

**Concordia University
Department of Computer Science
and Software Engineering**

**Advanced Program Design with C++
COMP 345 --- Fall 2014**

Project Final Build Grading

1. First Incremental Code Build Description

You must deliver an operational version demonstrating the full capacity of your system. This is about demonstrating that the code build is effectively aimed at solving specific project problems and completely implementing specific system features. The code build must not be just separated portions of the final project, but a fully operationally integrated software that can be demonstrated by its operational usage.

The presentation should be organized as follows:

1. Brief presentation of the Problem Analysis, Design, and Use of Engineering Tools as listed below under “Graduate attributes—skills”
2. Demonstration of the functional requirements as listed below under “Functional Requirements”.

You are graded according to how effectively you can demonstrate that the features are implemented. If you cannot really demonstrate the integrated features through execution, you will have to prove that the features are implemented by explaining how your code implements the features and what are the expected integration problems, in which case you may lose some marks, even if your explanations are satisfactory.

During your presentation, you have to demonstrate that you are well prepared for the presentation, and that you can easily provide clear explanations as questions are asked about your understanding of the problem being solved, the structure and functioning of your code, as well as your use of tools.

2. Team Identification

Team	Evaluator	Signature	Date	Time

3. Grading

Presentation	6
Effectiveness, structure and demonstrated preparation of the presentation	3
Knowledge of code base/clarity of explanations	3
Functional Requirements	30
Map creation and editing	8
User-driven interactive creation of a map as a grid of user-defined dimension	1
User-driven allocation of grid elements such as scenery, path, entry point and exit point.	1
Saving a map to a file exactly as edited	2
Loading a map from an existing file, then editing the map	2
Verification of map correctness before saving (at least 3 types of incorrect maps)	2
Game play	22
Game starts by user selection of a previously user-saved map, then loads the map.	1
User-driven placing of towers on the map, following the game's restrictions.	1
Implementation of currency, cost to buy/sell a tower, and reward for killing critters.	1
Implementation of <u>all</u> towers' level-dependent characteristics: cost to increase level, refund rate, range, power, rate of fire, special effects.	2
Allows the user to start a wave, upon which a wave of critters starts moving from the starting point to the ending point along path cells.	2
Implementation of three different kinds of towers that are characterized by having different special damage effects, e.g. splash, burning, freezing.	3
The towers can shoot at the critters, inflicting damage, and eventually killing them.	2
The towers can target the critters using different strategies, e.g. nearest to the tower, nearest to the end point, weakest, strongest, etc.	3
Wave-based play, i.e. when all critters in a wave have been killed, the player can place new towers, upgrade towers, sell towers, and start a new wave.	2
End of game, e.g. when a certain number of critters reach the exit point of the map, or the critters steal all the player's coins, or the player succeeds in killing a certain number of waves.	1
Tower inspection window that <u>dynamically</u> shows its current characteristics, allows to sell the tower, increase the level of the tower, and select the tower's targeting strategy.	2
Critter observer that allows to <u>dynamically</u> observe the current hit points of any critter.	2
Graduate attributes—skills	14
Problem analysis —demonstration of proper analysis of the problem to be solved and justified adoption of a particular solution.	3
Design —description of the rationale and structure of the architectural design and detailed design, demonstration/explanation of the correct use of three different design patterns such as those implemented in the individual assignments.	8
Use of Engineering tools —justified adoption and proper use of particular tools for the implementation (e.g. compiler, IDE, libraries, project management tools, etc).	3
Total	50