MTU1 Group Project

Design decisions

We decided to make an old school style D&D type game where a player encounters various monsters and has to beat them to earn Gold. We liked this idea as we thought it could be very easily scaled up or down to fit how much we could do on it.

The first thing we decided on was the basis for the creature class structure and how both the player character and the monsters shared quite a few attributes and functions. This was done by using a parent class BaseCreature class with child classes for players or creatures. We made the player character a class because it would clearly group all their information while also providing the option of adding more characters a player could control later on.

The next design decision was how to structure the game screen to be aesthetic, functional and easy to understand. We chose to have a background image with monsters appearing on it so it was clear what the player was facing. We then added a sort of overlaid hud that was split into three sections;

- The player's Information - which is always visible and the same, which shows the player's health, gold, defence and attack stats.
- The Action section - which provides the player with options they can take as actions. This changes depending on where they are in the sub-menus.
- The recap section - we decided this was important to add so it was clear what was going on in the game and to give the player clear feedback on their actions.

Another decision we made was to not use the jpanel buttons provided as they didn't fit exactly what we wanted. This was important as the buttons are integral to the gameplay for this style of game.

Handling combat between players and creatures was also an important decision to make. We decided the easiest way to do this was have the player choose a single monster to "enter combat with" at a time. While engaged with a monster every time you attacked or healed that monster would attack you back. We decided the simplest way to implement damage, attacks and health was simply to have the damage be equal to the attacker's attack stat minus the defender's defence stat so that if the player attacked with 30 attack verse the monster's 10 defence the monster would take 20 damage to its health.

Finally in all aspects of the game we tried to make things as modular as possible, and with room to add more, so that not only was it easier to work on the code together but also to allow for improvements and growth of the game features as time allowed (or for future updates).

Features of the Game

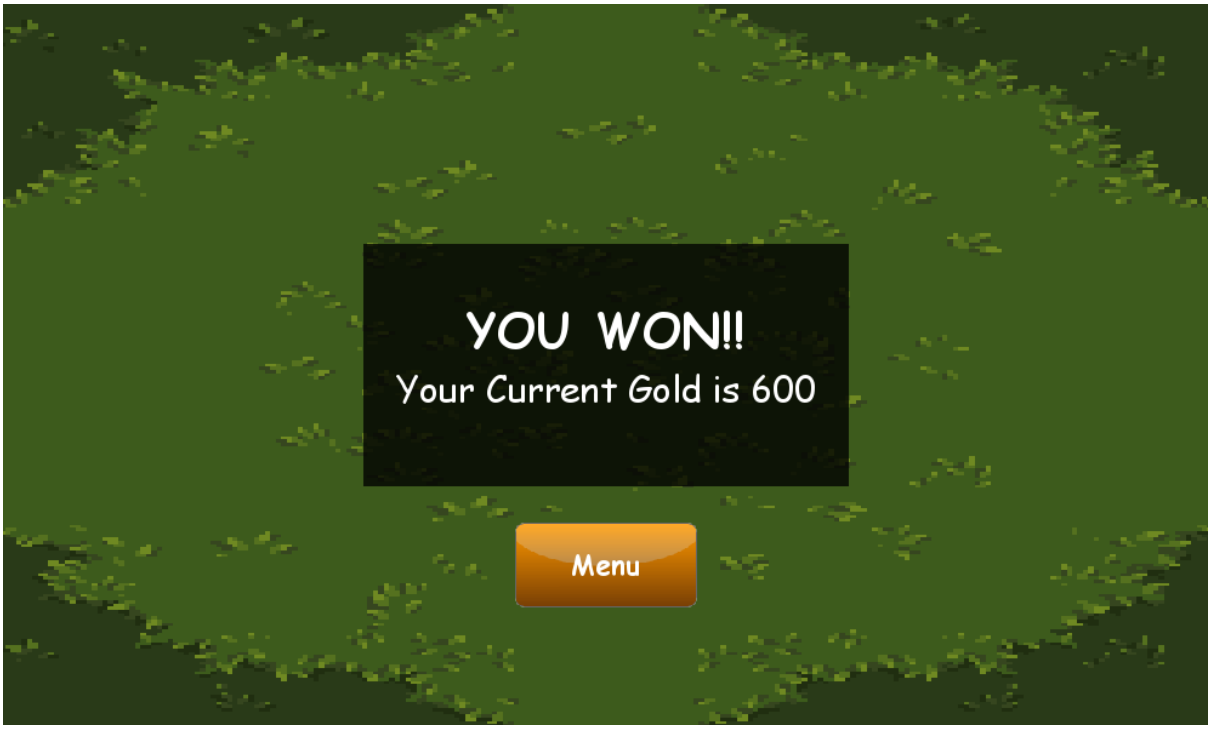
Our game provides the player with a character that has to fight various monsters. If they defeat all the monsters they win the game. However if the character runs out of their own health they die and have lost the game.

The player can select certain monsters to attack, in an attempt to defeat them. Attacking a monster will deal damage to the monster's health equal to the player's attack minus the monster's defence. If the monster is reduced to 0 hit points it will die and be removed from the battle and reward the player with Gold.

Players can also choose some other options like Run to runaway, or Heal to try and heal themselves. They can also choose to use an item. Each item has a different effect.

Screenshots


f
Health: 33.0/100
Attack: 30
Defence: 10
Item 1
Item 2
Item 3
Stink Bomb
This is a Bomb that doesn't hurt but smells really bad
Back
You attacked Knight
You killed Knight
You get 150 gold


YOU WON!!
Your Current Gold is 600
Menu

Contributions

Lachlan

- Backend Functions
- Class Functions
- Background Sounds
- Coding Game Mechanics

Brook

- Presentation
- Images and sprite creation
- General design decisions
- Headers and minor touch ups
- Button sound

Floyd

- Working on play screen & relative buttons
- Item functions
- Code organisation and readability
- Bug catching