Last Hope

Learning Framework
Scenario 2: Overview and Introduction
Team AC: Aksel Cakmak, Daniil Gannota, Gabriel Vanca, German Mikulski

Project title: “Last Hope”.
Target age-range: 10 - 15 years.
Materials, equipment and software required to create the project: C# programming language, Unity3D Engine, Adobe Photoshop, Microsoft Paint, Mac Paintbrush, Microsoft Visual Studio, MonoDevelop, GitHub, Todoist.
Equipment and software required to access the project: Operating system running Windows XP SP2+ or Mac OS X 10.8+. Support of DirectX 9+ for Windows or OpenGL 1.4+ for Mac.

Project objective: Our main objective for this project is to construct a viable, scalable and extensible educational framework that can be used by secondary school-students to learn programming on their own. At the same time, we aimed to inspire young minds about computing.

Project description: As mentioned above, the main idea of the project “Last Hope” is to create entertaining teaching materials for those making first steps in programming. Our game is structured in a way that it asks user to control character (mars explorer) by selecting and ordering certain programming commands.

No matter how distinct programming languages are, they all are based on the same concepts:

1. Functions to perform a certain action;
2. Conditional statements to perform an action only if something is true;
3. Loops, which help to set up iterative processes;

User will be able to use each of these elements to help our mars explorer, named “Last Hope” (as it is the last hope for humanity to explore the red planet, in our game scenario) to reach his target – a place, from which it will be able to move on to more adventures.

We use graphical interface to visualise programming concepts, that might sometimes be hard to understand for beginners. Our game directly shows which processes occur when a program is executed. Normally, these are hidden, and a user can only deal with input and output (I/O) of an application. We believe, that such background activities are especially interesting and have a large teaching potential for the novice programmers. Additionally, we try to illustrate the importance of these concepts, why programmers use them and the way efficiency of code can be improved.

As the user develops programming skills, we introduce new computing statements that will make every next level more complicated, interesting and enjoyable. We also make every level harder to develop a set of skills from making correct assumption to predicting character’s behaviour.

Lastly, working on Unity is quite a tough choice in a way that the program implies to know too many nuances, especially for a beginner. That is to say, programming on other engines/IDEs would be of much easier task.

Completed tasks:

1. Designed an attractive GUI for the game

2. Created the core functionality of the game with a possibility of easily adding more functionality and extending the game:
    a. Created a basic set of code blocks which can be used by the user to complete the levels. More code blocks can easily be added to the game.
    b. Added the drag & drop functionality
    c. Added the option to preview changes to the code
    d. Added the option to delete and reorder blocks and sets of blocks
    e. Created a simulated compiler that “compiles” the code blocks and transforms them into “machine code” (C# code running inside Unity3D)
    f. Added movement events e.g. reached final destination, fell into lava pool etc.
    g. Added the option to limit the number of times that the user can place a certain code block so that he is forced to create more intelligent and shorter programs
    h. Made the game screen and its content to properly rescale depending on the screen size
3. Created 11 levels that the user can access with the possibility of easily adding more levels
4. Added the option to save the user's game progress and the option for the user to navigate between previously completed levels
5. Created in-game tutorials for each level and also added in-game hints and information showed on pop-up on each level. These pop-ups can be used to directly access the tutorials. The tutorials can also be accessed for the main menu.
6. Added an in-game console which displays messages, warnings and errors from our simulated compiler
7. Done a lot of tests and bug-fixes

Outstanding tasks:

1. Add the option for the user to debug the code step by step
2. Add the option to convert the “program” made out of code blocks into a C# / Java program to teach the more advanced users how that code would translate into an actual programming language
3. Add more types of blocks (e.g. do while(), break(), continue(), jump())
4. Add more levels with bigger maps

Team Introduction:

Daniil Gannota @ I did not have particular programming experience before university. However, I used some applications available online that taught me basic concepts of computing. This is exactly what inspired me to do a game that will teach children in a similar manner. Moreover, the reason I like programming is because you have to follow certain logical steps in order to write software. Our game tries to show those processes.