

# 15-112 Term Project Proposal and Handbook

Shaojie Bai  
Andrew ID: shaojieb

May 1, 2014

## Abstract

This is a brief description/handbook for 15-112 Term Project, the game **Tentacle War**. I will discuss the modules I plan to use and my basic ideas below.

## 1 Introduction

Basically, using the knowledge about Python from course 15-112: Fundamentals of Programming and Computer Science, I make a partially-artificial-intelligence game named the **Tentacle War**. The primary module I use is *pygame*, but to construct a better user interface I also import certain outside images. While the user is playing the game, there is also background sound/music.

## 2 Rule

This is a cell-attacking-cell game. In the main menu, user can use keyboard to choose one of the four options: "Play", "Help", "Credit" or "Achievement". The "Achievement" is to record the activity of the player such that certain statistics can be saved. For example, how many levels have the user completed and how many enemy cells assimilated? And if "Play" is clicked, the user will be able to choose background, the level of the game: there are a few levels, and the user can go to the next level if and only if the current highest level is cleared. For instance, one is able to play level 5 only after level 4 is cleared. The higher the level, the more difficult the game. Moreover, Level 1 is the tutorial (with no AI).//

In this game, the player will be a green cell. For each cell, there will be one or two tentacles that can reach out to other cells. And for each cell, it has a life value that increases with time gradually. The goal of the user is to occupy and assimilate the other "enemy" cells, by clicking and dragging the mouse. The length of the tentacle reached out is proportional to the life value it carries. So as the tentacle getting increasingly longer, the life value of the cell drops. And if the distance between two cells is farther away than the life value can supply, you will have to wait for the life value to increase.

## 2.1 Types of Cells

- EMB-Embrace cell. This kind of cell appear only after level 3 is cleared (also in level1, for tutorial). It can move freely (so it's mobile) and once it gets in touch with any other cell (neutral, friendly or enemy, see below, its value will merge with the target value— it may cancel or add up based on the target's identity). This kind of cell is represented by a circle with color and three corns (triangles) behind it. **ITS MOTION REQUIRES ENERGY, WHICH IS ITS LIFE VALUE.** EMB has *no* tentacle. Its movement is based on strict physics formula in non-relativistic frame:

$$x = v_0t + \frac{1}{2}at^2 \quad (1)$$

$$v^2 - v_0^2 = 2ax \quad (2)$$

- ATT-Attacker cell. This kind of cell is immobile, but it can reach out tentacles with curly/wavy pattern to other cells. "Signals" that transfer the life value will be transported by the tentacles to the target cell. For instance, if the user is a green cell whose tentacle reaches a gray (neutral)/red smaller-in-life-value cell, the green life value it transports will attempt to assimilate the target. Once the assimilation is complete, the target becomes GREEN! But the chain remains so that the just-now assimilated cell will keep growing. The chain information is recorded in a dictionary called `self.dic`.

## 2.2 Cut

Once a tentacle connection is established, it can be cut by user dragging the mouse click "across" it— just like how you cut a line in life. As is introduced before, reaching out tentacles takes life value. So if the cell's life value reaches zero before its tentacle reaches the target, the tentacle gets back. Now say that the tentacle is worth of life value of  $v = 13$ . Note that the life value it worths has nothing to do with the life value **of the cell** that it transports. Then if we cut the tentacle, the part closer to the cell immediately collapses back to the cell; and the part closer to the target will quick "collapse to the target." Based on the color of the target, this  $v = 13$  may cancel or increase the life value of the target.

## 3 Outside source use

Currently, my background image, chain, signal (yellow) and the circle cell are made by my own. However, I also crop certain images, like the "sun," from the Internet. These sources are cited.

## 4 Artificial Intelligence

Besides the player end cell, there are enemy cell, which are controlled by the Artificial Intelligence (AI) algorithms. The algorithm will calculate the most

optimal choice to make, and then execute it. The principle of AI involved is the "Finite State Machine," which is a CS idea that using different states to judge the move of a computer. Based on the states of a cell, and the priority queue it updates every round, a cell will pop out the target it looks at. Python will judge based on the priority number (just like what we did in A\*!) to determine which cell to attack (or to assist, it friendly). It sends message to its allies if its weak. The algorithm is based on both the cell's own life value and its surrounding environment. The AI is designed in the way such that even if there are three "camps" and colors of cells— like in Level 7, the game AI still works perfectly fine.

The Artificial Intelligence for Level 1-3 is made easier intentionally to let the users adapt to the game. Then, for Level 4 and above, the AI will be much more complicated, and Computer will become very clever.