

AETD - Arch-Enemy Tower Defense

Group 6

Olof Ol-Mårs
Erik Nordenhök
Felix Wallén
Johan Gustafson
Jonas Hellgren

5. Design Details

5.5 Detailed Design

Class: AETD

Variables

vector<vector <Button>> menu
int current_frame
Graphics graphic_manager
Game game
InputParser input_manager
int difficulty
String player_name
Socket socket

Functions

void drawMenu()

Calls on the graphics manager to draw the vector<Button> at current_frame in the menu vector. Called by the menu functions.

void mainLoop()

The main loop for AETD. Iterates over the menu currently showing to decide what button that has been pressed.

void showSettingsMenu()

Sets current_frame to settings menu. Called by button.

void showNetworkLobbyMenu()

Sets current_frame to network lobby menu. Called by button.

void showNetworkJoinMenu()

Sets current_frame to network join menu. Called by button.

Class: Game

Variables

vector<Tower> towers
vector<Monster> monsters
vector<Projectile> projectiles
vector<Button> buttons
vector<Player> players

Functions

void mainLoop()

Calls the different update and draw functions in game. Called by AETD.

void moveMonsters()

Iterates over all monsters and calls moveMonster for each monster. Deletes the monster if the dead variable in monster is true. Called by mainLoop.

void updateTowers()

Iterates over all towers and calls attackMonster for each tower. Also calls the upgrade and sell functions for towers that are upgrading or selling. Called by mainLoop.

void updatePlayers()

Checks sockets for new updates. Calls the different set functions in Player when needed. If changes to the local player has happened socket is called. Called by mainLoop.

void draw()

Calls graphic with the interface and all entities it has to draw.

void drawDetail()

Calls graphic with data from the current selected target to draw the detailed information box.

void checkButtons()

Checks if any buttons has been pressed according to the data from input_manager.

Class: Player

Variables

int monsters
int towers
int lives
int gold
String name
int playerID

Functions

Get functions for all variables

Returns the different variables. Called by Game.

Set functions for all variables

Set the different variables. Called by Game.

Class: Entity

Variables

String image_name

Coordinate pos

Functions

Get functions for all variables

Returns the different variables. Called by Game.

Class: Tower

Tower inherits from Entity and every tower-type will have its own subclass inheriting from the tower class that contains specialized methods for the specific tower-type.

Variables

float range
float damage
int rate
int level
int time_to_fire
Monster target
vector<Projectile> projectiles

Functions

bool findMonster()

Iterates over the monsters currently on the playing-field to find any monster in range. If one is found it sets that monster to the target and the function returns true. Called by attackMonster().

void attackMonster()

Calls timeToFire() and if it returns true:Checks that a target is not null and that the target is in range. If not the findMonster() function is called. Else the target is attacked with the damage given by the variables. Called on by game.

void upgrade()

Upgrades the tower to the next level, increasing the different variables as appropriate. Called on by game.

void sell()

Sells the tower. Called on by game.

bool timeToFire()

Decreases time_to_fire with 1. If it reaches 0 it returns true and sets time_to_fire to rate. Otherwise returns false. Called by attackMonster().

void moveProjectiles()

Iterates over all projectiles and calls moveProjectile for each projectile. Called by attackMonster().

Get functions for all variables

Returns the different variables. Called by Game.

Class: Monster

Monster inherits from Entity and every monster-type will have its own subclass inheriting from the monster class that contains specialized methods for the specific monster-type.

Variables

int hp
int display_hp
bool dead
int level
float speed
list<Coordinate> path

Functions

bool calculatePath()

Calculates the shortest path between the current position and the goal. Returns false if there is no path. Called on by the Tower class when a tower is built or sold.

void moveMonster()

Moves the monster towards the next coordinate in the path at the set speed. Called on by game.

void printDetails()

Prints hp, speed and level to the information box, via Graphic.

Class: Button

Variables

`void (*f_pointer)(void)`

Functions

`void buttonFunction()`

Calls the function the button points to. Called by the class were the button is declared.

