# D3 Demonstration

15 minutes (no over-runs) for demo

You will be given a time for your lab demo – make sure you are there and set up ahead of time so as not to delay marking. Remember, we can only keep to time if you do.

Here is the breakdown of the marks:

1. 20% 2-3mins Source Code Description

(high‐level overview of each significant part while showing us the code itself)

Have someone available who can give us an overview of the code - and point to bits of it that are important, interesting or were challenging – try and do this in about 2-3 minutes to give you sufficient time

Set up access to the source code on a different machine to the one you will be playing the game on.

**Ben to talk through selected puzzle code. Val to talk through PlatformGame code, and a quick whizz through what all the other modules are. Alex to answer any questions on animation.**

2. 50% 8-10mins Demonstrate game by playing it

Most of the available marks are for your demonstration, so make sure the bits you show us work, and talk us through the demo as you are doing it.

**Suggest Ben to demo the game, as he’s team leader. But Alex can do it if he wants. We’ve got 8-10 mins so we don’t need to rush (in fact, if we rush through it, it will look like it’s too easy, which I’m worried that Jaejoon will say anyway).**

3. 20% 4-5 mins Markers play the game (it should work, and it should be fun to play)

As we try to play your game you need to remember two things – we have never played it before and we’re not gamers. Be prepared to step in and point out what we should be doing, where we’re going wrong etc., so that our 4-5 minutes of play are as stress free as possible. We won’t have time to play the game from the beginning to the end; if there is a particular part of the game you think you have designed well and we’ll enjoy, set that part up for our 5 minutes.

4. 10% short written instructions on how to play the game

Make sure your written instructions tell us something about the game, its goals and objectives, how the play progresses, what is/should be fun about it, the challenges, the different characters and situations we might encounter etc. Provide some pictures, as well as simple instructions on which keys to use.

**Do we need to add pictures to the instructions?**

Make sure you practice all this before the demo....

### Source Code

|  |  |
| --- | --- |
| AnimatedCollectible | Runs a thread to animate the collectible, cycling through the images |
| AnimatedHazard | Runs a thread to animate the hazard, cycling through the images |
| AnimatedPlayer | Runs a thread to animate the player, cycling through the images |
| Collectible | Inherits from Platform. Contains methods to get/set whether item has been collected.  This is used by PlatformGame (to pick up items), and by Game (to display collection) |
| Game | The main entry point for the game. Handles the main menu, the display of the evolving Christmas Room, and the items collected so far. |
| Hazard | Inherits from Platform. Contains method to set hazard animation |
| Level | Tracks the game level |
| MazeTile | Very basic handling of a puzzle tile |
| Platform | Handles a platform object. Contains methods for movement, getting and setting position, and to set the animation state (for collectibles) |
| PlatformGame | Runs the platform level, displaying the player, platforms, obstacles, hazards, and collectibles. Controls movement from keypress, inertia, gravity |
| Player | Handles the player object. Contains methods for movement, to check touching (left, right, above, below), to control lives, and to set state of player animation |
| Puzzle0 | Runs the maze puzzle |
| Puzzle1 | Runs the Christmas Tree Lights puzzle |
| Puzzle2 | Runs the food picture puzzle |
| Puzzle3 | Runs the sleigh puzzle |
| PuzzleTile | Handles the tiles used in each puzzle |
| Score | Manages the player’s score in the current level |
| Utils | Defines constants and common methods. For example, the position and images for every platform, obstacle, hazard, collectible |