SYSTEM DESIGN INFORMATION

CSC301 : SPRINT 2

CRC CARDS

CLASS NAME: Official Game

PARENT CLASS (IF ANY): StateBasedGame (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Puts all the game states together
- Main class; where we execute the whole game
- Creates the initial display window

COLLABORATORS:

- LoginState
- LoginAttemptState
- RegisterState
- MenuState
- StoreState
- LeaderboardState
- StatsState
- GameState

CLASS NAME: LoginState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays the Login/Register and Exit Game buttons
- When button is pressed, go to appropriate state/view

COLLABORATORS:

- OfficialGame

CLASS NAME: LoginAttemptState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays username and password text field
- Checks if the user has entered the correct username and password
- Goes back to LoginState

COLLABORATORS:

- OfficialGame

CLASS NAME: RegisterState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays username, password, email, DOB, first name, and last name text field
- Checks for valid characters used
- Goes back to LoginState

COLLABORATORS:

- OfficialGame

CLASS NAME: MenuState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Buttons to start single player game, view leaderboards, view personal stats, view store
- Exits the game

COLLABORATORS:

- OfficialGame

CLASS NAME: LeaderboardState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays the top scores in the game
- User able to search for their name in the global leaderboard
- Button to return to MenuState

COLLABORATORS:

- OfficialGame

CLASS NAME: Database

PARENT CLASS (IF ANY): SQL (from SQLConnection lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Have a connection to the online database
- Stores all information needed to login and save score etc.

COLLABORATORS:

- LoginState
- RegisterState
- GameState
- LeaderboardState

CLASS NAME: StatsState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays users personal high score
- Displays other interesting stats in the game (ex. Total zombie kills)
- Button to return to MenuState

COLLABORATORS:

- OfficialGame

CLASS NAME: StoreState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays items users can purchase
- Users can buy retextures here
- Button to return to MenuState

COLLABORATORS:

- OfficialGame

CLASS NAME: GameState

PARENT CLASS (IF ANY): BasicGameState (from Slick2D lib)

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Displays the player, map, HUD
- Player can kill spawned zombies
- Accumulate points over duration of game
- Player buys upgrades/weapons
- Menu to end/quit game

COLLABORATORS:

- OfficialGame

CLASS NAME: Actor

PARENT CLASS (IF ANY): NONE

CLASSNAME SUBCLASSES (IF ANY): Player, Zombie

RESPONSIBILITIES:

- Movement for actor
- Increment/Decrement health for actor

COLLABORATORS:

- Player
- Zombie

CLASS NAME: Player

PARENT CLASS (IF ANY): Actor

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Movement for actor
- Increment/Decrement health for actor
- Keep track of kills
- Keep track of ammo
- Keep track of weapons that player is holding

COLLABORATORS:

- Weapon

CLASS NAME: Zombie

PARENT CLASS (IF ANY): Actor

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Movement for actor
- Increment/Decrement health for actor
- AI for zombies (path finding)

COLLABORATORS:

- NONE

CLASS NAME: Weapon

PARENT CLASS (IF ANY): NONE

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Shoot when player clicks to shoot
- Deal damage to actor if collisions happens

COLLABORATORS:

- Player

CLASS NAME: GameLogic

PARENT CLASS (IF ANY): NONE

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Keep track of all actor's positions on map
- Spawning algorithm for zombies
- A store on the map for player to buy items
- Each level, increase difficulty

COLLABORATORS:

- Player
- Zombie
- Weapon

CLASS NAME: Bullet

PARENT CLASS (IF ANY): NONE

CLASSNAME SUBCLASSES (IF ANY): NONE

RESPONSIBILITIES:

