

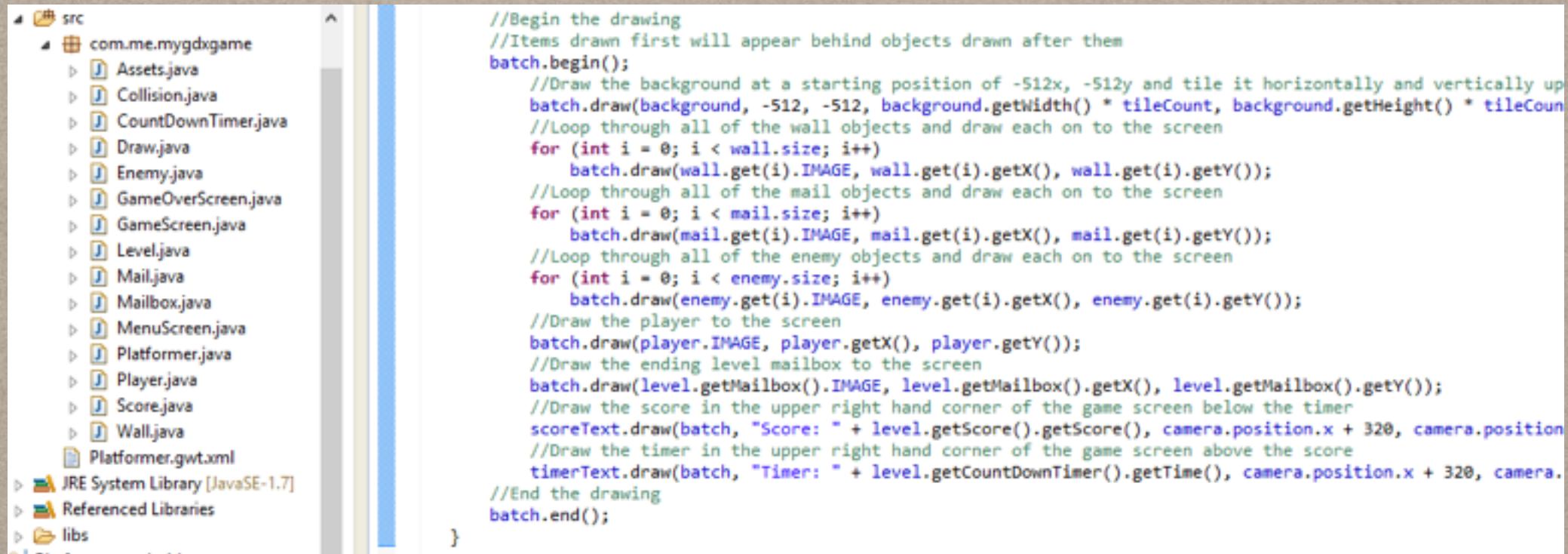
CS 386: TEAM SHADOWFAX

ZOWIE HAUGAARD, LEAH KALUSH, JORDAN
HARRIS, JUSTIN LIDDICOAT, SURIM OH

THE PROJECT: LIBGDX

- A library for making desktop and mobile games in the Java programming language
- Integrates with the Eclipse IDE
- Provides user friendly tools to make complex tasks simple

