ENTERTAINMENT AND INTERACTIVE SOFTWARE DEVELOPMENT

Create Functional, Interactive Software

Michael Cassar

MCAST | BSC4S

CONTENTS

GAME DESCRIPTION [PASS 2.1]

BACKGROUND STORY

The game was inspired by breakout style games, such as "Ricochet", the classic "Game Pong", and "Angry Birds", and incorporated various features to come up with a unique game play experience. The game features a paddle, from "Pong", a ball from the breakout game "Ricochet" and also physics enabled objects that can be hit by the ball, similar to "Angry Birds" but here the physics enabled objects are boxes which are situated on white pedestals. The game features retro graphics, and a simplistic design.

PLAYER TASKS

The player must successfully knock boxes off of the pedestals to make them green in order to complete levels. This must be done whilst retaining enough lives.

LEVEL DIFFICULTY

Level difficulty increases by significantly increasing the ball speed and the added generation of extra pedestals. This means that it will be harder to hit all targets while containing the ball within the game environment. In addition, lives are not reimbursed upon level swaps.

SCORE CALCULATION

Scores are only increased when a ball successfully collides with a box causing it to fall off. If a box is hit by another box, scores will not be granted so it is important to plan the balls movement to get maximum points per level. Each valid collision awards 1 point. You have 3 lives, if you lose the lives by allowing the ball to exit the game environment, from the left, the game will end.

THE ULTIMATE GOAL

The ultimate goal of the game is to pass both levels without losing the allotted number of lives, in order to successfully win.

HOW TO PLAY

The aim of the game is to use the ball to hit all the red boxes and make them turn green by dropping them off of the white pedestals. The paddle on the left hand side of the screen may be used to contain the ball within the game environment as failure to do so will result in losing a life. The mouse may also be used to fine tune the balls movement within the environment to enable a certain amount of accuracy when trying to hit the available targets. Pressing the hotkey 'k' allows the user to save the ball location, and pressing it again takes the ball back to the saved location.

MOCKUP
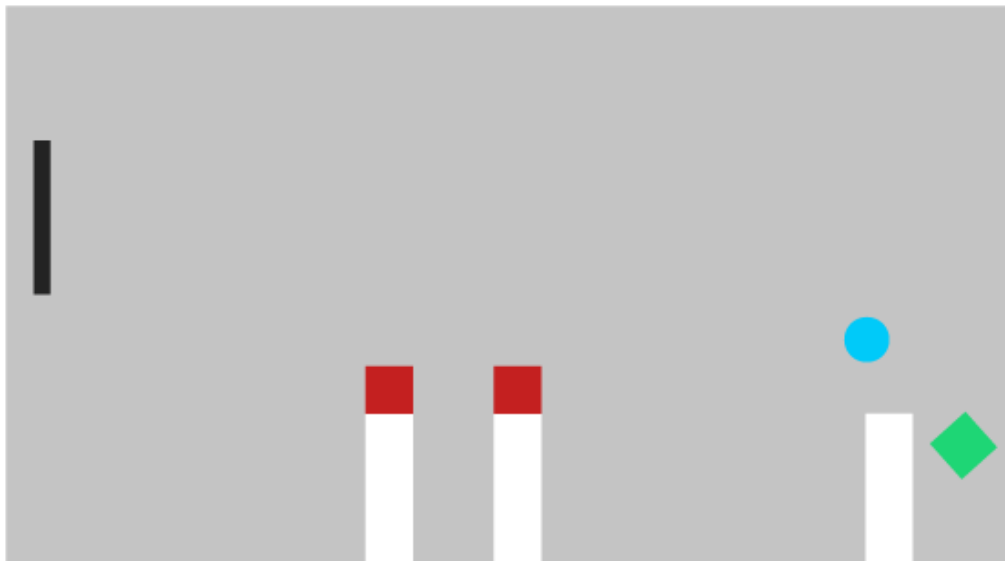
GAME PLATFORM DEPLOYMENT [PASS 4.2]

Due to the fact that platforms operate differently a number of code implementations will have to be tweaked to cater for the hardware and software integration of the target platform. First and foremost it is important to handle input functions correctly. If the game needs to be run on a PC as well as a Mac, keyboard support needs to be implemented, furthermore if the game is able to be ported on a mobile device let's say iOS or Android touchscreen capability needs to be included, maybe also, in certain cases on screen controllers might need to be added.

The second problem when it comes to porting is optimisation. This can mostly be seen in Mobile applications where hardware is somewhat limited due to the fact that smartphones are in their early stages specification wise. In order to handle this certain code functions may have to be ironed out or rebuilt to cater for certain efficiencies that make the game playable on mobile devices, these could be things from loading times to fps issues.

Another thing that comes to mind is catering for different monitor resolutions and sizes. It is not always as easy to port a game to a mobile device just by building to that target platform, issues may arise where sprites or user interaction / user interface objects may become too small or too large to allow for successful porting. The behaviour of objects will have to be tweaked via scripts or even possibly have to have custom levels created to allow for a successful port.

All in all, the above procedures could have extra practices added to them to enable a refined port, and must also be built via the project to that target platform.

IMPLEMENTATION COMMENTS [DISTINCTION 2.1]

QUESTION 1

Having had the time to program a similar game on a larger scale, I believe it would be more fun to play. It still is, but certain concepts could possibly need refinement for it to enable some level of immersion.

QUESTION 2

1. The addition of custom sounds where necessary.
2. The ability to randomly generate an infinite amount of levels.
3. Greater detail in animation, such as the bounce squashing when it bounces and particle effects.
4. A settings panel for graphic customisation.
5. A settings panel for audio customisation.
6. The addition of extra play modes.
7. Refining of the overall graphical design, using custom sprites and textures.
8. Adding shadows by having the light source at an angle.
9. Auto saving and allowing the user to load the last saved game to continue playing.
10. Ball physics could be tweaked to cater for a more realistic bounce.

IMPLEMENTATION COMMENTS [DISTINCTION 2.1]