Game Title
Design Document

Bjørn Fjukstad
BIO-8010 Communicating Science Module 3
Visualizing your science
github.com/fjukstad/bio-8010

1 Introduction

Code Lab is a game where kids collaborate on escaping from an underground dungeon by programming their in-game characters to fight monsters, solve puzzles and collect gems. The game is played in a collaborative environment such as the Tromsø Display Wall[1], where kids program on their own devices and run the game on a large shared display. The display wall environment provides an interactive arena where kids can collaborate on completing the game together.

Since the kids need to program the characters to perform different tasks, they will have to learn the basics of programming. The different levels will require them to learn about *variables*, *data structures*, *functions* and *control statements* such as *for*-loops and *if*-statements. As the kids play the game, the puzzles and problems they are faced with will increase in difficulty, making it necessary to design and implement more complex solutions.

The game is intended for children 10 - 16 years old, who already have some experience with graphical programming environments such as Scratch[2]. It is intended for kids that want to learn more about programming, specifically getting started with text-based programming.

CodeLab is open-sourced at github.com/fjukstad/bio-8010 and there is a playable prototype of the first-level at fjukstad.github.io/bio-8010.

2 Background

For teaching programming to children there exists a plethora of different systems and tools. For the youngest kids, a popular alternative is Scratch[1], where kids use a visual programming language where they can make games and other small projects. For older kids a popular choice is to go into game modding, specifically modding the popular video game MineCraft[2]. CodeCombat[3] is another alternative where kids program game characters through labyrints or different set of tasks. If kids want a more hands-on approach it is popular to program either Lego Mindstorms[4] or small Arduino computers[5].

CodeLab is a game that tries to make learning text-based programming more fun and collaborative through a video game that kids play in an shared environment. It takes the gameplay from CodeCombat and the physical hands-on interaction from Lego Mindstorms and Arduinos, making an interactive and collaborative learning environment for programming.

[1] scratch.mit.edu
[2] minecraft.net
[3] codecombat.com
[4] mindstorms.lego.com
[5] arduino.cc

3 Description

CodeLab takes place in a fictional dungeon, where each player is assigned a hero that he or she controls by programming their actions. The players equip their heroes with armor, weapons and other items that can help them complete the different levels. For each level, the players have to complete a set of tasks by programming their characters by using a programming language similar to the Lua programming language[6]. Players write the code on their local machine, be it a laptop or a smart phone, and see their characters perform the actions on a large shared display. Alongside the game view, the players see each others code making it possible to help out eachother if they encounter any problems.

The game is turn-based in the sense that every turn is a line of code or the execution of a program. The players can either run their entire program to pass the level, or they can interactively type commands that completes the level. Typically the first levels where the players would only move a character is suitable for an interactive solution, while later levels require more complex programs that are difficuty to complete by writing only single lines at a time.

3.1 Game Narrative

The game starts with the players being introduced to their heroes and the dungeon they find themselves in. The players get to know what they can equip their heroes with and the actions they can make their heroes do. Each set of item has special actions that the players can use, e.g. you need to equip a hero with boots if you want to move. Some boots give you the power to walk faster, while some give you the ability to climb obstacles. The players are also introduced to the programming that they need to do to complete every level. The first level introduces them to the four commands: $moveDown$, $moveUp$, $moveRight$ and $moveLeft$ that moves their character either down, up, right or left. Figure 1 shows a sketch of the first level. The player is the blue circle, and the goal of the level is to navigate down to the yellow square. In the full game you would image that the circle is replaced with a more graphical game character, and that the yellow square could be a person that needs help or assistance.

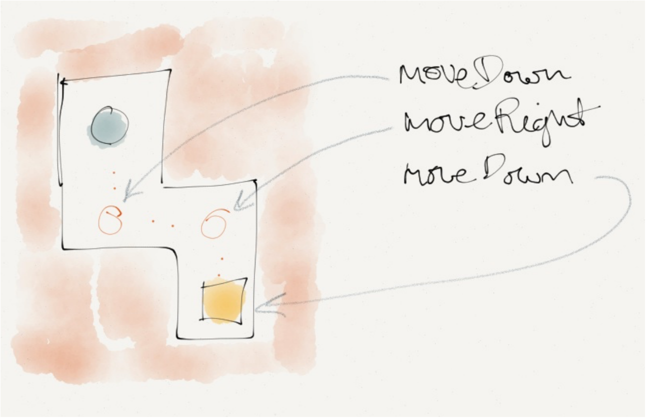

Fig. 1. A sketch of the first level of CodeLab. The goal of the level is to write code that moves the character (the circle) down to the yellow square. The player writes three commands, $moveDown$, $moveRight$ and $moveDown$ to complete the level.

In the first stages of the game the players would also need to familiarize themselves in the game setting. CodeLab is designed to run in a collaborative environment such as the Tromsø Display Wall, where each player writes code on their own device while the graphical output of game is shown on a large shared display. This encourages collaboration between the players, and creates a more interactive learning environment for the children playing the game. From my experience with the local Code Club, having something run in a shared setting makes the whole coding experience more fun and collaborative than working on your own computer.

As the game progresses both the difficulty of the levels and the complexity of the code the players produce increase. The first levels will concentrate on the basics of programming, with different commands that the players can execute. Following these the players are introduced to *conditionals* such as *if*-statements or *for*-loops, *data structures* and *variables*, and *functions*. As the game progresses the players will

[6] lua.org

have access to more commands and features, such as the ability to fight monsters, build structures and forge weapons.

