



# XBlast

Prototype Description





# **XBlast**

McGill University  
ECSE 321 – Winter 2014

Authors:

David Liu  
Yike Liu  
Eddy Lu  
Xuzhi Shu  
YaHan Yang  
Christopher Reny

April 2<sup>nd</sup>, 2014



## Java Code

---

### Discussion of Code

The majority of the code is implement as described in the SAD.  
A few new things were added.

### Multiplayer LAN Support

Uses class Client to send data and class Server to handle received data.

IP address of the other computer is taken as the only argument by Client class.

The Client class sends a String, which is a list of commands to the other computer. Every time a command is created, it is concatenanted into the String sendCommand. The Client reads the sendCommand until it finishes reading a complete command, sends the command, and then removes that command from the String sendCommand.

The server receives the command, stores it into a list of commands that will eventually be executed and then execute the commands by the order of arrival.

Client and Server run on different threads.

### Multiple enemies and bosses

There are 8 different kinds of enemies and 4 different bosses.

Enemies are capable of various attacks such as throwing scissors, charging at player, and kicking bombs.

Each different type of Enemy is a new class that extends the abstract Enemy class. Their speed, hp, exp/score provided, and abilities are hard coded into their respective classes.



## Multiple stages

12 stages are to be included. Every 3 stages, there is a boss.

Stages are loaded by the class LevelLoader. The LevelLoader loads stage based on the int variable currentStage. It calls the respective load method based on the current stage. The map is initialized with a 2 dimensional int array. With each GameObject mapped to a different integer, the int 2 dimensional array provide a visualization when setting up the map.

Besides GameObjects, there are indestructible obstacles in the game. Those are simply images rendered onto the map. A Placeholder object is put into the grids they take up to prevent the player and enemies from moving into them. All Placeholder objects are not rendered, so they act as transparent obstacles. The visual is provided by whatever image is render above the Placeholder objects.

## Animations

3 ways were implemented for animations. The first is using spriteSheet, second using .gif files, third is using array of images.

After trial and error, the third way is discovered to be the most efficient and flexible way to animate stuff. Thus most of the animations are done with array of images. Fragments of animation done with spriteSheet and gif still remain.

A class ImageSequence is used to store animation data. This class takes are argument a string, which is the path of the first picture in the sequence, and a int which is the number of picture in the sequence. The images must be properly name with numbers xxx (1),xxx (2),xxx (3).....

Each instance of ImageSequence includes an array of images loaded from the input path, and some variables for the width, height, x and y position of the images.

Images are automatically scaled to fit inside a grid. However, they could also be manually scaled to whatever size desired.

## Player voice data

The player will say set lines when certain conditions are met. These conditions include finding an item, taking enough damage, dead, leveling up, and using some abilities.

A Class called PlayerVoice stores the voice data.



# Gameplay

---

## Gameplay

When first the running the game, the user is presented with a main menu in which he or she has the ability to select one of three options: Story, Arcade, and Network. In order to navigate and highlight a choice, the user simply utilizes the computer's keyboard's left and right arrow keys. To select the choice, the user presses 'z' on the keyboard.

## Arcade Mode

Once the user selects the Arcade Mode, he or she is introduced to a character menu where five different characters can be chosen. Similar to the main menu, navigation is controlled by the left and right arrow keys and selection is done by pressing 'z'.

Once a character is chosen, the user is presented with another menu where the character attributes can be manipulated. These attributes include Hp, Mp, Soul, Speed, Damage, and Range. In order to highlight each attribute, the user navigates using the keyboard's up and down arrow keys. Furthermore, the attributes are increased and decreased by using the right and left arrow keys respectively. In addition, the user is allowed up to 40 points to spend on these different attributes. Once manipulations of these attributes are completed to the user's content, the user can now begin the game by pressing 'z'.

After pressing 'z' to begin the game, the user is presented with a quick animated introduction sequence before spawning in the actual game map. Instantly, the user witnesses the different enemies moving throughout the map. The map also implements a grid to make navigation easier for the user. There are also different obstacles in pre-designated locations. In order to move the character, the user simply presses the up, down, left, and right arrow keys. To drop a bomb, the player presses the 'z' key. During any point of the game, the player also has the ability to pause the game by pressing the 'p' key where a menu is displayed. This menu contains the options: Resume, Settings, and Quit. To return or go back, the user presses the 'x' key on the keyboard.



When a bomb is placed, it detonates within a certain timeframe. The bomb also acts as a physical barrier which the user cannot cross that corresponding space. If the character is in a gridblock that is filled by fire from the detonated bomb, the character loses health. This health is displayed on the bottom of the screen. On the other hand, this health also regenerates at a constant rate over time. If the fire of bombs reach obstacles such as stone walls, the obstacle is destroyed and a power up may spawn in place of the bomb. These power ups have different abilities such as increasing health.

Another attribute displayed on the bottom menu is the Mp. This attribute and display bar works similarly to the health bar in which it decreases when the character uses Mp and increases at a constant rate. The bottom menu also displays the different special abilities that the character has acquired.

After destroying all of the enemies, the player is presented with an inter-level menu where the score is displayed. The user then presses the 'spacebar' in order to proceed to the next level. If the character dies from having no more health, the user is presented with a menu to either restart the game or to quit.