User Manual and Documentation

Maria Floruß

September 15, 2014

User Manual

Controls

General

	Keyboard	GamePad
restart level	R	Back

Circle

	Keyboard	GamePad
move left	A	Left Thumbstick - left
move right	D	Left Thumbstick - right
press button	S	Left Trigger

Square

	Keyboard	GamePad
move left	Left Arrow	Right Thumbstick - left
move right	Right Arrow	Right Thumbstick - right
jump	Up Arrow	Right Bumper
press button	Down Arrow	Right Trigger

Menu

	Keyboard	GamePad
toggle menu	Escape	Start
navigate menu	Arrow Keys	DPad
confirm	Enter	A

Debug

	Keyboard
toggle physics drawing	F9
cycle through debug info	F10
cycle through visual helpers	F11
zoom in	Numpad +
zoom out	Numpad -

Game Mechanics

You start in the game with two players. They have to help each other to get to the end of the level (usually on the right). As you enter the tunnel at the end of the level you will be redirected to the next level.

There are several game objects, that can help the players to reach their goal.

Buttons

Buttons can be triggered by either square or circle exclusively. There are two kinds of buttons:

blue buttons can toggle the state of a platform or obstacle. If you press it once, the platform will stay visible.

yellow buttons need to be pressed down in order to move or toggle platforms and obstacles.

Platforms

Platforms are horizontal, physical objects, that help the players getting somewhere. They can be triggered by a button. As there are two buttons, there are two kinds of platforms.

blue platforms will be toggled on and off by a blue button. Once the button is pressed the platform stays visible until the button is pressed again.

yellow platforms will be triggered by a yellow button. They are only active as long as the button is pressed. They can either move or be toggled on and off.

Obstacles

Obstacles are vertical, physical objects, that usually are in the players way. They can be triggered by a button. As there are two buttons, there are two kinds of platforms.

blue obstacles will be toggled on and off by a blue button. Once the button is pressed the platform stays visible until the button is pressed again.

yellow obstacles will be triggered by a yellow button. They are only active as long as the button is pressed. They can only be toggled on and off.

Idea and Design

Finding an Idea

At the beginning I played 'Brothers - A Tale of Two Sons', developed by Starbreeze Studios and published by 505 Games released on 7 August 2013. In this game the player takes control over two brothers, who are controlled by the left and right part of the gamepad respectively. This singleplayer co-op principle was very impressive to me and I started thinking about games like this, where one player controls two characters. When we started with the lecture I already had some ideas about how my game could have looked like. There were many ideas, of which many did not make it into the final game due to complexity and time constraints.

Ever since I started developing the game I wanted to make it aesthetically appealing as well, even though the art would not influence my grading.

Development

Box2D

As the physics are a major part of this game and I didn't have the time to do it all by myself, I used Box2D as a physics engine. At the beginning this was a bit cumbersome, because I didn't know how to use it properly, but in the end I figured it out and everything went better than expected.

