BLG439E COMPUTER PROJECT I

DESIGN & IMPLEMENTATION OF TEACHING ALGORITHMS & PROGRAMMING GAME FOR KIDS

BARAN KAYA 150130032

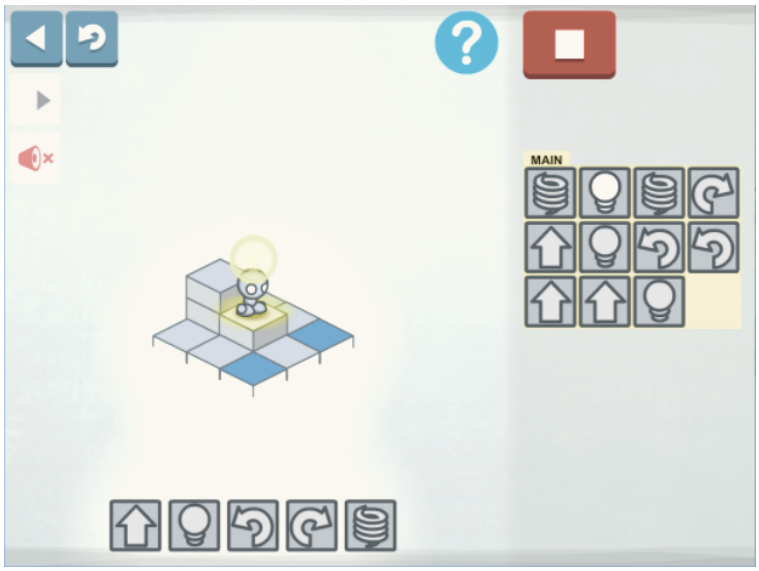
KADİR ENES KARSLIOĞLU 150130047

MEHMET ALİ OSMAN ATİK 150140804

December 2017

Example Games

1) Lightbot



Pros:
- Great animations and musics.
- 3 different level types: Basics, Functions, Loops.
- Easy to control and delete the movement blocks.

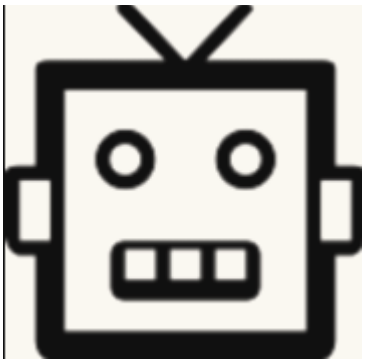
Cons:
- Does not show the errors that user made.
- Loops that this game teaches are recursive functions. Overall they are do the same thing but their logics' should be different than each other.

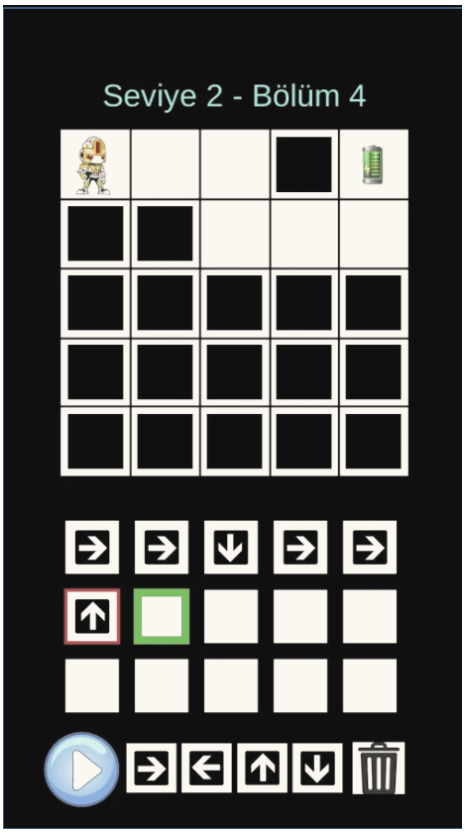
2) Code Karts

Pros:
- Good animations and musics.
- Lots of levels.

Cons:
- Full game is locked and can be unlocked with money.
- Levels are similar to each other.
- Every time game runs it starts with the first level eventough the higher levels are unlocked.
- Block colors are the same with the turn colors so user does not have to look at the road's direction, user can use the colors for the algorithm.

3) Çocuklar için Programlama (Programming for Kids)





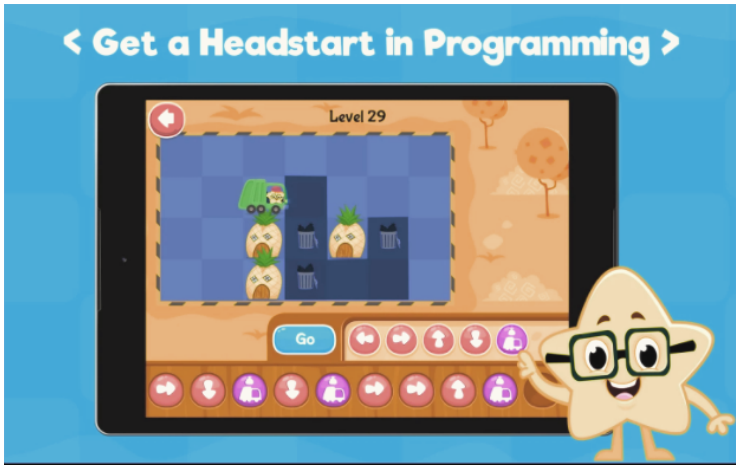
Pros:
- Basic animations and musics.
- Lots of levels.

Cons:
- Levels are similar to each other.
- Usability is not so good and easy.
- User can only move around the board and collect the batteries.

