



C++ Project

The Revenge of the Pigs

ELEC-A7151

Meri Mäkelä, Tuomas Rislakki, Aleksi Sohlman, Henri Välimäki



First draft of the graphics

1. Overview	2
1.1 Introduction	2
1.2 Features and functions	2
1.3 Graphics	3
2. Structure	4
2.1 Classes	4
2.2 Files	5
2.3 Modules	5
3. Libraries	6
3.1 SFML	6
3.2 Box2D	6
4. Division of work	6
5. Schedule	6

1. Overview

1.1 Introduction

Our object is to create an Angry Birds-like shooting game. The object of the player is to destroy all the birds hidden in between destructible and indestructible blocks. The game will keep track of the points of the player according to game logic.

The program is used by clicking on icons on the interface and possibly scrolling with the arrow keys. For shooting a pig, click and drag is utilized.

1.2 Features and functions

The game will have 3 different throwable pigs, all with different special abilities. One will have no ability, the second will explode when activated and the third will get a speed boost into the direction it is facing. There may be more pigs added depending on time-frame.

The levels for the game are coded in files to make the game more easily expandable. There will be at least 3 playable levels which vary in difficulty. The levels consist of breakable and unbreakable objects among which the birds hide. Some of the unbreakable objects will be static, representing hills and other terrain features.

When playing a level the score of the player and the amount of remaining birds is displayed as a number on screen. The remaining pigs can be seen either from a line, like in the Angry Birds games or as a list with the next pig on top.

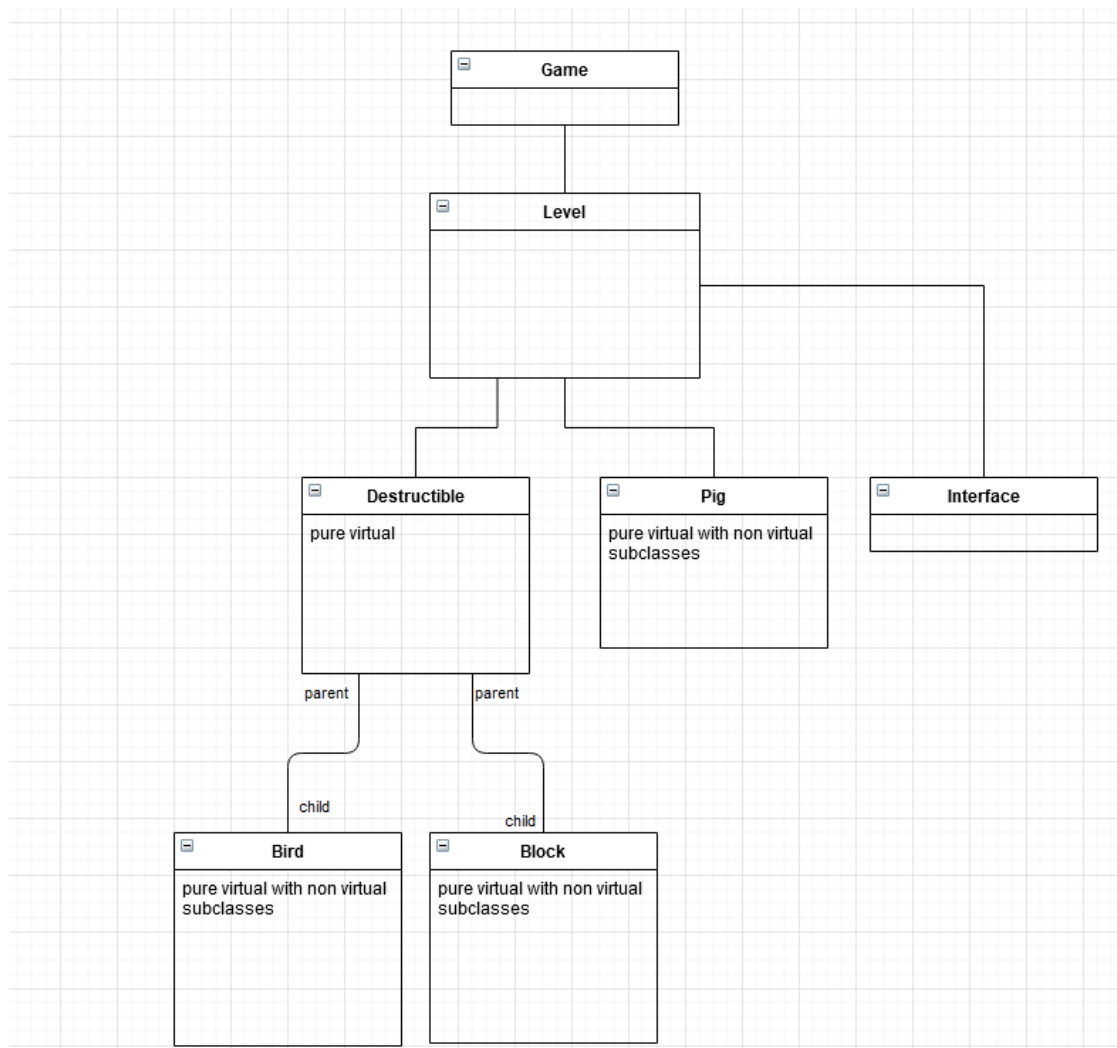
Opening a level will be done in a level selection screen where the player will see as many icons as there are levels and clicking on one will load up the level. It is yet unclear will it be possible to transfer directly from a completed level to the next without going to the level select.