Level Generation

In order to get the collision of my relatively complex levels I wrote a level generator, which would allow me to draw my levels as .bmp files. A red dot would mark the beginning of a static physical body, which would contain one or more edges. A black dot would determine the end of an edge and the beginning of a new one. The gray lines would connect the red and black dots.
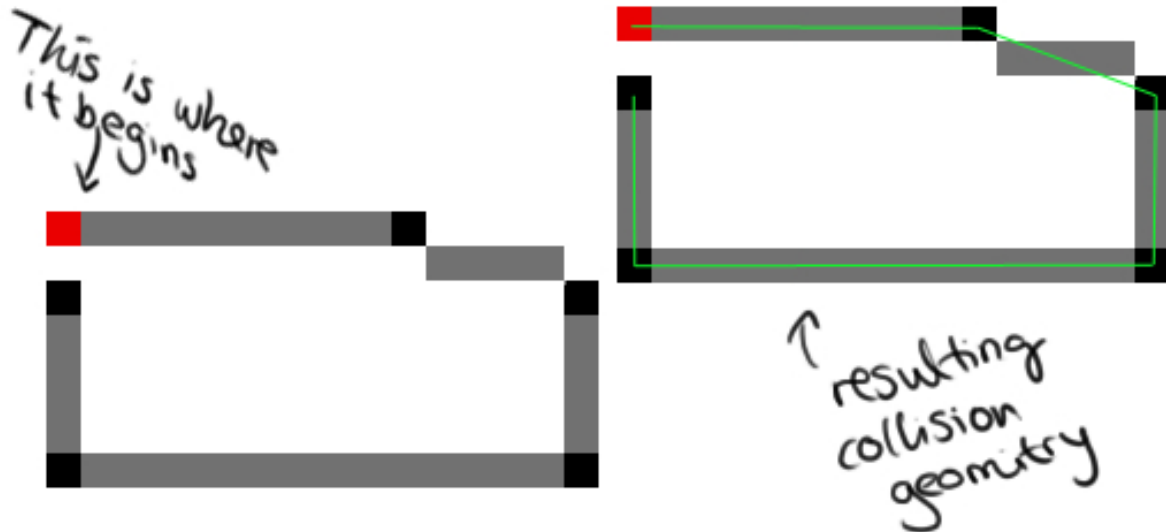
Configuration Files

To describe my levels, e.g. the position of the game objects, the maximum speed of the players or the connection of a button to the according platform, I used a configuration file reader, written by Manuel Maier. This allowed me to balance my game and create new levels very easily. Internally the game objects and players use the configuration files to initialize themselves.

Excerpt of a configuration file:

```
1   [circleTrigger]
2   type = toggleButton
3   drawOrder = 20
4   position = 1145, 457
5   proximityRadius = 20
6   master = circle
7   state = inactive
8   onButtonOn = togglePlattform
9   onButtonOff = togglePlattform
10  textureOn = objects/button_circle_1_on
11  textureOff = objects/button_circle_1_off
```

Camera

The camera is a modified version of Khalid Abuhakmeh's post on stackoverflow.com: `http://stackoverflow.com/a/723918` (date: 2014-09-09).

I had to add some features, because I had two players, which should be followed by the camera. In order to achieve this, I used a proxy game object, which is positioned in between circle and square. I also added view bounds, to make sure the view port won't leave the level.