4) Coding For Kids – Learn To Code With Play

Pros:
- Great animations and musics.
- Different chapters that have different scenarios. All chapters have lots of levels.
- Teaches sequences, functions, loops and debugging.

Cons:
- Some chapters beginning and final animations and long and they play every levels start and the end.
- The game was not optimized well so that on my phone the game freezed a couple of times.

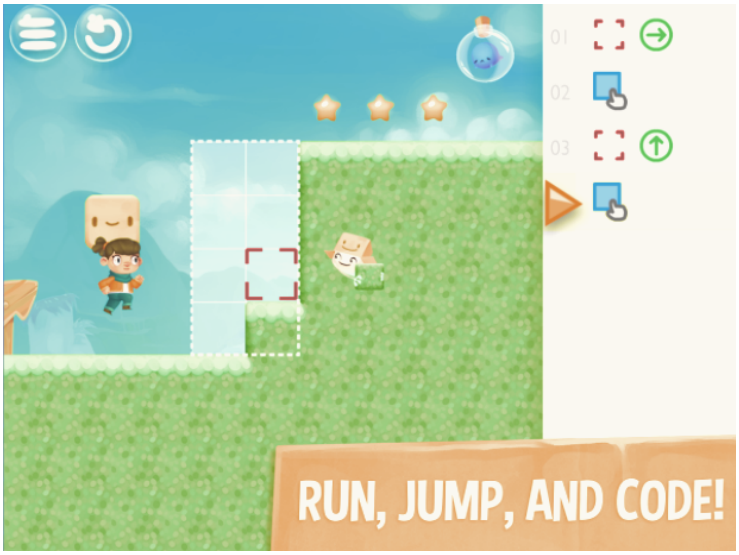


5) Coddy Free

Pros:
- Great animations and musics.
- Levels maps can be rotate so that user can see levels at different perspective.
- In the main menu it shows some programming informations like definition of class etc.
- Teaches basics, functions and loops.

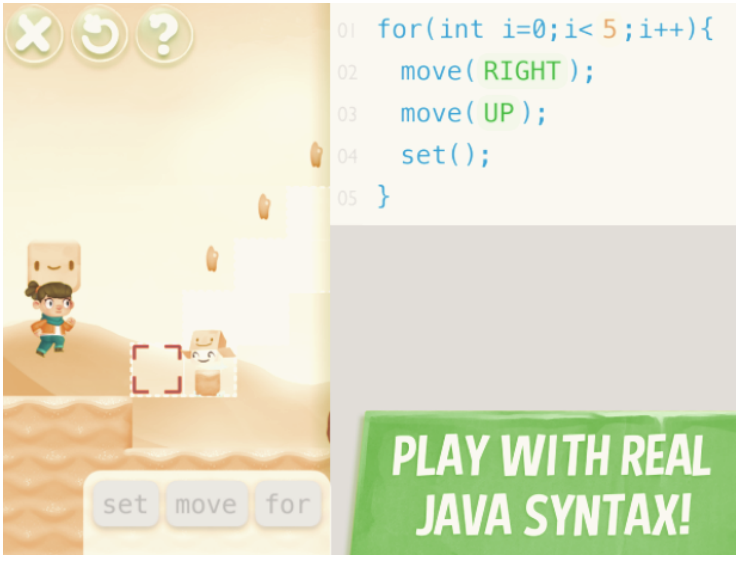
Cons:
- Full version is not free (120 levels).

6) Spritebox



Pros:
- Great animations and musics.
- The game does not have just algorithm parts it also have some movement parts like normal games.
- The game teaches java programming in it.

Cons:
- Free version has only 5 levels.

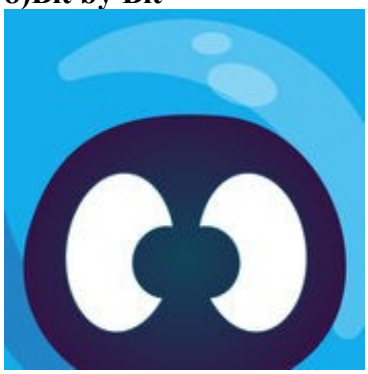


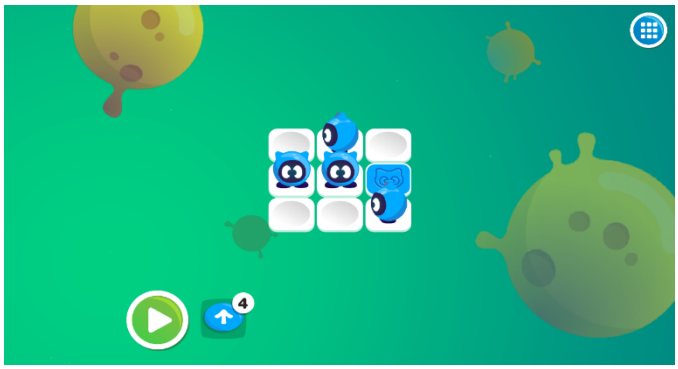
7) Run Marco!

Pros:
- Good animations and musics.
- It teaches basic instructions and sequence of commands, different types of iteration and conditional logic.
-The game is translated in 26 languages.

Cons:
- Levels are similar to each other.

8)Bit by Bit





Pros:
-Great animations and musics.
-Learning of planning ahead, creative problem solving, analytical and logical thinking.
-It can be played without knowing how to read or count.

Cons:
-Not easy to figure out some instructions in game.

