Modern Computer Games

COMP 521, Winter 2015

Assignment 1

Due date: Wednesday, Feb 4, 2014, by 6:00pm

Note: Late assignments will only be accepted with prior **written** permission of the instructor. You must **explain** all answers and **show all work** to get full marks! Please make sure your code is in a professional style: **well-commented,** properly structured, and appropriate symbol names. Marks will be very generously deducted if not!

Description

For this assignment you will need to install the Unity game development environment from "`http://unity3d.com/`". This is a commercial product, but for everything you do in this course the free version is more than sufficient. Please use the latest currently downloadable version. As well, for work submitted in this course, please restrict your usage of unity assets to free versions.

Note that the tasks below focus on building game mechanics and structures. Aesthetics are not a factor in grading—solid objects should be opaque rather than wire-frame, but you do not need to find or use complex models or textures, and creative use of simple assets and asset-combinations can be used to accomplish all objectives. You may use either first-person or third-person camera viewpoints.

Also note that the Unity site has links to helpful forums and short tutorials on using Unity for different purposes, and those will be your main resource for dealing with the software.

This assignment requires you implement a relatively simple "mini-game" in Unity.

The player begins in an above ground location containing a single entrance and exit to/from an underground complex. The entrance and exit are one-way, so once inside the player must find their way to the exit.

The complex is formed of a (randomly generated) "dungeon-like" room/tunnel structure constrained within a 30×30 grid space. It includes the *start* room where the player arrives, an *exit* room, from which the player is able to leave the underground complex, and 3 room "sets" (A, B, C), each of which consists of 2 connected rooms (a *primary* and a *secondary* room). All rooms are the same size, 3×3, and support at most 4 doorway/entrance/exits (i.e., one in each wall).

The *start* room connects to 1–3 *primary* rooms. Each *primary* room connects to its *secondary* room, and to 0–2 other *primary* rooms. Access to *secondary* rooms is (initially) blocked.

Each *secondary* room has one (initially blocked) doorway to its *primary* room, and exactly one *secondary* room connects to the *exit* room, again (initially) blocked. Ignoring blocked doorways, the overall structure should form a single-entry, single-exit braided maze (although it does not need to fill the entire 30×30 grid space).

The player gets through the 4 blocked doorways by shooting projectiles at them. However, each blocked doorway may only be destroyed with a specific projectile: the *start* room contains a single projectile that will break down a doorway to one of the *secondary* rooms associated with a *primary* room reachable from the start room, and projectiles for the other doorways are located, one per *secondary* room. Each projectile may only be used once, and until used projectiles are objects that the player is able to pick-up, carry around, and drop. The player may only carry one projectile at a time; use left mouse-button to fire, "," to drop, and "." to pick up. Using a projectile consumes it, whether or not it opens the corresponding doorway.

The precise room/projectile association is created at the same time as the maze is constructed (before/as the player enters the complex), does not change during the game, and constitutes the puzzle component of the mini-game. You must somehow ensure that the game is solvable, requiring the player to enter all *secondary* rooms before leaving the maze. The correct mapping between doorways and projectiles should given somehow to the player through subtle visual queues. It should be possible for the player to fail, and to succeed.

Specific requirements:

1. You must provide a non-trivial initial (above-ground) terrain. This does not need to be a complex terrain, but should be bounded, freely navigable, have some visual interest/variation (eg differences in terrain height), and at least 2–3 obstacles of some form. The location of the underground entrance and exit should both be relatively easy to find, and must be contiguous with the above ground space (so entering/leaving the underground does not involve teleporting). **10**

   Note that neither the entrance nor exit is protected by a door. However, it should not be possible to enter the exit, nor leave the entrance (once entered).

2. The underground complex should be constructed as a random braided maze, as described above. It should be randomly generated so it is (very likely) different each time the game is started. It should be fully navigable (modulo blocked doorways), and include point-sources of light. (Note: the complex does not need to be strictly planar in an overall sense; you can have terrain variation.) **20**

3. Doorways and projectiles should be defined with interface, behaviours, and interactions as described above. This includes creating a projectile placement and room/projectile association that guarantees the puzzle is solvable. *Include a separate text document clearly describing your strategy.* **15**

4. The state of each doorway (blocked or not) should be visually clear. Projectile/correct-doorway and projectile/anything-else interaction should be visually apparent, and unambiguously distinct. **3**

5. Exiting the maze constitutes winning. Losing the game is less obvious—if winning is no longer possible the player should be allowed to continue to explore, but it should be made unambiguous to the player that they have lost. **2**

6. Describe the abstract game structure (“narrative”) as a 1-safe Petri-Net, drawing the actual net. Your model should not represent a specific map layout, but it should include structures that allow you to represent all significant game actions, including projectile usage and item take/drop. Be sure to label each node in your Petri-Net representation, indicate the starting marking, and identify which nodes correspond to win/lose conditions. **7**

7. Based on your Petri-net model, is your game p-pointless for a finite p? Give a clear argument either way, providing a specific value for p if appropriate. **3**

What to hand in

Assignments must be submitted on the due date **before 6pm**. Submit your assignment to *MyCourses*. Note that clock accuracy varies, and late assignments will not be accepted without a medical note: **do not wait until the last minute**.

For the Unity questions, hand in an exported project containing all files needed in order to reconstruct and run your simulations. Note that Unity exports can be extremely large, and take non-trivial time to upload (another reason last-minute submission may not work out well).

For non-Unity questions, submit either an ASCII text document or a .pdf file *with all fonts embedded*. Do not submit .doc or .docx files. Images (plots or scans) are acceptable in all common graphic file formats.

This assignment is worth 15% of your final grade. **60**