The ultimate goal of the game is to escape the dungeon that the player is trapped in, and have learned the basic skills needed to get started with text-based programming.

3.2 Game Environment

Figure 2 shows an illustration of the shared environment CodeLab I envision children playing the game in. Kids collaborate on writing the code that makes it possible to complete a level. They can write the code on their own devices, but must run the game on the shared display where other kids can view their progress and code.

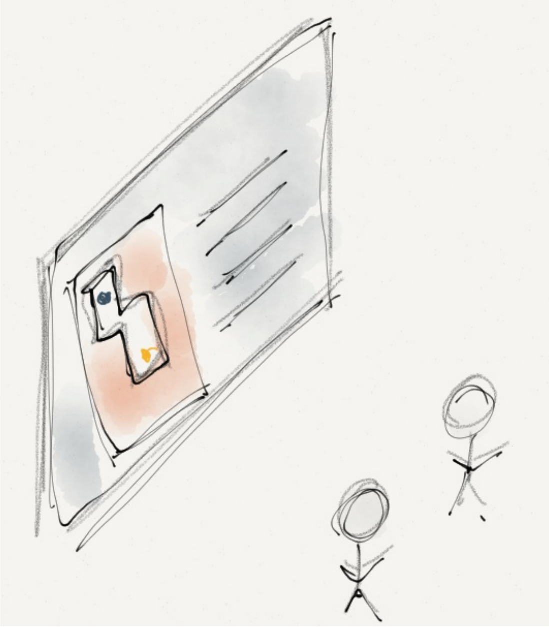

Fig. 2. The CodeLab environment. Players collaborate to solve a level. Both the graphical window where the game runs, as well as the source code is shown on a large display.

In addition to just having one game session on the shared display, some levels of CodeLab requires that kids collaborate to continue to the next level. Figure 3 illustrates how players are given similar levels that they have to complete together to proceed. Collaborating on the different levels will help the kids learn more about the collaborative side of programming by sharing code and helping others.

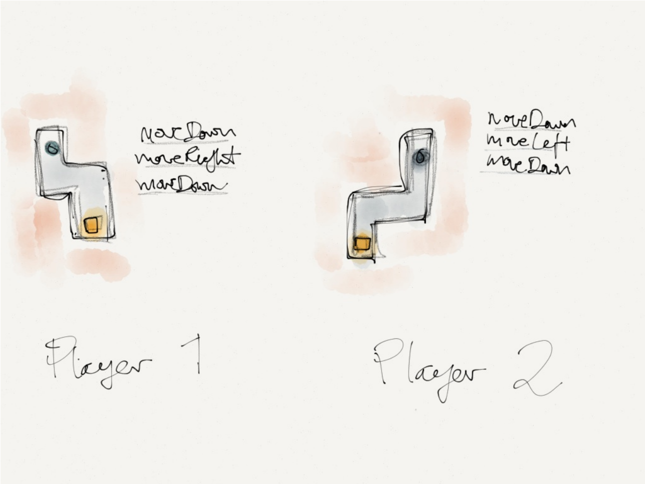

Fig. 3. Two players trying to complete the first level. This view is shown on the large display, where they can help each other write the necessary code.

3.3 Game Tasks

The overall goal of CodeLab is to escape the dungeon that our heres has fallen into, and by doing so learning text-based programming. The game itself is broken into smaller levels that the players need to complete by programming the actions of their characters. Every level will have one or more specific goals that the game characters have to complete, e.g. defeat an enemy or get to a specific location. Some levels will also require that the players use specific programming skills, such as placing actions within a *for*-loop or a *function*.

4 GAME MECHANICS

CodeLab is in essence two games in one. The first being the actual dungeon where the heroes live, where they must fight enemies and escape. The other is the coding-part where kids use their skills to control their heroes through real programming. These are tightly coupled to each other, new items allow the players to use new functions and so on.

The game within code lab is a role playing game where the characters wear armor and fight enemies with swords, spears and other medieval weapons.

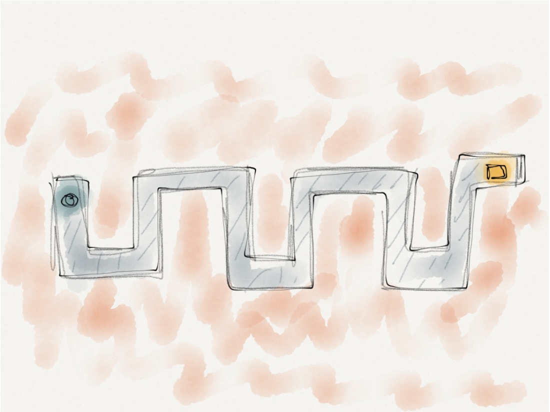

Fig. 4. A level that can be completed by writing a simple loop.

4.1 Progression

4.2 Reward and Motivation

4.3 Balancing

5 Platform

5.1 Art

5.2 Music and Audio

6 Production and Team

7 Competition and Inspiration

References

[1] O Anshus, Daniel Stødle, T Hagen, Bård Fjukstad, J Bjørndalen, L Bongo, Yong Liu, and Lars Tiede. Nineyears of the tromsø display wall, 2013.

[2] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.