Player’s Guide

Intro

In “Castle Siege”, you are a conqueror!
It is your chance, to lay siege on castles, destroy towers, and bring down the enemies’ strongholds.

The game has a set of levels. In each level, you lead an army to conquer the castle. The castle is protected by an array of towers. They will fire at your soldiers.

The amount of units you have available is limited. Think carefully where you place them.

Actions

Actions are simple.

Click on a soldier in the bottom right corner. Move your mouse to a position near the lower end of game screen. Click to summon 10 soldiers of that type.

The soldiers will move forward automatically and attack anything in sight.
They move in formation, as it is part of their combat training. When they start fighting, however, order is forgotten. They cannot think clearly in the heat of battle.

After a couple of levels, you will likely start meeting magical individuals. They will offer you their help and, in doing so, give you access to spells.

Using a spell is similar to summoning a soldier. Spells are located at the bottom left corner. When hoovering with the mouse, you will see the range of the spell. This will let you determine which units or towers will be affected. Casting a spell takes a bit of time, so do not get annoyed at the spell caster.

The Heal spell, cast by the Angel, restore health to all units of soldiers within range.
The Haste spell, cast by the Priest, increases the speed (both movement and attack) to all units of soldiers within range.
The Slow spell, cast by the Warlock, decreases the attack speed and damage of enemy towers within range.
The Reanimate spell, cast by the Necromancer, summons zombies from the invisible corpses of units within range.

You can also press Backspace to restart the level.

Towers

There are two types of defending towers. Ballistas shoot at a target unit quickly. By comparison, Catapults are slow but deal high Area of Effect damage. If you put all your soldiers in front of a catapult, they will be hurt very badly.

Moral

The keeps track of your morality. Low morality means a high chance of meeting evil magical creatures. High morality means a high chance of meeting good magical creatures.

Your morality is affected by how you play the game. Being inconsistent and frequently changing from “good” to “evil” means that you have a low chance for both good and evil friends.

Destroying towers decreases your moral.
Sparing your soldiers and sparing towers increases moral.
Casting spells pushes your moral in the direction of the spell.
Denying offers from magical creatures pushed your moral in the direction opposite of them.
Additionally, at the end of every level, you will be asked to influence the newly conquered castle. Your answer will affect your moral.

Your Game

The game is yours to change.

If you are familiar with Java and think that the game could really use another type of unit (say a knight), you can add it quite easily. Create a Java class and extend “Soldier”. Create some constants and use them to define the abstract methods. Make sure you also provide static fields for moral and health. Those are found using reflection.
You can look at either the Footman class or the Archer class, to see how easy it is to add Soldiers. Make sure you then go to the Player class and tell it about your new class.

You can add Defence towers in a similar way. Ballista and Catapult both inherit “Defence”. Just extend that, implement the two abstract methods, set the static constants, and viola!

If you are wondering how to add Defence towers to the game, well … You can create custom levels. Each level is class that get instantiated when the level starts. Simple extend the abstract class “ALevel”, and then define what resources are available to the user, what defences it has, and what is the question after it. You need to go back to the previous level and tell it that your new level is the next one. You may have to look at how levels 1, 2, 3 and 4 do it.

The most exciting part is, of course, you can add your own spells. Extend the Spell class, define its one abstract method to describe what the spell does. You need to associate the spell with a magical creature. Go to the CreatureManager class, and add a new creature, the interaction with it, and then reference the class of the newly created spell.

Enjoy!