- Display itself on the game screen
- Travel through the screen

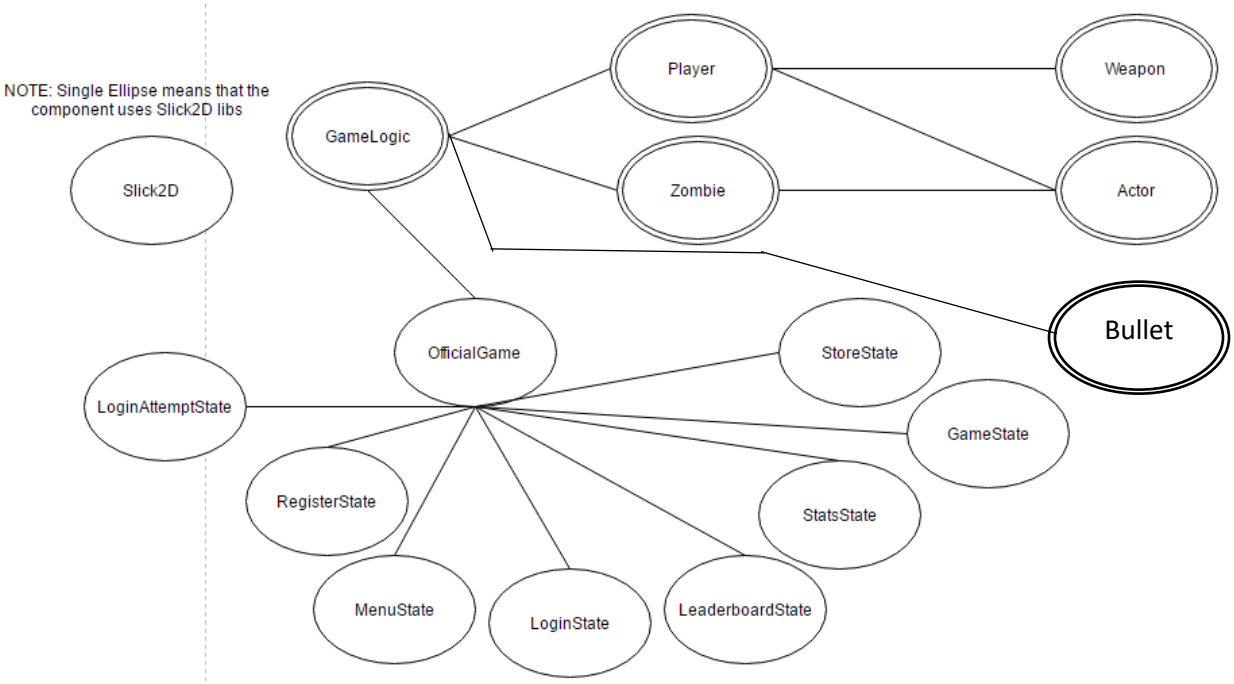
COLLABORATORS:

- Player
- Zombie
- Weapon

SYSTEM INTERACTION

- Dependent upon Slick2D to actually run the game
- Dependent upon MySQL to create a database for the users
- Requires Java 1.8
- Works on all OS that support Java

SYSTEM ARCHITECTURE



SUMMARY OF CRC AND SYSTEM ARCHITECTURE

LoginState: This is the first view the user sees when they launch the game. It connects to the LoginAttemptState and the RegisterState, and switches views to the according state.

LoginAttemptState: This is the view where the user can login to the game if they have an existing username/email and password; the user authenticates by senting a query to our database that we have set up. This connects to the MenuState and LoginState and switches views to the according state.

RegisterState: This is the view where the user can register a username, password, etc. and sends all the information to the database.

Database: This is the class that controls all database functions; for example runs search and insert/update queries as needed.

MenuState: This is the view where the user has logged in and can select where they would like to go. The user may go play a game, view their personal stats, the global leaderboards and store. This connects to the GameState, LeaderboardState, StatsState, StoreState and switches views to the according state.

GameState: This is the view where the user actually plays the game. The user may play the game as intended. The user may quit out of the game or die to submit their score to the database.

LeaderboardState: This is the view where the user may view the top player's scores.

StatsState: This is the view there the user may view their personal stats.

StoreState: This is the state where the user may buy map retextures, player retextures and etc.

Actor: This is an interface where the Player and Zombie uses.

Player: Keeps track of self-values, for example health, armor and ammo.

Zombie: Keeps track of self-values, for example, health, boss Boolean, and pathing range.

Bullet: Keeps track of self-values, draws itself and moves accordingly to the direction given.

OfficialGame: This is the driver of all the states. The user must launch this file in order to run the entire program. This instantiates all states and important values to keep track of through the program's runtime.