Project Team:

Garrett O'Melia

Christian Rebelo

Matthew Tully

Christopher Opitz

* Game Title
  + Hippo Volleyball
* Overview
  + Two players compete against each other to knock the volleyball into the other player's court. Each player controls a hippo, which they maneuver around using the keyboard to hit the ball. When the ball lands, a point is given to one of the players. This continues until one of them wins the game.
* Gameplay Modes: Versus Mode, Practice Mode
  + Setting: Beach volleyball court
  + Perspective: Side view
  + Interaction model
    - General movement: Arrow Keys, WASD
    - Jumping: Lightly tap Up or W for a short hop, press and hold Up or W for a full jump
    - Running and jumping: Press W while holding A or D
  + Challenges: Player vs. Player, each player must hit the ball into the opponent’s court while keeping it out of their own.
  + Actions: Jumping, bouncing off the net, hitting the ball.
* Core Features
  + Movable objects: Ball and players
    - Each of these elements will be subject to gravity as well as collision from other movable objects. Players in particular must respond to commands, and the ball must detect collision with players and calculate a trajectory after being hit.
  + Non-movable objects: Screen boundaries, center net
    - These objects exist as simple geometry that restricts the movement of movable objects. The screen boundaries must keep movable objects from clipping off the screen. The net in the center of the volleyball court dictates a certain height at which the ball must be lobbed, otherwise the ball will bounce off.
  + Physics system
    - The physics system must provide gravity, which acts constantly on movable objects.
  + Scorekeeping system
    - This system must detect when the ball lands, then increment points based on which side of the court the ball landed. Additionally, it must declare a game winner after a certain number of points.
  + Sound effects and music
    - There will be sound effects for jumping, hitting the ball, scoring points, and winning the game. For general gameplay, there will be music.
  + Opening screen
    - A short screen introducing our group and the name of the game.
  + Menu screen and interaction
    - This will be the main screen the user looks at before playing a game. There will be a few choices (Versus, Practice, Options, Exit), and this screen is interacted with using a mouse.
  + Game screen, interaction, and controls
    - This is the screen where the game is actually played. The main form of interaction will use the keyboard. WASD and Arrow keys will control movement, as described above. Pressing the escape key will pause/resume the game. While the game is paused, an option to return to the main menu screen is presented, which is clicked using the mouse.
  + Graphics
    - Graphics will be drawn for the players, ball, net, opening and menu screen, and game screen background.
* Internal Economy
  + Points, from scoring on the opponent’s court.
* Game Balance
  + Positive Feedback
    - Scoring on the other player rewards you with a point. You will become the server for the next round.
    - Reaching a certain number of points (could be anywhere from 7 to 15; undecided) while maintaining a 2-point lead rewards you with a game victory.
  + Negative Feedback
    - Letting the ball fall in your court gives your opponent a point. Your opponent will become the server for the next round.
    - If the opponent reaches a certain point threshold along with a lead (see above), you are defeated.
  + Adjusting the game's difficulty (possible menu setting):
    - Change size of the ball to make hitting it easier/harder
    - Change restitution of the ball to make it bounce more/less
    - Change density of the player to make movement faster/slower
    - Etc.
* Victory Conditions
  + How do you win the game: By reaching the point threshold while holding a lead.
  + How do you lose the game: By allowing the opponent to do so first.
* Project Organization
  + Coding
    - Opening/Menu/Game Screens - *Christian Rebelo, Christopher Opitz*
    - Player Controls - *Garrett O’Melia*
    - Physics: Gravity, Collision Detection and Resolution - *Garrett O’Melia*
    - Scorekeeping - *Garrett O’Melia*
    - Networking (unsure if this will be completed) - *Christopher Opitz*
  + Assets
    - Sound and Music - *Matthew Tully*
    - Graphics - *Christian Rebelo*
  + Documentation - *Garrett O’Melia, Christian Rebelo*
    - Instructions/Tooltips
  + Testing - *Christian Rebelo, Christopher Opitz*
  + Development Environment
    - IDE: IntelliJ IDEA
    - Graphics: Photoshop or GIMP
    - Sound effects and music: Audacity
    - Revision control: Github