9) Codespark academy with the foos

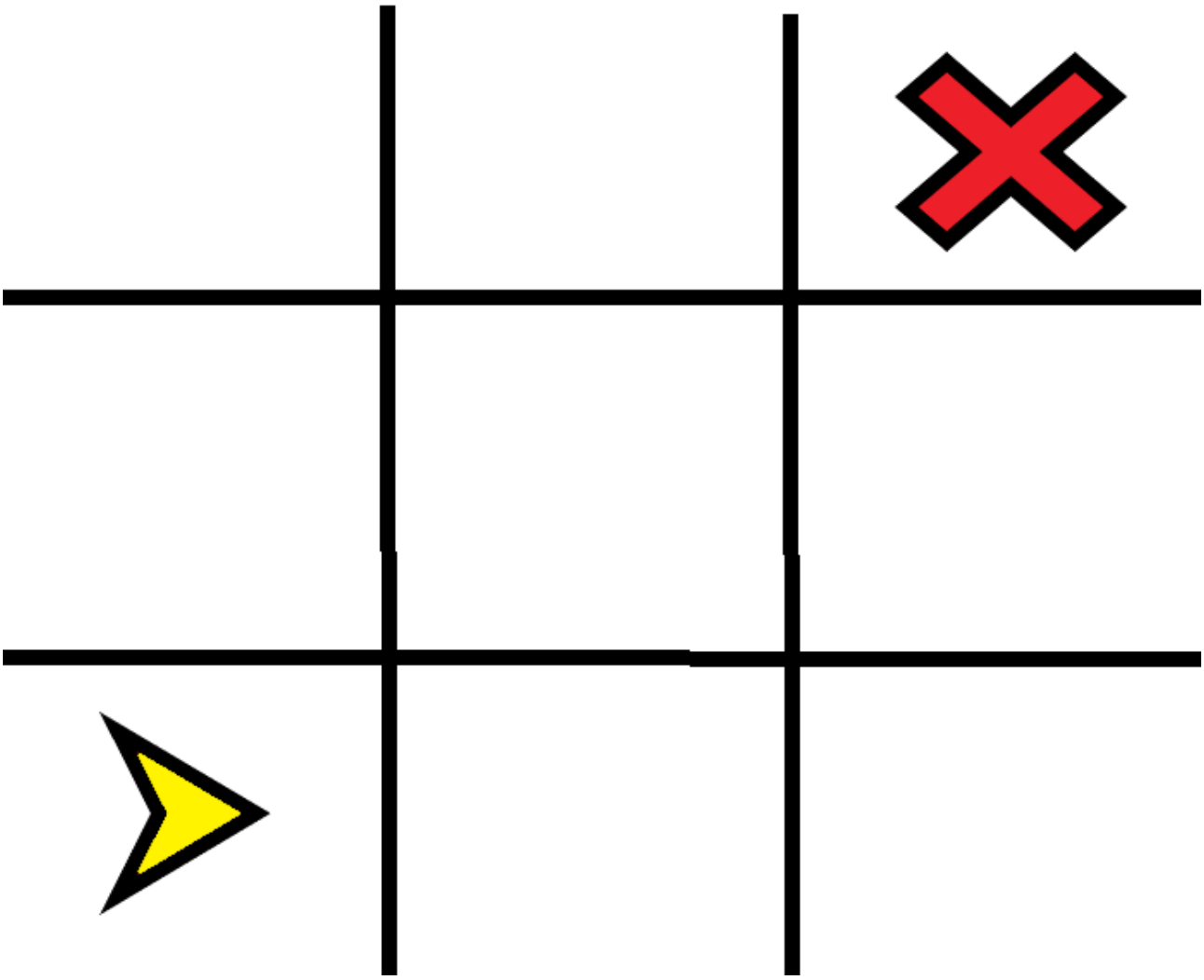
Pros:
-Great animations, musics and characters.
- Learning key programming concepts and use them to code.
-Based on research from MIT, Princeton and Carnegie Mellon.

Cons:
In-app help isn't available to guide kids.

Level Desings

We want to desgin a plane game that teahes the basics of algorithms for programming. Our game's main character is the plane that carries the parachuters. The puspose of the all levels is reaching the target location. However some levels require extra movements and different actions for compliting the level successfully. Level designs, characters, targets and other things explained below. For developing we tried different enviroments such as Unity, Android Studio and Stencyl. However we could not finish our game in time completly.