Audio

For audio I'm using XACT which comes with XNA. It is mainly used to play the background music in a level. Due to the complexity of the API I wrote an `AudioManager` that wraps the API and provides just the functionality that is needed for my game.
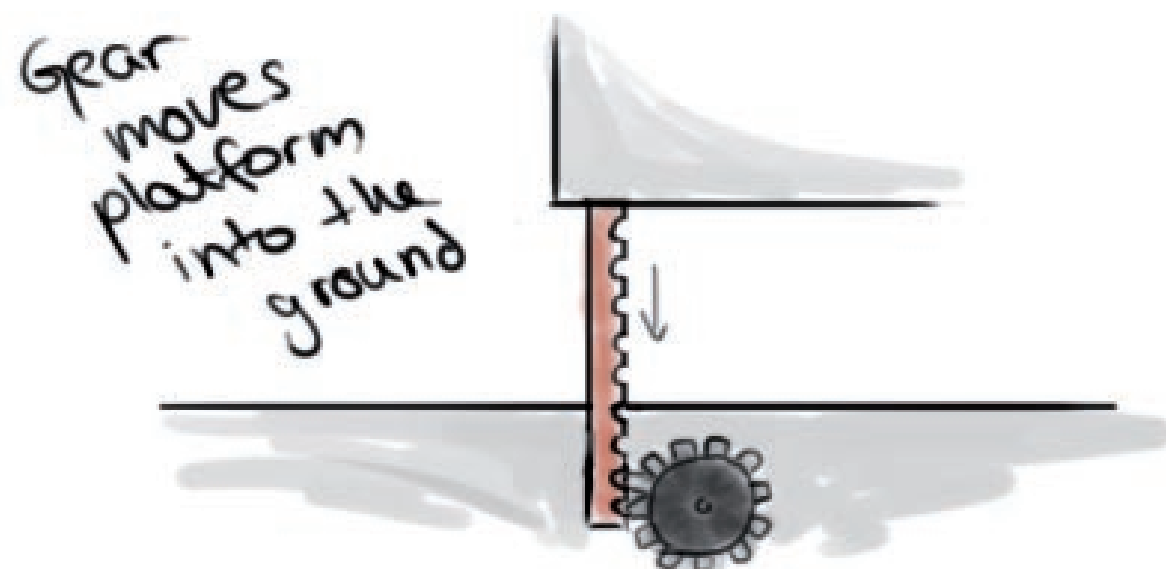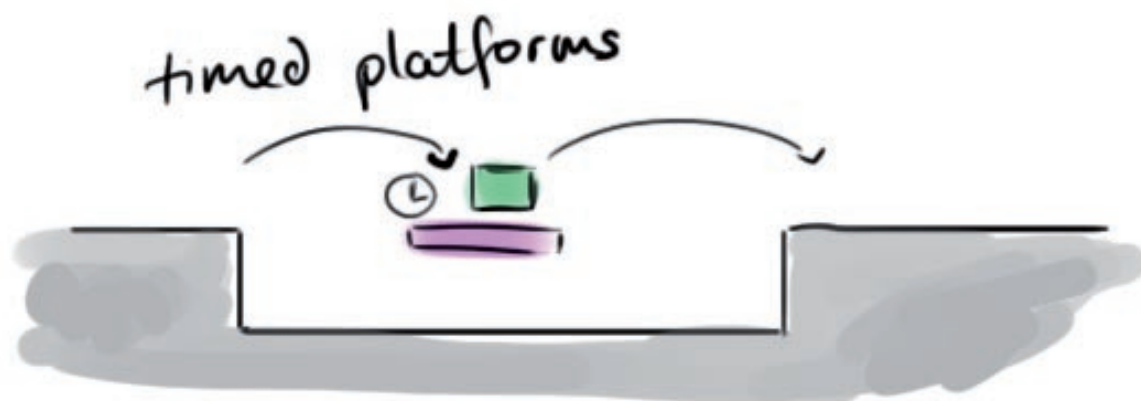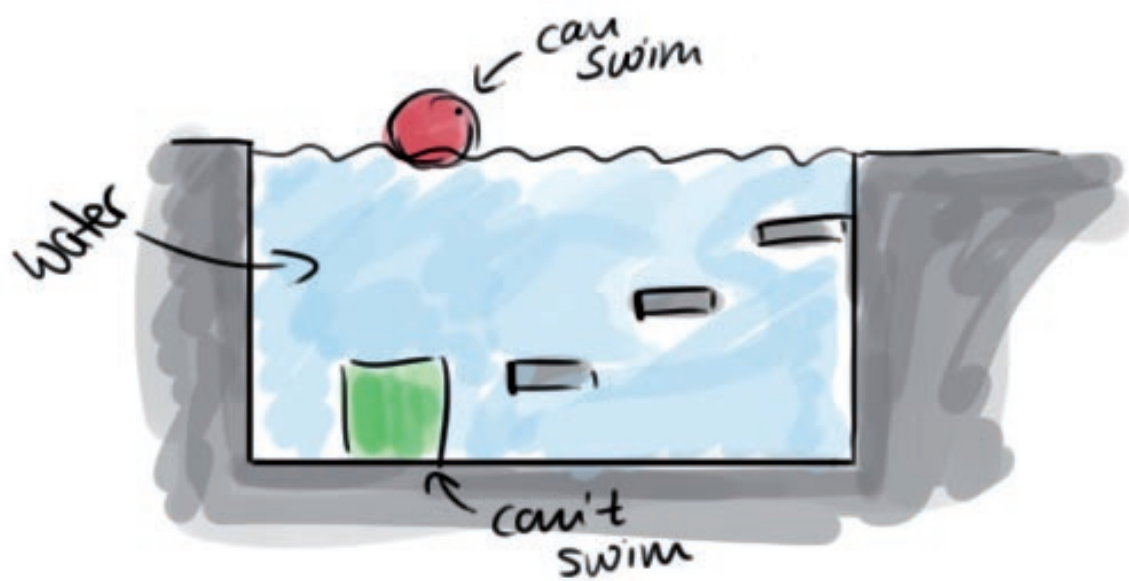
The following files are taken from bigcat_smauls at `http://www.freesound.org/people/bigcat_smauls/`