Features that are the first to be added past the baseline are the high score tables and a star system. The high scores will be viewable from a separate selection screen where the player chooses the level and it shows then the 10 best scores for that level. The star system will be implemented by adding point values which are required for a certain amount of stars. The amount of stars will be displayed in the high scores list after the points.

1.3 Graphics

We aim to design some of the graphic ourselves. Items for which we will be drawing the pictures include the cannon with which the pigs will be shot, the birds which will be shot at and the pigs itself. The different types of pigs and birds will have different designs.

2. Structure



2.1 Classes

2.1.1 Pigs

Pure virtual class with three subclasses for different types of pigs. The attributes include position, speed, mass, orientation. Virtual method is the special ability. Destructor is called when the pig stops moving for x amount of time.

2.1.2 Destructibles

Pure virtual class which is inherited for the birds and blocks classes. It has the attributes position, speed, mass, orientation, hit points, points on destruction. The attributes may be subject to change when implementing the immovable and indestructible blocks.

2.1.3 Level

Contains the information of the loaded level. Information is loaded from file. Attributes include a count of all the remaining birds, amount of points gained, a list with the remaining pigs, and a list / vector with all of the objects in the level. Also possibly a current pig which is the currently shot pig, removed from list of pigs beforehand. If the star system is implemented a vector with the required points for the star.

2.1.4 High score

The class constructs a high score list from a file with a maximum of 10 scores. The class has a vector that holds all of the high scores in the format of pairs, nickname and score sorted by score. When the destructor is called it writes a new high score file with the possibly updated information.

2.1.5 Interface

Responsible for all multimedia properties, the class is not yet fully refined and may merge partially with another class. Attributes are probably the scene and graphical objects and mouse position.

2.1.6 Game

The main loop responsible for calling for updates. This will be refined later. For now this includes the menus but they might become own classes later on.

2.2 Files

There will be at least one type of file used to save the levels and if the implementation of the high scores is done the scores will have their own type of file.

2.2.1 Level file

The file will consist of blocks to group the objects together. Each block starts with a distinct header that starts with a #. On each row separated by semicolons are the needed parts of initializing one object. The level files will all be in the same directory and will differ from each other by their number.

2.2.2 Score file

The file will have up to ten rows each consisting of two parts the score and the entered nickname separated by a semicolon. The files will be in their separate directory and will have matching numbers with the level.

2.3 Modules

The modules are not yet clearly defined, but very probably the individual classes and separate helper functions like file handling functions will be their own modules.

3. Libraries

3.1 SFML

This library will be used for all multimedial purposes and detecting mouse and keyboard input.

3.2 Box2D

The physics library will be used for the simulation of the game to calculate the effects of the pig hitting something.

4. Division of work

Phase 1

- * Game + main loop - Aleksi
- * Level - Tuomas
- * Destructables - Meri
- * Pigs - Henri
- * Pictures - Tuomas

Phase 2

- * File handling - after objects done
- * Interface (multimedia) - to be decided
- * Cmake - to be decided
- * Level design - to be decided after level class done

Phase 3

- * Possible animations and pictures - to be decided
- * High Score - to be decided
- * Others?

5. Schedule

- Project plan ready 17.7.
- Phase 1 - 31.7.
- Phase 2 - 15.8.
- Phase 3 - if applicable 25.8.
- Project ready 28.8.