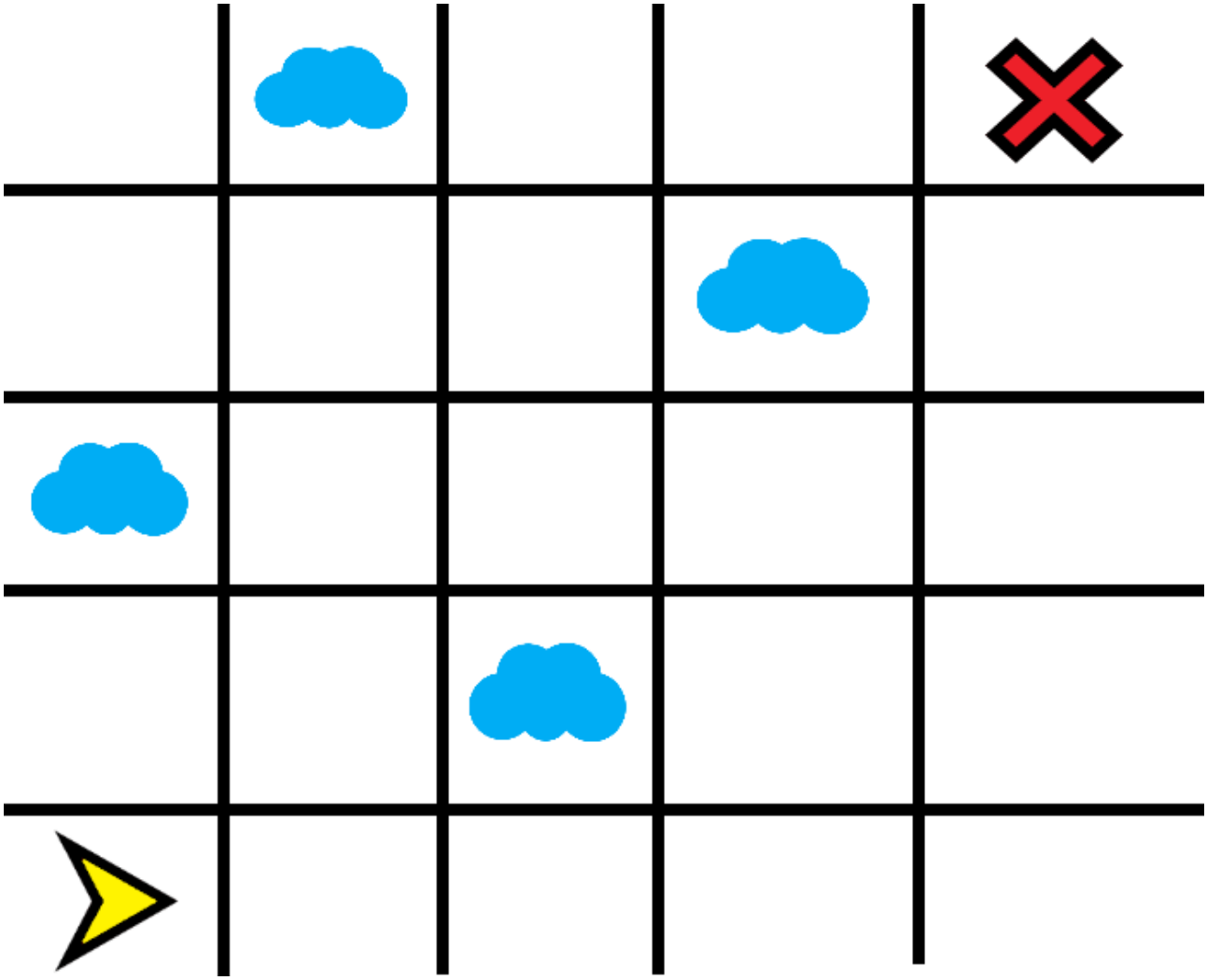
Level 1 – Basics



- Yellow arrow -> Plane
- Red X -> target location
- 8 block space -> User can use max 8 blocks in main
- 3 movement blocks -> Go Straight, Turn Right, Turn Left

Some movement examples
- Movement1: Go Straight, Go Straight, Turn Left, Go Straight, Go Straight (5 blocks)
- Movement2: Turn Left,Go Straight, Go Straight, Turn Right, Go Straight, Go Straight (6 blocks)
- Movement3: Go Straight, Turn Left, Go Straight, Turn Right, Go Straight, Turn Left, Go Straight (7 blocks)

Level 2 -> Obstacles



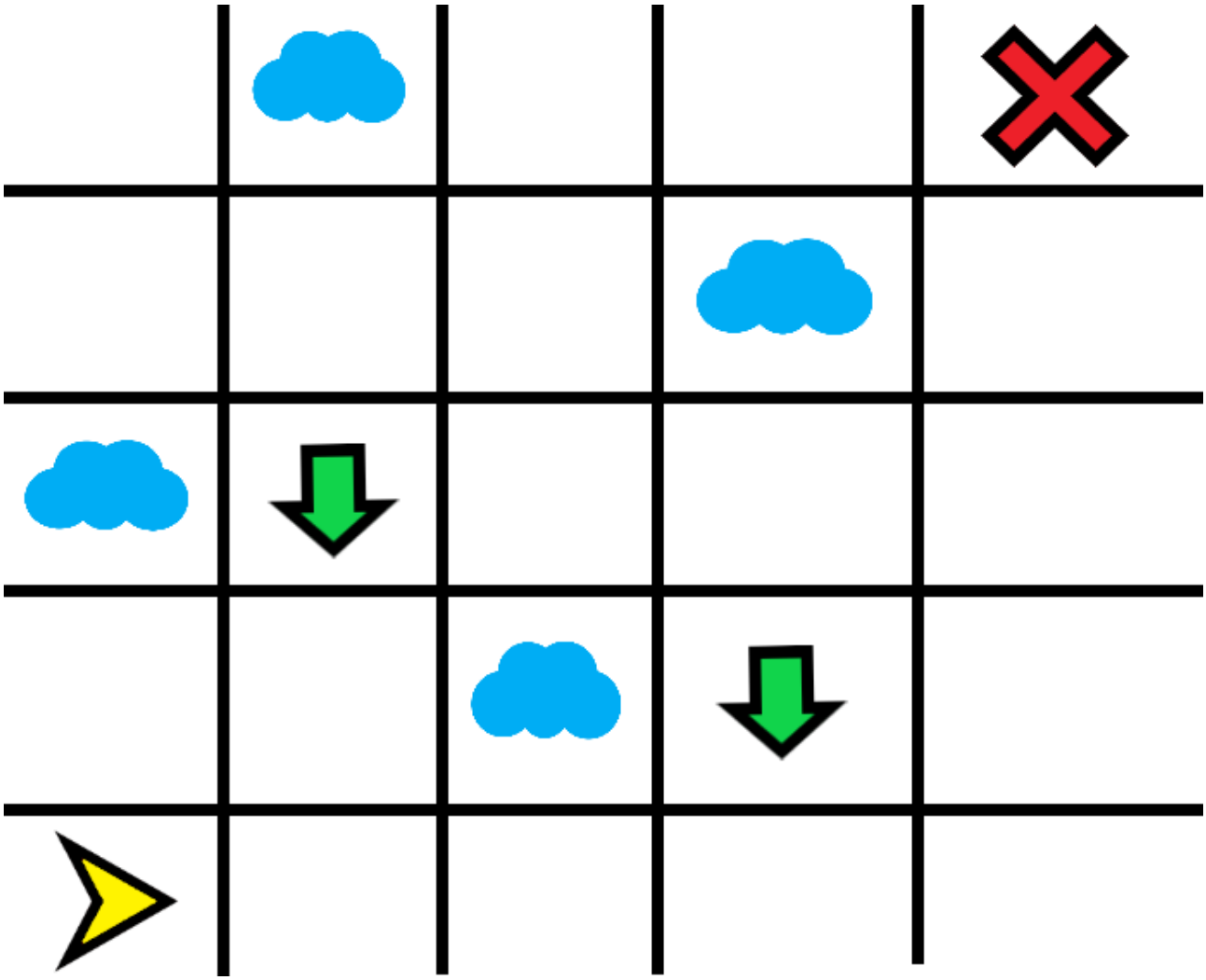
- Yellow arrow -> Plane
- Red X -> target location
- 20 block space -> User can use max 20 blocks in main
- 3 movement blocks -> Go Straight, Turn Right, Turn Left
- Clouds -> obstacles