- 222802_bigcat-smauls_like-a-dream-loop.wav

- 223722_bigcat-smauls_keep-at-it-loop.wav
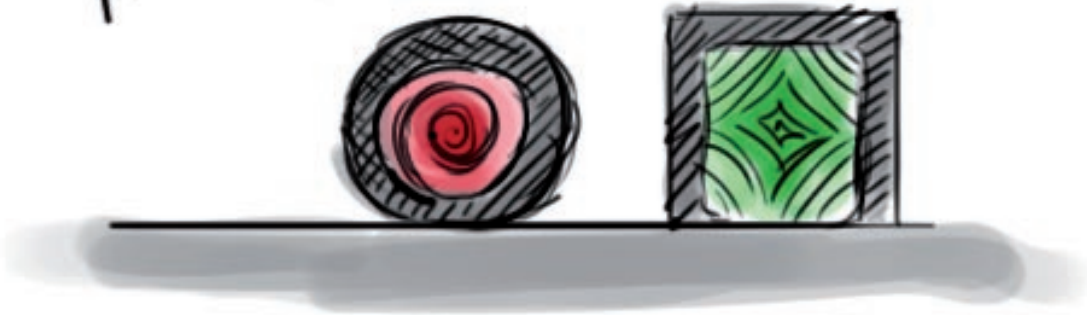
- 222800_bigcat-smauls_gur-durge-loop.wav

Licensed under the Creative Commons 0 License, these items are in the public domain.
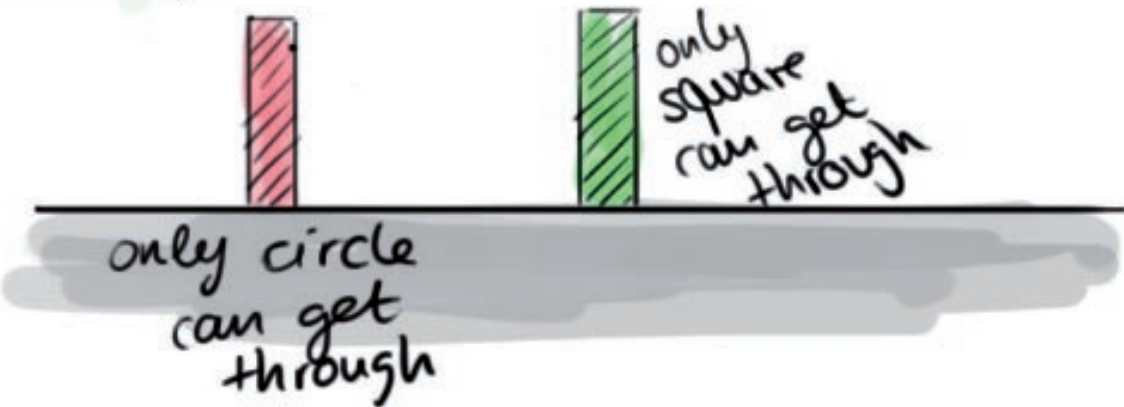
Future Concepts

The following is a collection of concepts that sadly did not make it into the final game due to complexity and time constraints.

can swim

water

can't
swim

timed platforms

Gear
moves
platform
into the
ground

portals

semipermeable obstacles

only square
can get
through

only circle
can get
through

ramp for circle