learn as much as I possibly can before graduating and try my best to not drop into the zone where helpless but no one can help.

In another way, If I learned and mastering knowledges in the future, I want to promise that I’ll be the one who is always willing to provide help to others, not because I’m good, but because I know that I’ve experienced one period of time, a period of time that everyone is willing to give me a helping hand, saying that “You are good, I'll help you to become even better”. I believe everyone deserves that and I want to be the one who takes the baton of helping and taking action on sharing things.

I believe this should be the best thing learned from this warm class.

Thanks for everyone here in my 2100 class.

1. I’m only claim on the specific game on myself, I definitely believe that others are all hard-working students and there exists a lot of board game that is very very hard to design and very fantastic that itself is an art treasure. [↑](#footnote-ref-1)