Some movement examples

- Movement1: Go Straight, Go Straight, Go Straight, Go Straight, Turn Left, Go Straight, Go Straight, Go Straight, Go Straight (9 blocks)

- Movement2: Turn Left, Go Straight, Turn Right, Go Straight, Turn Left, Go Straight, Turn Right, Go Straight, Turn Left, Go Straight, Turn Right, Go Straight, Turn Left, Go Straight, Go Straight, Turn Right, Go Straight, Go Straight (18 blocks)

Level 3 -> Parachuters (If statement)



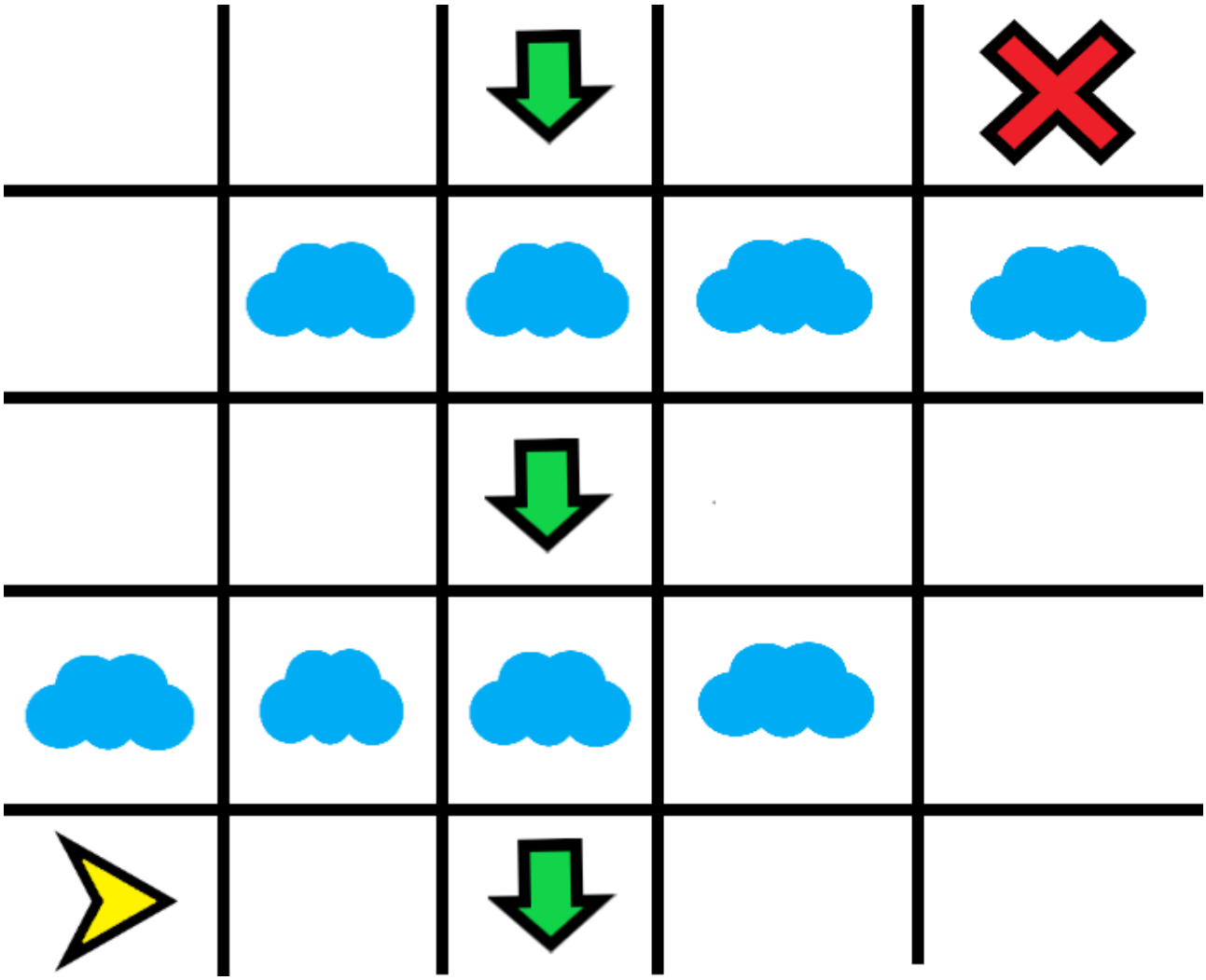
- Yellow arrow -> Plane
- Red X -> target location
- 20 block space -> User can use 20 blocks in main
- 4 movement blocks -> Go Straight, Turn Right, Turn Left, Drop
- Clouds -> obstacles
- Green arrows -> Drop point for parachuters

