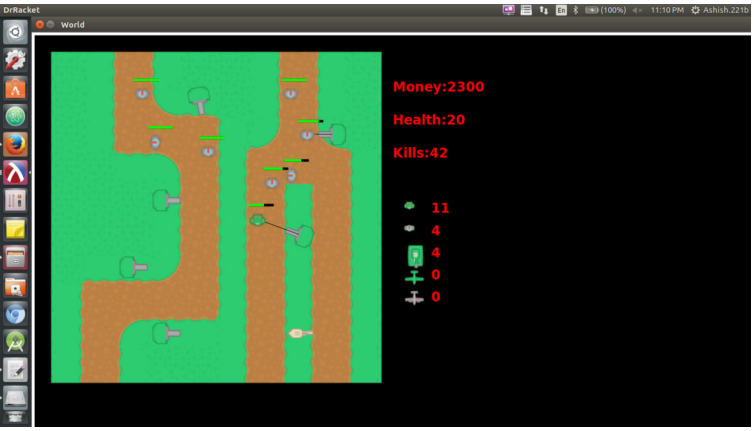
***PROJECT REPORT***

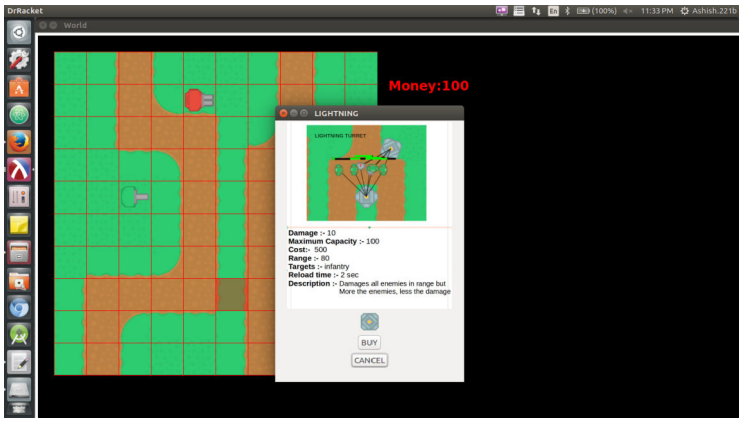
# CS 154 PROJECT :--

*BY ASHISH MITTAL (160050022)*
*AND AMAN JAIN (160050034)*

## 1. Glimpses of the Game :-


1.

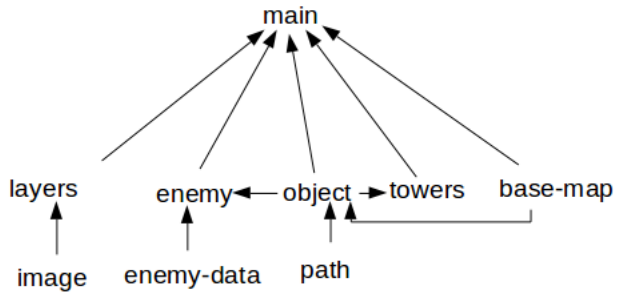




## What is our Project??

Our project is classical tower defense game in which you have a map of the game. A fixed amount of funds are provided to the user at the start of the game. The purpose is to intelligently place the towers on the map so as to stop the enemy troops from crossing the map. There are six types of towers, each with a unique ability of dealing with enemies. The enemy troops consists of varied types of enemies:- land units, tanks, and air-troops. Different enemies have different paths, health and speed.
You have to clear 8 levels to be victorious.

## 2. Structure of the Program :-




Main -: Contains rendering part state variables and User Interactive Gui implementation.
Layers-: Contains all the layers used in the game which are generated by various lists passed by the main.
Images-: Contains all the images which are required in generating layers.
Enemy-:Contains various functions to expand the information stored in Enemy data.It sorts of unzips the information in usable form and also creates enemy unit objects.
Enemy-data-: Contains just lots of list returning functions. Information is in quite concise format.
Object-: Contains the enemy class which have different properties according to "type" initialisation field.It also contains functions to create enemies.
Path-: contains the paths to be followed by enemy units.
Towers-: contains the tow class which once again has "type" dependent properties. Contains projectile class which is class for missiles.
Base-map-: contains map list and map generator functions.

## IMPLEMENTATION OF GAME:-

The Game uses **2htdp library** for graphic rendering and **racket/gui** for user interaction.The lists are the main data structure which stores most data. Example list of active enemies, fired missiles etc. Game is based on repeated calling of a function which calls all the functions of the enemy objects , towers and projectiles etc. It also updates various state variables.So basically the frame to be rendered is a function of **Game state**
which indeed is comprised of various global variables.

## Prime Programming Features:-

**Objects:-** The game uses 3 classes

**Tow**:- For all towers. They behave quite differently according to there type variable.This class contains nemerous data member to give a very realistic look to the turrets

**Enemy**:- for all the enemy units this class has been used.It also shows a kind of polymorphism according to type variable.

**Projectile**:- for the missiles.

**Lists:-** There is an extensive use of lists in our game. Lists are used as major data structure to pass data from one file to other.

**Map:-** map functions has been the major tool used in all processing of global variables which are lists.

**Foldr:-** foldr has been used in genrating all of the layers.

**Higher Order Functions:-** Program frequently make use of higher functions which have time and again reduced code repitition or other programming conveniences.

Few examples of higher order functions are **move-along** function in enemy class, enemy list sorting in tower class and **map-generalised**.

## Other Key features:-

- **Tracking algorithm:-** It has been used for automatically genrating path of missiles by locking it on a target.
- **Enemy list sorting :-** The order in which towers attack enemies is a property of each tower.
- **World State Polymorphic:-** Separate worldstate(frame to be rendered) is a dependent on which state the game is in.This helps in removing unecessary layers and thus optimising the game.

## Limitations & Bugs:-

- Overlapping of Enemy units due to effect of slowing down towers.
- Game proceedings are adversely affected by keyboard key being kept pressed. This very limitation was the reason we were not able to take user input using.The reason of bug is repeated calling of on-key/on-mouse even on mouse drag or move function (though they may be void) which affects game play.

**Workaround for the mouse-input:-** We used racket/gui to take the mouse input because it is executed in a separate frame.


## Aknowledgement:-

- Our Professor ***Prof. Amitabh Sanyal***
- Game images are taken from https://opengameart.org/content/tower-defense-300-tilessprites .