

# CSCI2120 INTRODUCTION TO SOFTWARE ENGINEERING

## PROJECT – PLANTS VS. ZOMBIES

Group Registration Deadline: 30-Mar-2016 (Wed), 23:59

Project Phase 1 Deadline: 24-Apr-2016 (Sun), 23:59

Project Phase 2 Deadline: 15-May-2016 (Sun), 23:59

Project Demo: 16-May-2016 (Mon)

### 1. GROUP FORMATION

This is a group-based project. Each group consists of two or three members. Please register your group by the group registration deadline shown in the above at Blackboard (<http://elearn.cuhk.edu.hk>). For ungrouped students, we will randomly assign groups to them after the group registration deadline. Please check your CUHK email (CWEM) for the latest grouping information.

### 2. PROJECT TASK

In the project, you have to design and implement a game, which is similar to a famous game called “Plant vs. Zombie”. The goal of this game is to beat your opponent by using the unique ability of Plants and Zombies.

---

#### 2.1. INTRODUCTION

In this game, there will be a human player and a computer player. They play the role of Homeowner/Zombie respectively. Your program should do the following when it starts.

1. At the start of the program, a game menu should be displayed and the human player is asked to enter his/her name. The name may contain spaces and should contain at least one alphabet.
2. The human player is asked to which role (Homeowner or Zombies) he would like to play. The computer player will play the remaining role.
3. After setting up, a battle field should be prepared with 6 (height) x 10 (width) grids, and the game begins.

---

#### 2.2. PLANTS AND ZOMBIES

The homeowner can grow plants while the zombie army can produce zombie warriors in each round. Different plants and zombies have different attributes, e.g. damage, HP, attack range, and cost. Plants have different attack range, while zombies can only attack plants one grid in front of them (i.e. the attack range of the zombies is one grid). Design the values of damage, HP, *etc.* according to the tables to make the game reasonable.

Plant	Damage	HP	Attack Range	Cost (Number of “sun”)	Remark	Abbreviation
Peashooter	Normal	Low	Long	Low		PS
Bonk Choy	Heavy	Normal	Short	Normal		BC
Threepeater	Heavy	Low	Long	High		TP
Wall-nut	None	High	None	Normal		WN
Sunflower	None	Low	None	Low	Generate additional “sun” in every round.	SF

Zombie	Damage	HP	Attack Range	Cost (Number of coins)	Speed	Abbreviation
Basic Zombie	Normal	Normal	Short	Low	Normal	BZ
Buckethead	Normal	High	Short	High	Normal	BH
Football	Weak	High	Short	High	Fast	FB
Conehead	Heavy	Normal	Short	Normal	Slow	CH

---

## 2.3 BASIC RULES

### INITIAL STATE

The homeowner starts with an initial hit point (HP) and some number of “sun”, while the zombie army starts with some number of coins.

Behavior of plants and zombies:

- The zombie army can produce at most one zombie warrior at each of the rightmost grids of the battle field. (Therefore, at most 6 zombie warriors can be produced in a round).
- The zombies move from the right to the left.
- Each grid can only be occupied by one plant or zombie.

- ## EVENTS IN EACH ROUND

- ## WINNING CONDITION

You can design your own text user interface (TUI). Graphic user interface (GUI) is optional. You may use abbreviations to represent plants and zombies in the TUI. Please make sure that your program displays important information, so that we can check whether your program is working correctly during the demonstration.

[illegible]

The AI of computer may not be tactical but should be reasonable, which means it should produce some plants/zombies during the whole game.

### 3. DELIVERABLES

In this project, please submit the following artifacts in the two phases:

#### 3.1 PHASE 1

##### 3.1.1 REQUIREMENT WORKFLOW ARTIFACTS (10%)

- A use case diagram for the program
- A brief description for each use case
- A step-by-step description for each use case

##### 3.1.2. ANALYSIS WORKFLOW ARTIFACTS (30%)

- A state chart for the program
- For each use case
  - A number of scenarios
  - A class diagram (in stereotype) for each use case
  - A collaboration diagram or a sequence diagram for each scenario in each use case. (Note that the number of collaboration diagrams and the number of sequence diagrams should be roughly the same.)
  - A flow of events for each collaboration / sequence diagram.

##### 3.1.3. DESIGN WORKFLOW ARTIFACTS (10%)

- A final class diagram with attributes, attribute formats, and methods (including accessors and mutators if necessary) for the program.

#### 3.2 PHASE 2

##### 3.2.1 IMPLEMENTATION WORKFLOW ARTIFACTS (50%)

- Source code in C++
  - You should enforce some good programming practices to improve the program readability, such as indentation, meaningful variable names, and comments.
  - You should enforce information hiding, such as declaring private/protected member variables.
  - You should write classes in separate files.
- User manual
  - A user manual to show how to play the game and list information (e.g. the range of attributes for different classes, the actual damage caused by different skills).

Although it is not required to submit anything about the test workflow, you have to perform adequate testing on the program before submission.

#### 4. ASSESSMENT

The project will be graded according to the correctness and object-oriented software engineering principles, such as information hiding, encapsulation, inheritance, and polymorphism.

Please make sure that your C++ program can be compiled and executed successfully. Otherwise, you will get a very low mark. Please also be reminded that your TA will read your Phase 1 design and Phase 2 code and can identify similar submissions easily. Consequences of confirmed plagiarism cases can be failing of the course and receiving a demerit.

#### 5. SUBMISSION

For Phase 1, please submit the various artifacts in soft copies to Blackboard. Any member of the group can submit, but please include all students' names and students' IDs of the whole group. Resubmissions are allowed. But only the latest one will be counted. Late submissions within three days can only receive 70% of the marks. Late submissions more than three days will not be graded.

For Phase 2, please submit the user manual in soft copy to Blackboard. Any member of the group can submit, but please include all students' names and students' IDs of the whole group. Resubmissions are allowed. But only the latest one will be counted. Late submissions within three days can only receive 70% of the marks. Late submissions more than three days will not be graded.

More information about the submission of source code will be announced later. Since a demonstration session will be arranged for you in the next day after the Phase 2 deadline to show the correctness of the implementation, no late submissions are allowed for the source code.