Some movement examples

- Movement1: Go Straight, Go Straight, Go Straight, Turn Left, Go Straight, Drop, Go Straight, Turn Left, Go Straight, Go Straight, Drop, Turn Right, Go Straight, Turn Right, Go Straight, Turn Left, Go Straight, Turn Right, Go Straight, Go Straight (20 blocks)

- Movement2: Go Straight, Turn Left, Go Straight, Go Straight, Drop, Turn Right, Go Straight,Go Straight, Turn Right, Go Straight, Drop, Turn Left, Go Straight, Turn Left, Go Straight, Go Straight, Go Straight (17 blocks)

Level 4 -> Straight road&same movement (Function)

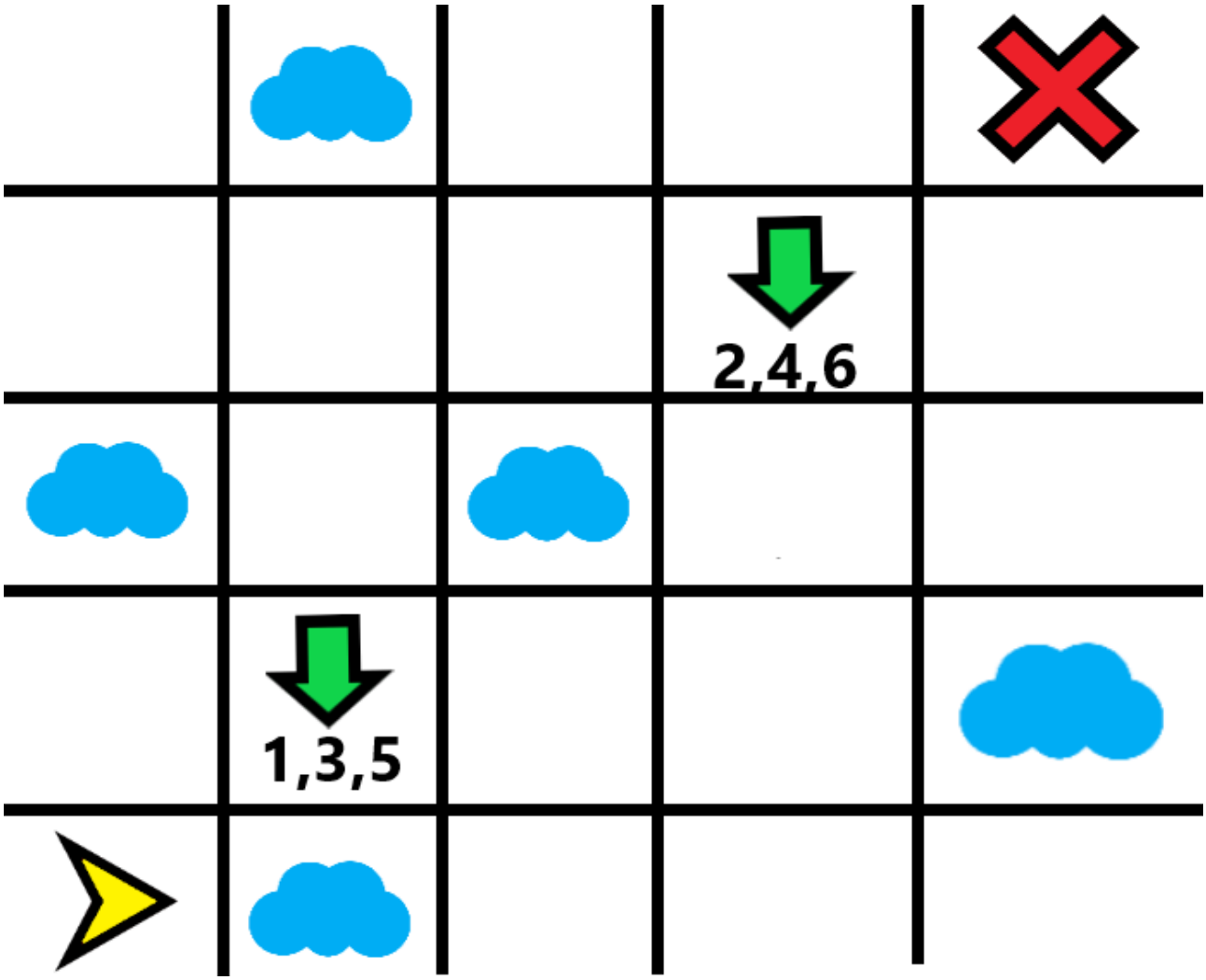


- Yellow arrow -> Plane
- Red X -> target location
- 11 block space for main
- 5 block space for function
- 5 movement blocks -> Go Straight, Turn Right, Turn Left, Drop, Function
- Clouds -> obstacles
- Green arrows -> Drop point for parachuters

Some movement examples
- Main: Function, Turn Left, Go Straight, Go Straight, Turn Left, Function, Turn Right, Go Straight, Go Straight, Turn Right, Function (11 blocks)

-Function: Go Straight, Go Straight, Drop, Go Straight, Go Straight (5 blocks)

Level 5 -> Same movement (Loops)



- Yellow arrow -> Plane
- Red X -> target location
- 12 block space for main
- 7 block space for function
- 6 movement blocks -> Go Straight, Turn Right, Turn Left, Drop, Function, Multiplier
- Clouds -> obstacles
- Green arrows and number-> Drop point for parachuters and the orders of the drops

