Concept and technical pitch

BE LOVE

Cart253 - Creative Computation I Section B

Professor Pippin Barr

Ariana Hipsagh | 22689812 | November 7, 2017

SUMMARY

The game, Be LOVE, is about a child who is frightened by a monster in his bedroom at night. He jumps off his bed to collect love (which is everywhere) as his weapon in the hope of transforming the monster into a cuddling teddy.

The child can move using the traditional arrow keys, wasd, and can go airborne by clicking the mouse. The monster moves from the closet to the edge of the room. If the monster makes it across the bedroom (from left to right) before the child has collected enough hearts, the monster wins and the child is left rocking back and forth.

On the other hand, once the child has collected the required number of hearts, the child can dive bomb the monster. The computer randomly decides if the monster will then transform into a teddy or the child is left rocking back and forth.

MEDIA

Below is free art downloaded from the internet that may be used in the game.

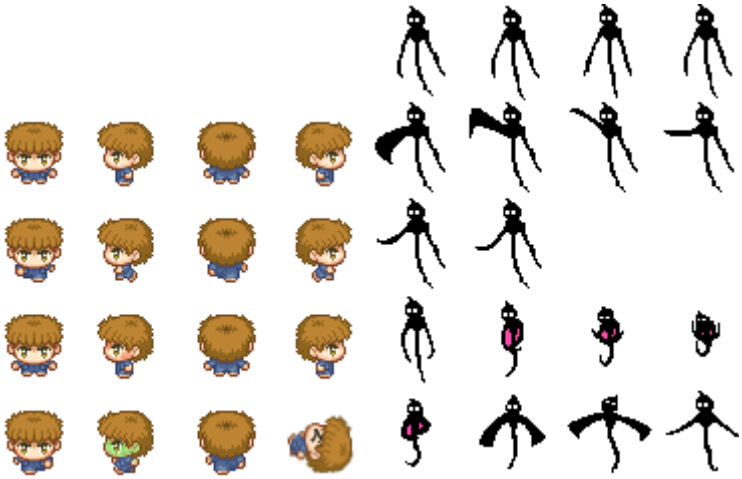

Figure 1 Player

Figure 2 - Monster

Figure 3 - Explosion

Figure 4- Background

Figure 5 - Collectible

The player (figure 1) is woken by a noise in his bedroom (figure 4) at night. From the closet emerges the Monster (figure 2) who begins to creep across the room.

The player collects love in the form of hearts (figure 5) by collision. The hearts appear and disappear at random. The player can use the arrow keys, wasd, and mouse click to go anywhere in his room to collect love.

Racing against a timer of 42 seconds, 42 hearts are required before the player can dive bomb to collide with the monster (figure 3). Just like in life, there are no guarantees that all the love will result in the monster transforming into a teddy.

The game will randomly decide whether it ends with a teddy or a child rocking back and forth.

INSPIRATIONS

Be LOVE is Inspired by Bruce Lee who is quoted to have said:

"Be like water making its way through cracks. Do not be assertive, but adjust to the object, and you shall find a way around or through it. If nothing within you stays rigid, outward things will disclose themselves.

Empty your mind, be formless. Shapeless, like water. If you put water into a cup, it becomes the cup. You put water into a bottle and it becomes the bottle. You put it in a teapot, it becomes the teapot. Now, water can flow or it can crash. Be water, my friend."

In the game Be LOVE, Bruce Lee's quote about water is replaced by love. Of course, the meaning of life is closely linked and therefore our timer and number of collectibles required to win the game of life is 42.

TECHNICAL APPROACH

Player – collisions with hearts and monster. The more hearts collected the more the colour of the player changes (maybe a health meter as well). Sound effects include the player walking, the hearts appearing, disappearing and collisioning.

Monster – collision with the player. The monster grows (health meter) as it moves across the bedroom. The player can reduce the size (health) of the monster by dive bombing hearts. Sound effects include dive bomb explosion, monster sounds, monster collision with player.

Collectible – The hearts are generated randomly at random locations in the room and they have a random life span. They are collected by collision with the player. Sound effects include the hearts appearing, disappearing and collisioning.

Score – The game is scored by counting the number of hearts collected during the 42 second session. No guarantee that the monster will be defeated even if a successful dive bomb of 42 hearts takes place. The game is randomly decided.

GameController – The game controller has the splash page, the main level and the credits page. Sound effects include background, win and lose

TECHNICAL RESEARCH

This game design uses concepts and knowledge acquired throughout the course.

The resources that I will use include:

CART253 slides from the class by Pippen Barr

Learning Processing by Daniel Shiffman

Learning-Processing-Exercise-Solutions

YouTube Channel the Coding Train by Daniel Shiffman and the supporting github library https://github.com/codingtrain