

SYSC 3110 Project – Plants vs Zombies: the Puzzle!

The goal of this team project is to implement a puzzle version of the popular Plants vs Zombies (PvZ) game. PvZ is a game in the tower defense genre available for most gaming platforms. If you don't know the game or don't own the suitable platform, find out more here: <http://www.popcap.com/plants-vs-zombies-1>

The basic idea of the game is that you drop plants with various magical powers in your garden to defend against invading zombies. One plant, the sunflower, gives you “sun” points that you can use to buy more plants. Other plants shoot at the zombies at various time intervals or damage points. Other plants prevent or slow down the advance of the zombies. Plants have a different cost in “sun” points depending on their type. Also, once you drop a plant you must wait some time before plants of its type are available again for planting. This wait time also varies depending on the type of plant. There are also different kinds of zombies: some more resistant to damage, some faster, some that aren't slowed down by certain obstacles, some that have weapons of their own, etc. When a zombie gets near a plant, it starts chewing on it until the plant is destroyed...

The game has multiple levels. At the beginning of each level, the kinds of monsters that will be attacking are presented to the player, to help him/her select the kinds of plants he/she wishes to use for the level. During the level, zombies come in attacking waves. The level is won if the plants manage to destroy all the attacking zombies. The level is lost if one or many zombies manage to get through the plants and reach the end of the garden, thereby entering the house and eating brains!

There are of course many more subtleties to the game, but you should at least cover the features listed above. But one main aspect of the version of the game you are asked to implement is that it is in “puzzle” form, i.e. it is not real time. The player is prompted to enter a move, and nothing moves or is changed while the player makes a decision. This will allow you to work on a text-based version of the game in the early part of the project. Also, only in such a “puzzle” version would it make sense to ask for unlimited “undo” and “redo” actions as well the possibility to save and load a game at any point...

Finally, you are asked to work on a game level editing tool. For each level, the user should be able to specify the type and number of zombies, as well as the timing of the waves of arrival of the zombies.

I don't expect fancy visuals. A global view of the game as a grid of buttons that the player can press to select or drop a plant would do the job, but of course feel free to paint the world on a canvas.

The project is divided into 4 iterations, each ending with a milestone corresponding to deliverables that will be graded. You will be able to use the TA's and instructor's feedback from iteration i for iteration $i+1$.

Milestone 1: A text-based (i.e., console-based input and textual representation), bare-bones but playable version of the game (sunflowers, one other type of plant, one type of zombie, just one level), UML modeling of the problem domain (class diagrams, sequence diagrams, complete variable and method signatures), detailed description of the choice of data structures and relevant operations: you are providing an initial design and implementation for the Model part of the MVC. Do not worry about any GUI yet.

- Deliverables: readme file (see explanation below) + code + UML diagrams + documentation, all in one zip file.
- Deadline: Wednesday Oct 23rd. Weight: 15% of the overall project grade.

Milestone 2: GUI-based version (now you're adding the View and the Controller!) of the bare-bones version of the game + Unit tests for the Model. The code is allowed to "smell" at this point.

- Deliverables: readme file + design + corresponding tests + code + documentation, all in one zip file. In particular, document the changes you made to your UML and data structures from Milestone 1 and explain why. Proper division of the project into several packages at this point is recommended.
- Deadline: Monday November 4th. Weight: 20% of the overall project grade.

Milestone 3: Complete Implementation: fully-featured with various types of plants and zombies, and unlimited undo/redo feature.

- Deliverables: readme file + code + refined tests + refined design + documentation. The program must work robustly, and the code must be "smell-free" (we will be hunting for smells!). Make sure that you document the changes since the last iteration, and the reason for those changes.
- Deadline: Monday November 18th. Weight: 30% of the overall project grade.

Milestone 4: Two more things: 1- Save/load features. You may use Java Serialization to achieve this. 2- Game level builder. The levels may be saved in XML. 5% overall project bonus: try a real time version of the game and/or porting it to the Android platform.

- Deliverables: readme file + code + tests + documentation. Your project should be well packaged, and the program(s) should be easy to install and run.
- Deadline: Monday December 2nd. Weight: 35% of the overall project grade.

Milestones must contain all necessary files and documentation, even those items that are unchanged from previous milestones. Missing files can not be submitted after the deadline, no exceptions. Verify your submission contains all necessary files (in particular, don't forget to include your source code!) before submitting on cuLearn.

The "readme" file, listed as a deliverable for each iteration, is typically a short text file that lists and describes: the rest of the deliverables, the authors, the changes that were made since the previous deliverable, the known issues (known issues are graded less severely than the undocumented ones!) and the roadmap ahead.

"Documentation" includes up-to-date UML diagrams, detailed descriptions of design decisions made, complete user manuals, and javadoc documentation.

Note that nobody is stopping you from working ahead of schedule! In fact, iteration $i+1$ will very often give you good insight into iteration i .

This is a **team** project. Each team should be composed of 4 members (or, exceptionally, 3). A project's success depends on contributions from each member! So divide the work and cooperate. **Each contribution (source code, documentation, etc.) must contain the name of its author:** we will use this to determine whether there is any significant difference in the quality and quantity of the contributions of the team members. If any such difference is detected, the individual grades will be adjusted accordingly.

You are expected to use Github or Gitorious to manage your project (version control, issue-tracking, wiki, etc...), but the deliverables for each milestone should also be submitted for marking on WebCT. The TA and/or the instructor will also be members of each of the project groups on Github, so that they can track your use of the tool, verify that all group members are contributing, and maybe even post bug reports and feature requests!

Have fun!