LIBGDX

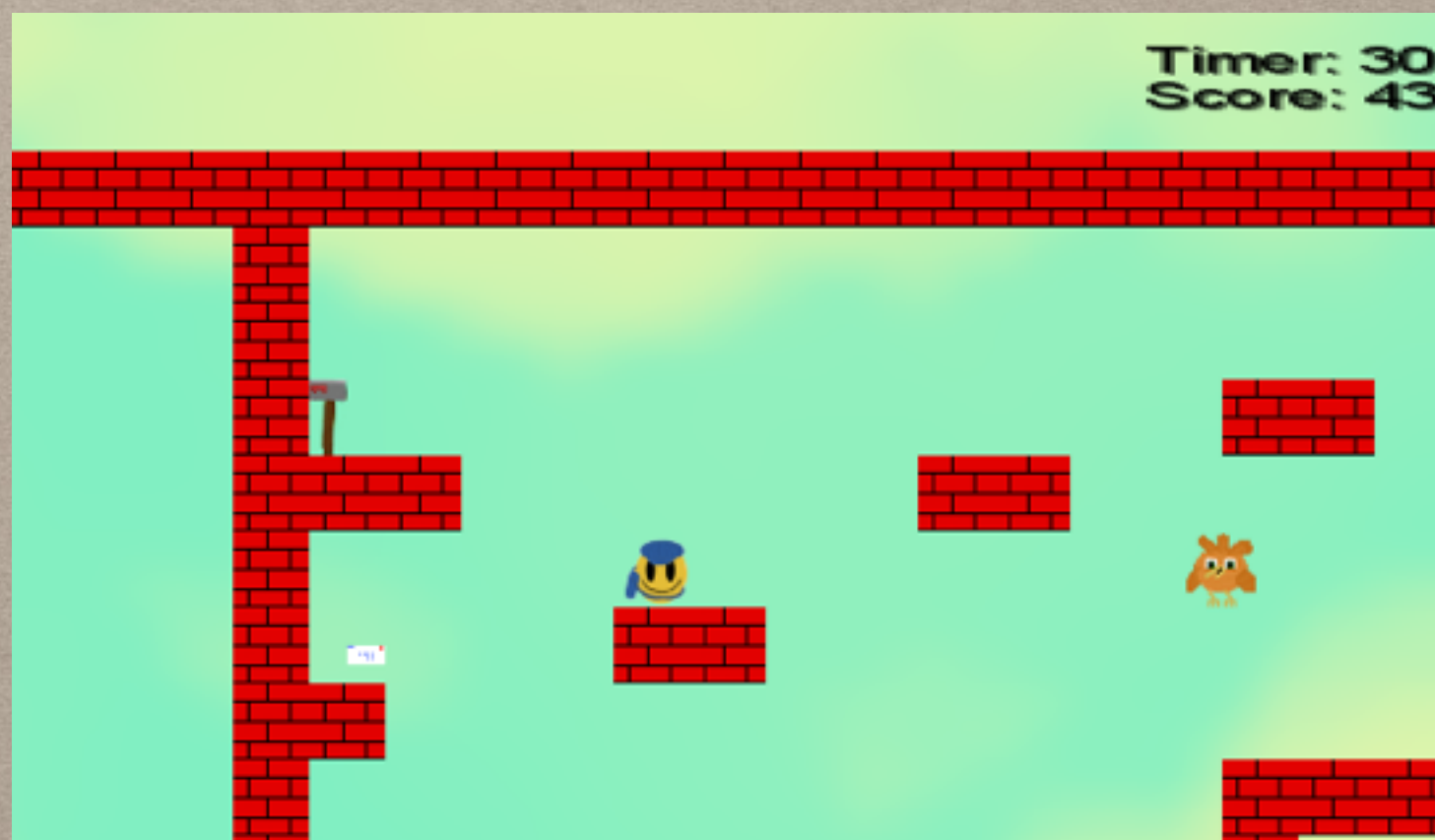
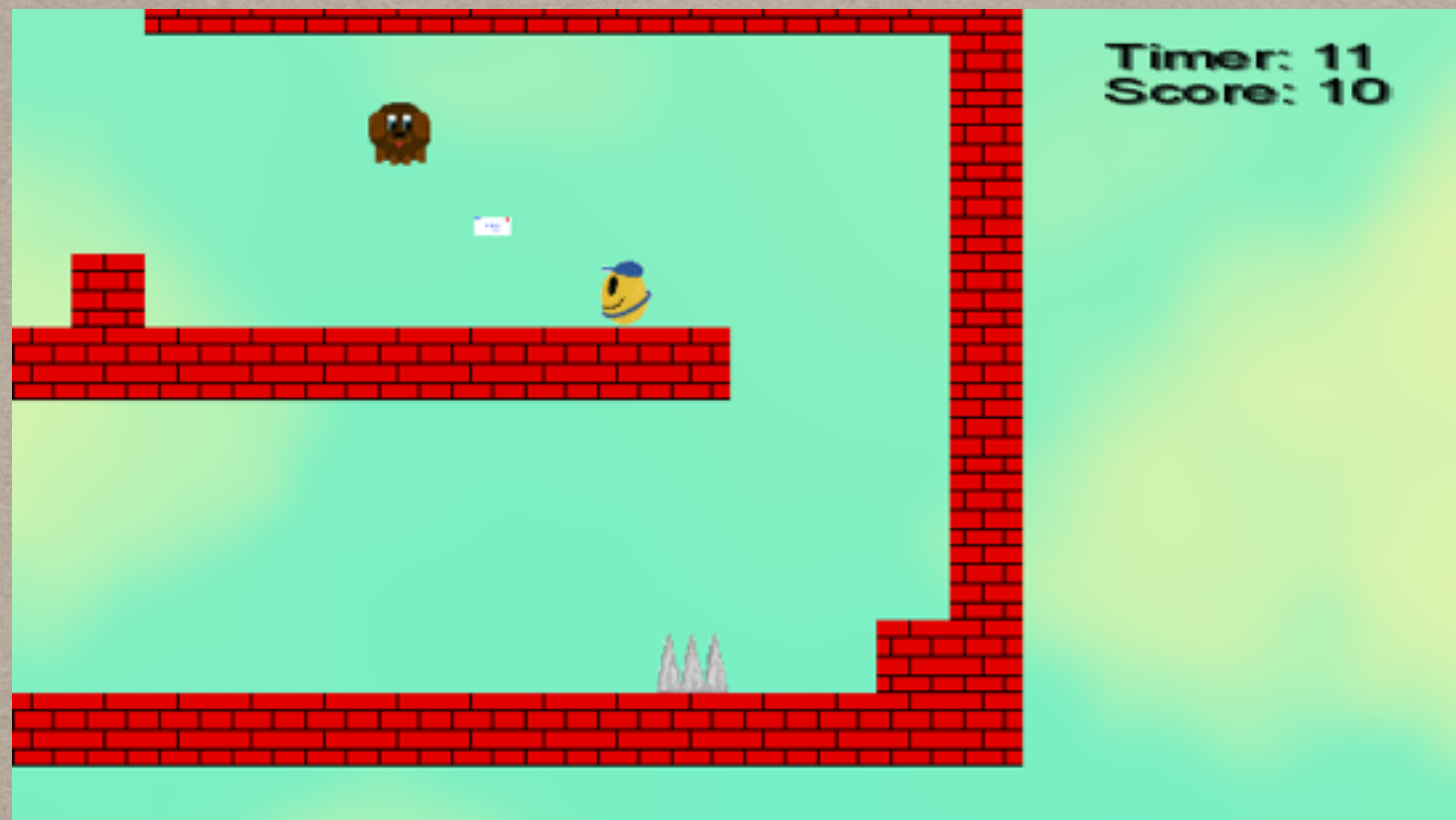