Class: Projectile

Variables

Monster target

float speed

Functions

void move()

Moves the projectile towards the target. When the projectile hits the target it sets its display_hp to hp, if the hp is less than or equal to zero it sets the dead variable to true. Called by game.

Class: Graphics

Variables

Functions

void printString(String, Coordinate)

Prints a string at the position given by the coordinate. Called by everything that wants to print a string on the screen.

void drawEntity(Entity)

Draws an entity with the image and at the coordinate specified by the entity variables. Called by everything that wants to print an entity on the screen.

void drawSquare(Color, Coordinate, Coordinate)

Draws a colored square between the coordinates specified.

Class: Chat

Variables

vector<String> history

Coordinate pos

Functions

void addInputMessage(String)

Calls Socket and adds the String to history. Called by input manager.

void addNetworkMessage(String)

Adds the String to history. Called by socket.

void printHistory()

Calls graphic with the history variable and the positions given by the coordinate. Called by game or AETD.

Class: Server

Variables

```
queue<String> out_data  
queue<String> in_data  
vector<int> port  
vector<int> ip
```

Functions

void sendTo()

Sends out_data to the clients.

void recvFrom()

Receives data from a client and stores it to in_data.

void socket()

Creates an endpoint at the server.

void bind()

Function for setting up the connection with the clients socket.

Class: Client

Variables

```
queue<String> out_data  
queue<String> in_data  
int port  
int ip
```

Functions

void sendTo()

Sends out_data to the server.

void recvFrom()

Receives data from server and stores it to in_data.

void socket()

Creates an endpoint at the client.

void bind()

Function for setting up the connection with the server socket.

Class: InputParser

Variables

bool left_state

Coordinate pos

char key

Functions

Get functions for all variables

Returns the different variables. Called by Game or AETD.

5.6 Package Diagram

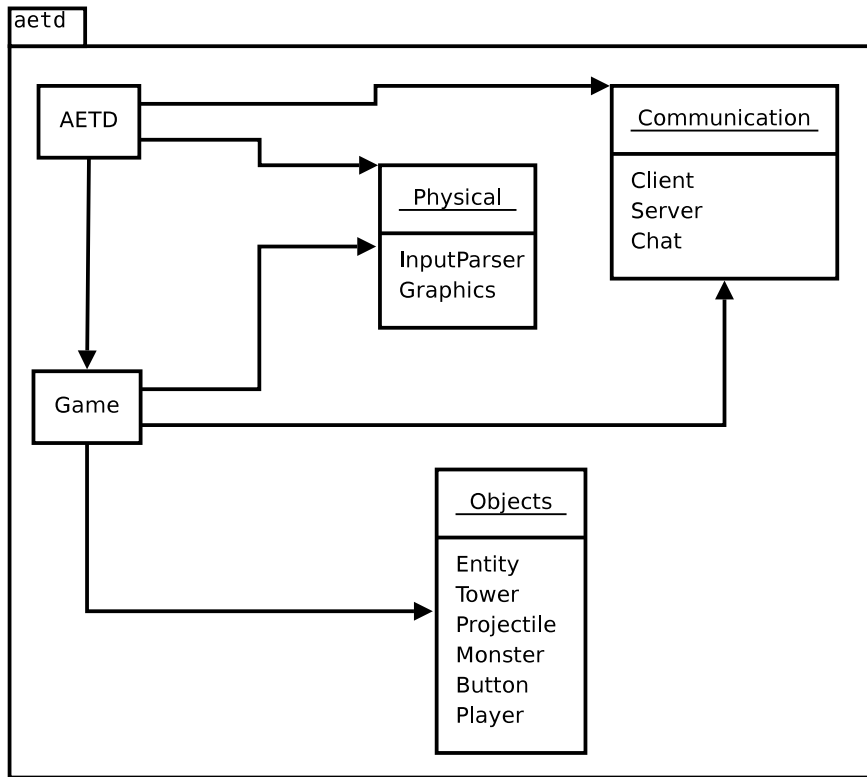


Figure 1: Package Diagram