

ANGRY BIRDS PROJECT PLAN

05.11.2021

1. Personal information

Developers:

Vadim Kuznetsov 907271

Aleksander Truss 994925

Shahram Barai 729983

Islroilkhon Salikhodjaev 783437

2. Overview

Angry Birds is a Finnish game developed by Rovio Company.

In this game, the player uses a slingshot to launch a bird into pigs placed on the surface or inside various structures, with the intention of destroying all the pigs on the playing field. When a player passes this level (or episode), a new level and new birds are opened for him.

The player hurls birds or similar objects towards a fortress to destroy enemy targets. The player should use as few throws as possible for a higher score. Fortresses are built of destroyable and non-destroyable construction elements. Failing, not destroying all the enemies, typically means the level is reset and all points are lost.

The game has very simple rules, but as in any well-crafted game, the player's points should correlate with their throwing skills. This project subject does not require you to make a clone of an existing game, but to implement a one with the same general idea.

Basic features

- These features are the minimum requirements for this project.
- Basic graphics, no animation needed. The view follows the bird as it moves sideways.
- At least 1 bird with a special action, for example: speed boost or explosion.
- Controlling the attack with mouse (at least throwing and special action).
- Game levels are loaded from files, not hardcoded.
- At least 3 game levels with increasing difficulty.
- Simple user interface that shows information such as points and throwables left, enemies left.
- Physics simulation

Additional features

Implement some additional features if you're aiming for a good grade. You can suggest your own additional features if they stay relevant to the project. If so, remember to ask the teaching assistant overseeing your project if the feature would be worth some additional points.

These features below usually exist in games. Points of the features are listed with them. You can make other features too, check for example Angry Birds games for inspiration! Just remember to document the idea for the feature well enough before implementation.

- High score list, saved per level, player can enter a nickname
- Stars to rate how well player played and some logic to calculate them
- Level editor to create levels and to save them in a file
- Sounds effect
- Better graphics; animations, nice textures, camera animations etc.
- Different game modes: reach goal in time, collect all items, other kind of challenges
- Multiplayer: game can be played from a "client" and a "server side" stores the scores to one place

3. Description of the use case and draft of the user interface

The program opens a window where it draws the level, its objects, and the game character. The program responds when the user clicks and moves the mouse. When the user presses the left mouse button program starts responding to mouse movements. When the left button is released, the bird is launched. Usually, other objects on level do not move until they interact with birds. Then created gravity should move them. In the original game, pigs die when some other object hits them.

4. Structure plan of the program

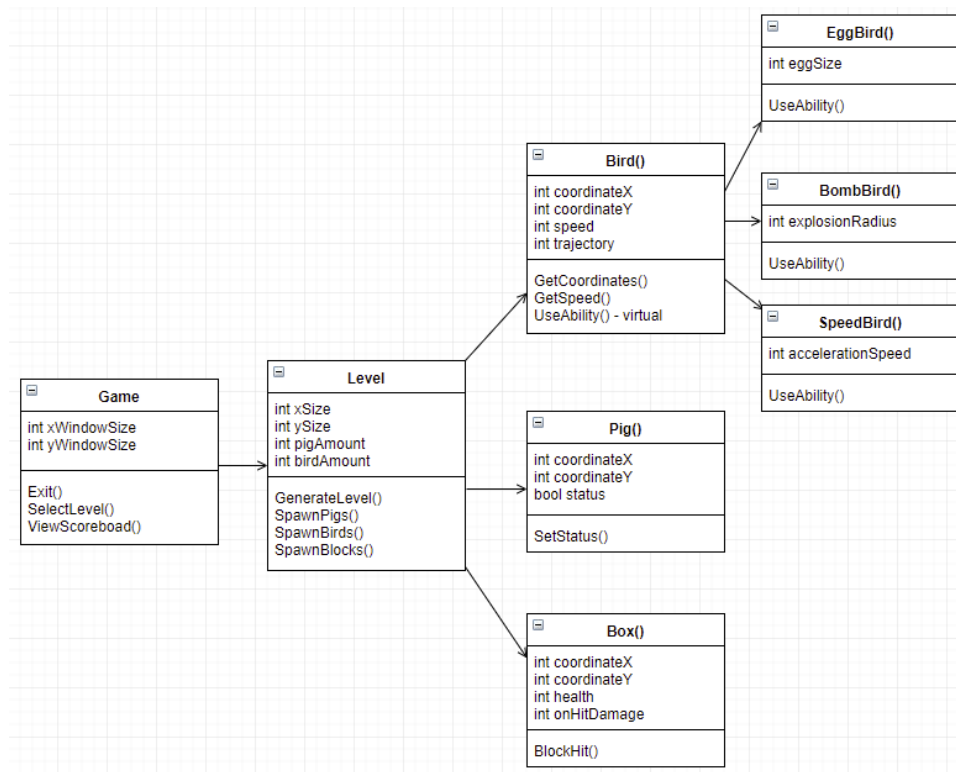
Class Bird() = Describes a game character and includes methods for movement. It detects collisions with other objects. There is a parameter that tells you the position in the xy coordinate system. Subclasses will have individual movement parameters or ability.

Class Block() = Describes blocks from which structures are built. It is a base class for other blocks. Subclasses will have their individual health parameter that describes how fast it breaks.

Class Pig() = Describes enemies on level. Has parameter dead which is true or false. It is true when it is in touch with bird. When pig is dead it gets removed.

Class Game() = Describes launched game. Has methods for launching level, viewing scoreboard and exiting.

Class Level() = Describes levels of game. Has parameters for creating level from file. Places blocks, birds and pigs in their positions.



5. Data structures

Game doesn't require a lot of data structures. At this point, we would like to use text file with characters. All items will be created based on characters. For example P for pig and B for block. Scoreboard will be saved in a text file.

6. Algorithms

Idea is that most methods are based on the comparison of xy coordinates with other objects. If they are equal something happens. At this moment we think all the algorithms will be based only if-else and loops.

7. Test plan

Games get tested by playing them through multiple times. Even large game projects contain errors after publication. The most important things that are not allowed are the game crashing and the game characters should not fall off the game level. The classes have to be tested to be working as intended. So are functions. Visual bugs should appear first, then the 'invisible' ones will be discovered.

8. Libraries and other tools

We will use Qt library and Box2D. Qt will be used for drawing objects and Box2D for simulating physics.

9. Timetable

We will try to complete project before 1. December as later starts exam week. Optionally, it is in our greatest interest to complete the project as early as possible to have more time to test it for possible bugs and optionally add additional features.

10. Responsibilities between the group

We cannot decide responsibilities yet as we do not know what issues we are going to have. But ideally, we will divide the workload fairly, and since some of the members are more advanced in the library and other topics, they might guide others that are not so aware. Overall, we plan to work together as a team and help each other with the issues that might appear.

11. Literature references and links

We will use guides on the Internet and video tutorials.

One of the possible reference videos are: <https://www.youtube.com/watch?v=9iYjOkRDzBs>