Modern Computer Games

COMP 521, Fall 2020

Assignment 1

(Optional) Quick sanity check: Wednesday, September 30, 2020, by 6:00pm
Final due date: Friday, Oct 2, 2020, 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!

Note: Submitting your assignment by the optional sanity check deadline will allow TAs to do a quick verification that your assignment at least loads and executes, to avoid problems due to versioning or incomplete project files.

Description

This assignment requires you implement a relatively simple "mini-game" in Unity3D. You will thus need to install and become familiar with the Unity game development environment from "`http://unity3d.com/`". This is a commercial product, but for everything you do in this course the free (personal) version is more than sufficient. **Please follow the instructions given on MyCourses with respect to Unity version and use of Unity assets.**

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 external models or textures; creative use of basic assets and asset-combinations can be used to accomplish all objectives. Assume a first-person perspective is required unless otherwise specified.

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.

Overall design

The main, static play area generally consists of a large rectangular area, styled as an outdoor terrain (add some vegetation and ground texture; small plants are ok, but no trees). The area is fully surrounded by impassible terrain (mountains or walls), to ensure the player does not leave the area. The overall area is divided into two, disjoint areas due to a deep canyon, which the player should not be able to get out of if they fall into.

The player should be spawned at one end of the area. From that point, there should be a visually obvious and traversable wandering path, leading to the middle (approximately) of the canyon edge. This path should be a single, non-intersecting path (unicursal), randomly generated at the start of each playthrough, so it is (probably) different each time. The path should have multiple sharp turns, and a non-trivial length that would require the player at least a full seconds to follow (without making movement sluggish). Note that the path should allow for continuous movement (even if discretely constructed), and the player is not actually constrained to move along this path.

Placed at random, distinct locations along the path, are 8 or more visually apparent objects that represent projectile resources. Each time the player passes over such an object it is removed from the level representation, and the player acquires a(n additional) projectile resource. The number of projectiles the player has should be visually apparent in some fashion.

The canyon is crossed by traversing a perfect maze, randomized and dynamically constructed to be different on each playthrough. This maze should be based on a 5x5 discrete grid, with all areas reachable, spanning the canyon and ensuring entrance and exit on opposite sides of the canyon. The maze itself is represented by separate floating platforms for each grid cell. The player is required to jump from one cell to another, but should not be able to successfully jump between non-adjacent cells (you can either prevent the player from doing so, or allow them to fall, but correctly jumping should not be a difficult task). Which cells are adjacent or not should be clearly visually apparent, but it is up to you how to represent that.

Once the player reaches the other side of the canyon, they need to destroy the maze solution using their projectiles. Player

projectiles should be visible objects moving rapidly (but noticeably) in a straight line out from the player in the direction of the camera. A projectile destroys the first maze cell it encounters. The player's task to complete the level is to ensure the maze can no longer be solved—they must remove at least one cell that is part of the maze solution. Projectiles that encounter other parts of the level, or which exceed the level boundaries should disappear. on contact. Only one projectile may exist at a time, and each projectile fired consumes one projectile resource.

Once the player gets through the maze and eliminates the maze solution, the game should indicate a winning state. If a player uses all projectiles without eliminating the maze solution, is not able to get to the other side of the canyon, or falls into the canyon, a losing state should be indicated. In either case no further action should be possible by the player.

Specific requirements

1. You must provide a non-trivial initial, static terrain as described above. It should be bounded and styled as described above. Use a standard WASD/mouse controller for motion and looking around, with a a first-person camera view (you do not need to code the camera and basic motion control, you can use the normal Unity assets). Use the space-bar for jumping, and a left mouse click for firing a projectile. Ensure sufficient ambient or other light is present so objects are visible. **10**
2. A dynamically constructed unicursal path must lead to the canyon/maze, with at least 8 projectile resources that the player can "take" by moving over them. **10**
3. The maze structure is uniquely generated on each playthrough, with structure and behaviour as described above. Specifically, **20**
   (a) A solvable, perfect 5×5 maze with entrance and exit on either side of the canyon, composed of floating platforms.
   (b) Platform adjacency is clear, and enforced in terms of a player's ability to successfully move from cell to cell.
   (c) Dynamically and randomly generated to be (highly likely) different in each play.
4. The player should be able to solve the canyon maze by jumping from cell to cell. **5**
5. The player can fire one projectile at a time, which moves visibly and directly away from the camera in a horizontal plane. Projectiles interact with maze cells properly, and otherwise disappear on contact or if going too far. The number of remaining projectile resources must be apparent. **10**
6. Winning and losing conditions should be properly detected and indicated. **5**

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 excessive use of assets can make Unity exports extremely large. Allow ample time to upload your assignment solution, and please follow the general Unity instructions given in MyCourses.

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