The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure is under the 'src' directory, with a package 'com.me.mygdxgame' containing several Java files: Assets.java, Collision.java, CountdownTimer.java, Draw.java, Enemy.java, GameOverScreen.java, GameScreen.java, Level.java, Mail.java, Mailbox.java, MenuScreen.java, Platformer.java, Player.java, Score.java, and Wall.java. There is also a 'Platformer.gwt.xml' file. Below the project structure are 'JRE System Library [JavaSE-1.7]', 'Referenced Libraries', and 'libs' folders. The code editor displays a Java code snippet for the drawing phase of a game loop. It begins with a comment '//Begin the drawing' and '//Items drawn first will appear behind objects drawn after them'. The code then calls 'batch.begin();'. It draws the background with 'batch.draw(background, -512, -512, background.getWidth() * tileCount, background.getHeight() * tileCount);'. It then loops through wall objects with 'for (int i = 0; i < wall.size; i++)' and draws each with 'batch.draw(wall.get(i).IMAGE, wall.get(i).getX(), wall.get(i).getY());'. Next, it loops through mail objects with 'for (int i = 0; i < mail.size; i++)' and draws each with 'batch.draw(mail.get(i).IMAGE, mail.get(i).getX(), mail.get(i).getY());'. It then loops through enemy objects with 'for (int i = 0; i < enemy.size; i++)' and draws each with 'batch.draw(enemy.get(i).IMAGE, enemy.get(i).getX(), enemy.get(i).getY());'. It draws the player with 'batch.draw(player.IMAGE, player.getX(), player.getY());'. It draws the ending level mailbox with 'batch.draw(level.getMailbox().IMAGE, level.getMailbox().getX(), level.getMailbox().getY());'. It draws the score with 'scoreText.draw(batch, "Score: " + level.getScore().getScore(), camera.position.x + 320, camera.position.y - 100);'. It draws the timer with 'timerText.draw(batch, "Timer: " + level.getCountDownTimer().getTime(), camera.position.x + 320, camera.position.y - 100);'. The code ends with '//End the drawing' and 'batch.end();'.

```
//Begin the drawing
//Items drawn first will appear behind objects drawn after them
batch.begin();
//Draw the background at a starting position of -512x, -512y and tile it horizontally and vertically up
batch.draw(background, -512, -512, background.getWidth() * tileCount, background.getHeight() * tileCount);
//Loop through all of the wall objects and draw each on to the screen
for (int i = 0; i < wall.size; i++)
    batch.draw(wall.get(i).IMAGE, wall.get(i).getX(), wall.get(i).getY());
//Loop through all of the mail objects and draw each on to the screen
for (int i = 0; i < mail.size; i++)
    batch.draw(mail.get(i).IMAGE, mail.get(i).getX(), mail.get(i).getY());
//Loop through all of the enemy objects and draw each on to the screen
for (int i = 0; i < enemy.size; i++)
    batch.draw(enemy.get(i).IMAGE, enemy.get(i).getX(), enemy.get(i).getY());
//Draw the player to the screen
batch.draw(player.IMAGE, player.getX(), player.getY());
//Draw the ending level mailbox to the screen
batch.draw(level.getMailbox().IMAGE, level.getMailbox().getX(), level.getMailbox().getY());
//Draw the score in the upper right hand corner of the game screen below the timer
scoreText.draw(batch, "Score: " + level.getScore().getScore(), camera.position.x + 320, camera.position.y - 100);
//Draw the timer in the upper right hand corner of the game screen above the score
timerText.draw(batch, "Timer: " + level.getCountDownTimer().getTime(), camera.position.x + 320, camera.position.y - 100);
//End the drawing
batch.end();
}
```


CONTRIBUTION

- We were unable on our own to find a suitable enhancement to add to the project's code base
- Decided to ask the development community for suggestions
- Asked to develop a demo
- Built a game using the LibGDX library to act as a demo project for new users



CHALLENGES

- Biggest challenge was finding something to work on
- Spent first two weeks trying to identify a contribution
- Once we figured out what to work on, implementation was easy and straightforward

LESSONS LEARNED

- We learned that walking into a developed project with no previous involvement is incredibly difficult
- Documentation is great for learning a system, but working with it is the only way to really understand it