Project Plan: Other-1 (smash)

ELEC-A7151 - Object oriented programming with C++ F2020

Antti Huttunen, Anssi Lehtonen, Eevi Rimmi, Juho Tuomaala

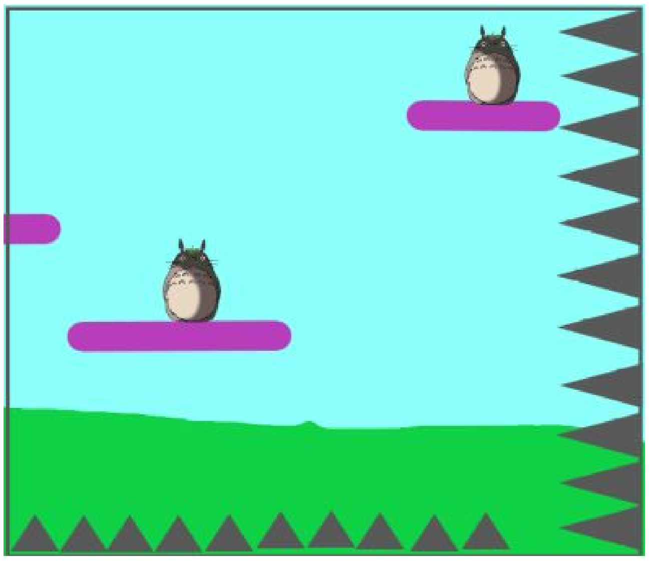
INDEX

Scope of the work

Two player 2D fighting platformer game with gameplay distantly reminiscent of Super Smash Bros.

The game is a local multiplayer, in which the keyboard is shared by two players. The players try to push each other off the level, which consists of multiple platforms. The level is constantly moving, so the players will have to successfully jump their way through so as to not hit the sides or the bottom, both of which kill on impact. Players use a keyboard for moving and attacking. Music and Sound effects are added to the game.

There is at least one attack button, which will shove the competitor sideways if successfully hit.



Bonus features(If there is enough time):

- Ammo situated on the level for a special long ranged attack, which incapacitates the competitor for a few seconds.

- Extra attacks.

- Level editor

- Different creatures

High-level structure of the software

Main modules

main.cpp & main.h
responsible for running the game

gui.cpp & gui.h
responsible for game window

creature.cpp & creature.h
responsible for characters

map.cpp & map.h
responsible for platforms (immovable objects) and background picture

physics.cpp & physics.h
responsible for physics and NPC-logics

game.cpp & game.h
responsible for combining all the elements of the game and running the game loop

test.cpp & test.h
contains unit tests

Main classes

Logic classes
Includes all the logic that goes into the game (position, actions, movement, stats etc.)

Game class
Owns and contains creatures, platforms etc.

Creature classes
Objects to represent player (or NPC) characters

Map/platform classes
(possibly) divided into subclasses of different tiles that the creature objects interact with

contains information about the position and size of the tile

Graphics classes

each graphics object “owns” at least one corresponding logic object, and updates graphics based on the object’s member values/methods

- **GUI class**
  - Owns all the other graphics objects
  - calls graphic drawing and updating methods

Libraries

Graphics & sounds
From Qt-framework:

- **QtWidgets**
  - **-QGraphicsScene**
  - **-QGraphicsView**
  - **-QGraphicsItem**
- **QtGui**
  - **-QKeyEvent**
- **QtCore**
  - **-QTimer**
  - **-QObject**

Physics library: Box2D
The creatures will be implemented with gravity to enable satisfying jumping.
Bonus: pushable boxes (used as projectiles against the opponent).

Responsibilities

Name	Title					
Antti	Algorithms Designer	core class structure: Window & Keypress	character design	unit testing	Game physics & attacks	
Eevi	Graphics Designer	core class structure: Item/Platform	Map design & Content	unit testing	Other game logic	
Anssi	Game Designer	core class structure: Scene/View	Map design & Content	BG music & sound design	Game physics & attacks	
Juho	System Designer	core class structure: Character	character design	sound design	Other game logic	

Schedule

Qt platform setup for all team members -> create window	30.10.2020-6.11.2020
Square movable by wasd	30.10.2020-6.11.2020
Single box that walks and jumps on rectangles	6.11.2020-13.11.2020
Two fighting boxes, sides and ground kill -> cause program to end	13.11.2020-20.11.2020
Ok graphics	20.11.2020-27.11.2020
Menu, win state screen, program runs until quit	20.11.2020-27.11.2020
Unit testing	6.11.2020-4.12.2020
(Bonus features, including better graphics)	27.11.2020-4.12.2020
Demo session	7.12.2020-11.12.2020
DL Documentation in git	11.12.2020
Project deadline	**11.12.2020**