Some movement examples
- Main1: Turn Left, Go Straight, Turn Right, Go Straight, Function, Multiplier (x5), Turn Left, Turn Left, Go Straight, Turn Left, Go Straight (11 blocks)

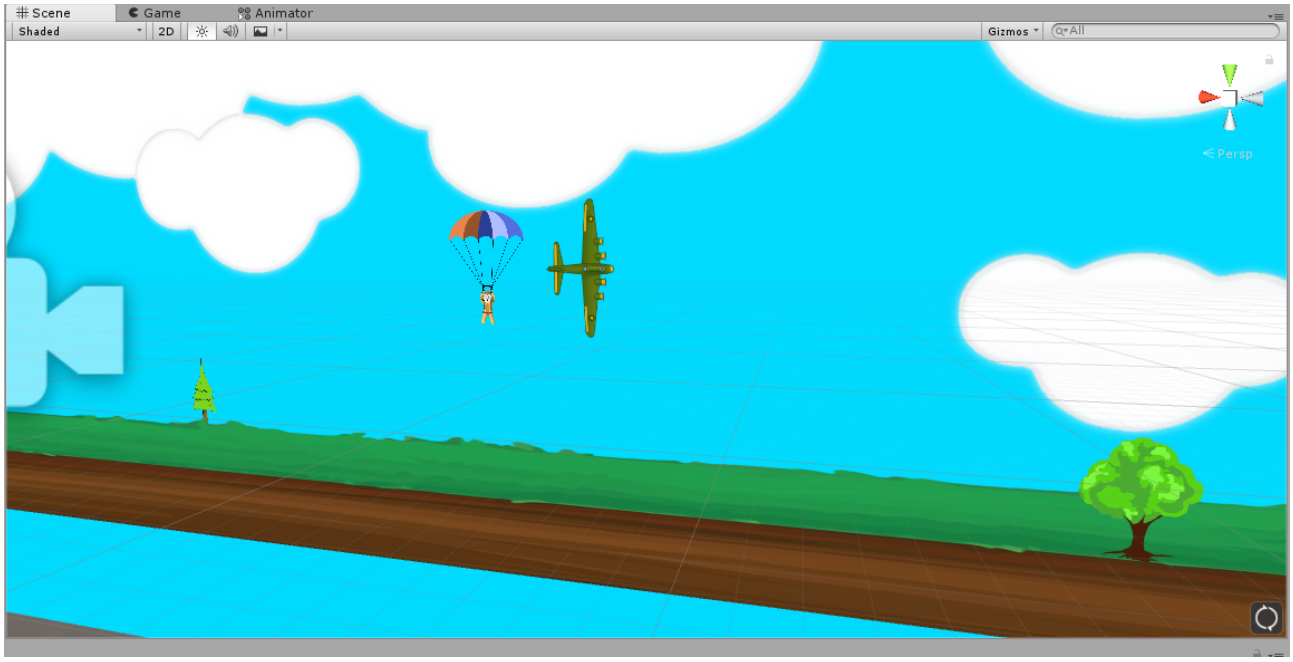
- Main2: Turn Left, Go Straight, Turn Right, Go Straight, Function, Multiplier (x5), Turn Right, Go Straight, Turn Right, Go Straight (10 blocks)

-Function: Drop, Go Straight, Go Straight, Turn Left, Go Straight, Go Straight, Turn Left (7 blocks)

Development Enviroments & Products

1) Unity Game Engine

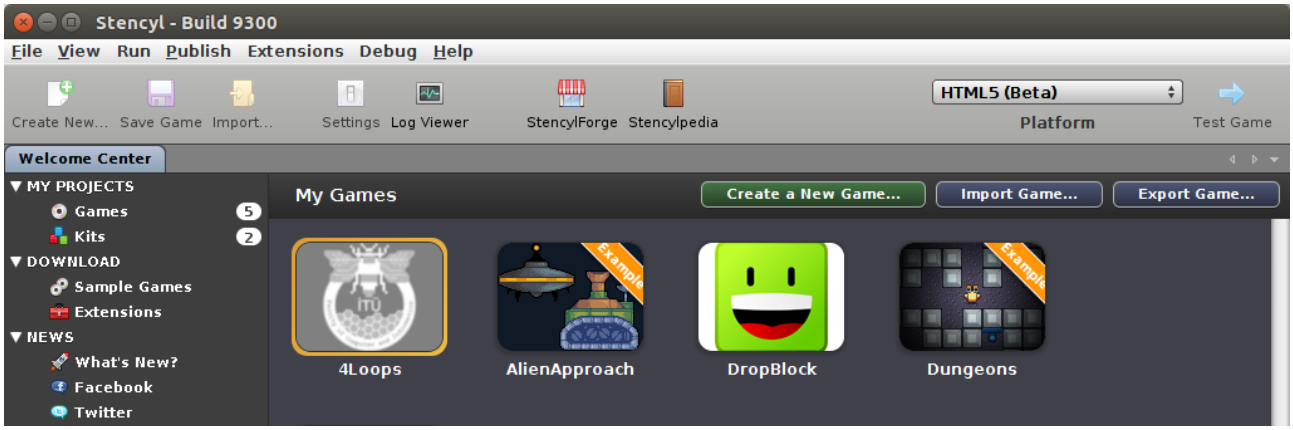
I (Baran) tried to use Unity for developing our game. However due to the lack of time i could not finish this project. I added some animations and characters to the game but could not go further. Therefore i can only share this project's development pictures. If I can finish this project plane will be our main character and the road will be the target for landing. Also ground and trees will slide for moving effect.



2) Android Studio

After trying Unity I (Baran) tried to develop our game with Android Studio. However due to the diffşculty of designing games with Android Studio and lack of time I could not finish this one. I designed and coded 6 different screens. One for main screen and the other 5 for each levels. Main screen has buttons for every levels and Quit button. However inside of the each level there are only movement buttons (Go Straight, Turn Left, Turn Right) and no other things like characters or obstacles.

3) Stencyl



Stencyl provides multiplatform game development tools with no programming skills required. I (Ali Osman) tried to develop an other game for this project and named it 4Loops. It is a simple maze game which the road to exit is obvious but character movements needs to be programmed within single or consequtive for-loops to reach the exit according to the level difficulty. For loop format should be a in standart format with modifyable input areas for control such as :

- **For [...] times do [... + ...]**

which can be modified by user as :

- **For [4] times do [→ → → + ↓ ↓]**

which wold be identical to a for loop:

```
for ( i=0; İ<4; i++){
  move_left;
  move_left;
  move_left;
  move_up
  move_up;
}
```

and also consequtive commands would be possible such as :

- **For [2] times do [→ → + ↓]**
- **For [1] times do [← + ...]**
- **For [2] times do [↓ + →]**

I designed three simple levels for demonstration of usage of the game, on all levels it is sufficient to follow the obvious path to reach the exit but not mandotory.

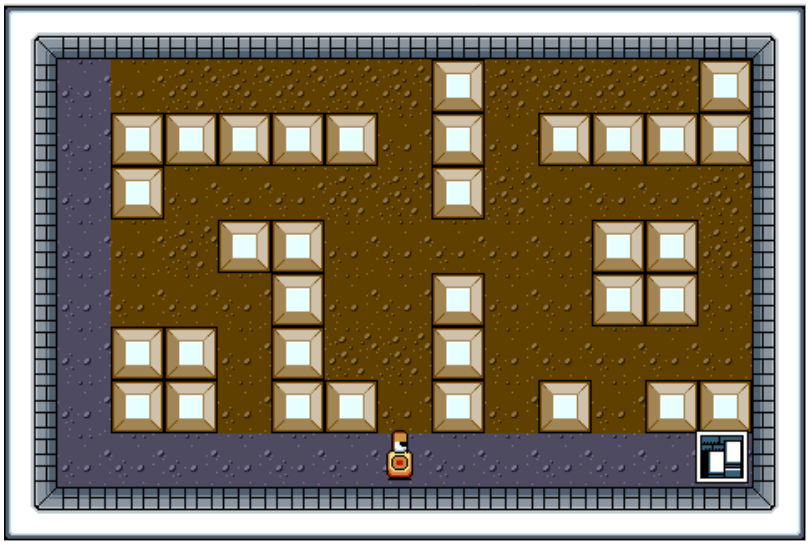
Level 1 : can be solved by simply **For [1] times do [** + **]**

or alternatively

For [7] times do [+ **]**

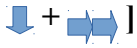
followed by

For [12] times do [+ **]**



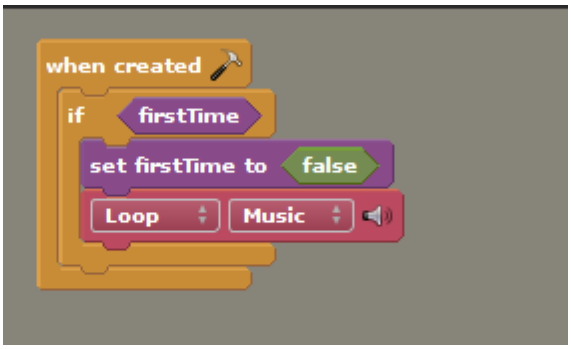
Level 2 : can be solved by simply **For [2] times do [** + **]**

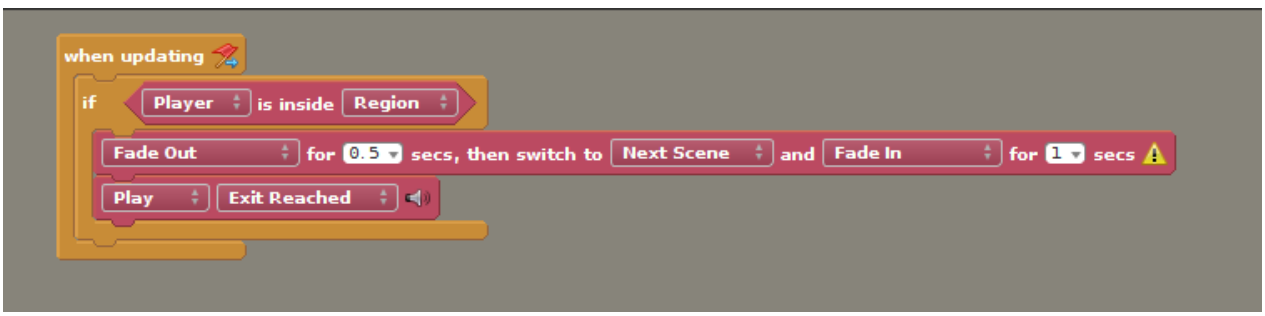
alternative ways are always possible but going into dirt is not desired.

Level 2 : can be solved by simply **For [7] times do [**  **]**

It is obvios that last loop wil not finish but reaching to exit is enough to pass the levels.

Design screen gives provides building blocks of shapes and textures, controlling behaviours of game are simply done in visual logic blocks and does not require much programming skills.





Controlling stack is not implemented yet.

Contributions

Kadir Enes Karslıoğlu:

- Research and examination of some model applications.

Baran Kaya:

- Examination and design of our Android application on Adroid studio and Unity.

M. Ali Osman Atik:

- Research and development of our Android application on Stencyl.