Apocalypse Defense

CSCI 576 Project 3

Pat Kujawa

Iterative Design

Feedback

The following is a list of the comments we received from project two and how we chose to address each one.

1. Rethink your use of hardware buttons; some phones don't have them.

   This is true, but we are not targeting those phones/tablets, and android provides an emulated screen for those buttons should we choose to re-target.

2. If we don't have many options on the main screen, why do we make users swipe?

   Good point. **Changed to all three buttons visible at once**.
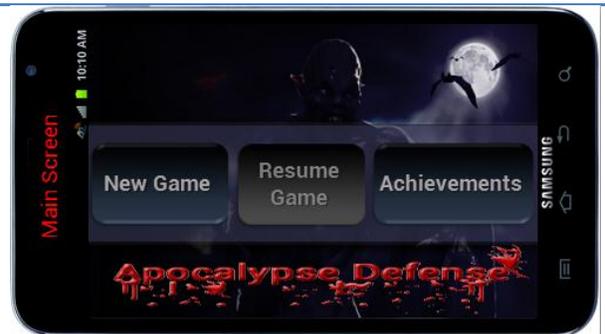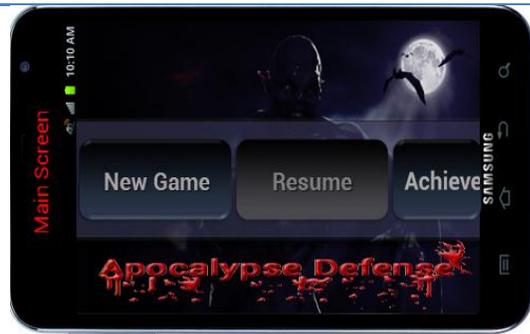
3. On the game play screen, it isn't obvious how to upgrade towers?

   True. **Added text to tutorial saying, "Tap a tower on the map to show upgrade/sell menu".**

4. Maybe you don't want to automatically save when the user exits the game.

   We disagree. It is more usable to have the game autosave and then let the user start a new

game to overwrite the old one. But **a confirmation screen before overwriting has been added.**
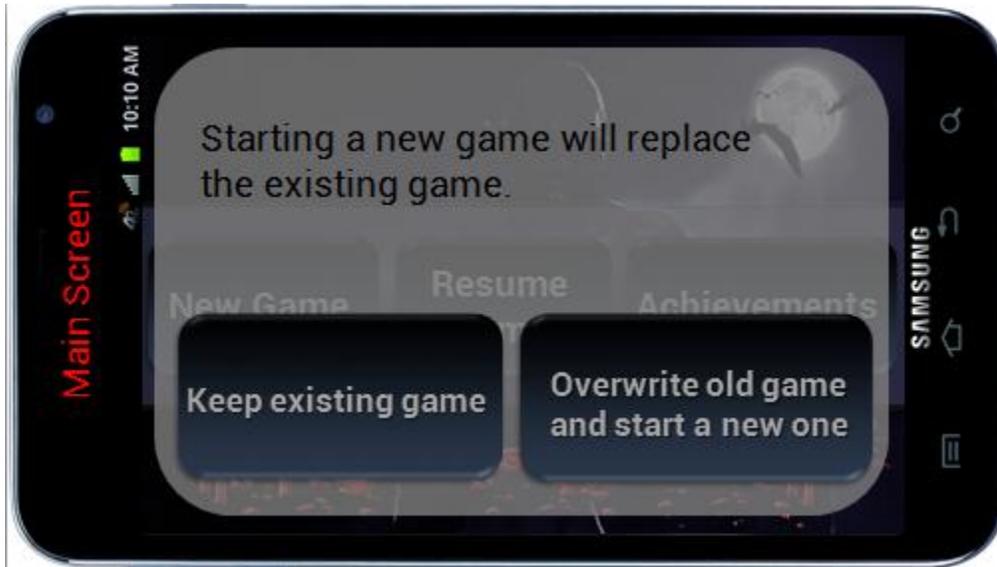
5. Why is the "continue playing game" button larger than the other buttons?

The team is not unanimous on this point. Pat argues that the Microsoft Ribbon is a good UI example and stresses different-sized buttons for opposite tasks. However, it may be better for uniformity and to not appear to "force" the user to continue the game. **Changed buttons to same size**.

6. If you hit a new game button, won't it overwrite the saved game?

Yes. See #4.

7. The cost of the tower is not displayed.

Whoops. **Put a $10 overlay on the tower icon in the gutter area.**

8. The sell button being green makes it seem like a "positive" choice, when in reality it is probably something that the user won't want to do very often.

Agree. **Made $ text red to indicate that it's not necessarily what you want to do.**

9. Make the bottom of the game screen (gutter) as small as possible.

Agree. Yeah, the gutter should be as small as is reasonable so that the map takes up most of the space.

10. Think about upgrade menu and how to make it better.

This isn't specific so we'll just see if usability testing tells us to change the upgrade mechanism.

Cognitive Walkthrough

Effect: Will users be trying to produce whatever effect the action has?

Visible: Will users see the control for the action?

Recognition: Once users find the control, will they recognize that it produces the effect they want?

Feedback: After the action is taken, will users understand the feedback they get so they can go on to the next action with confidence?

Users
- Alice
  - Android expert
- Bob
  - Android novice
- Charlie
  - Tower defense lover; moderate android skills

Assume application has been opened from app icon and **no previous games have been saved**.

Start playing a new game with the default map on Easy difficulty

1. Tap "New Game" button
   i. Effect: Yes, they want to start a new game
   ii. Visible: Yes, the button is large
   iii. Recognition: Yes, clearly labeled and affords tapping
   iv. Feedback: Yes, screen changes when tapped
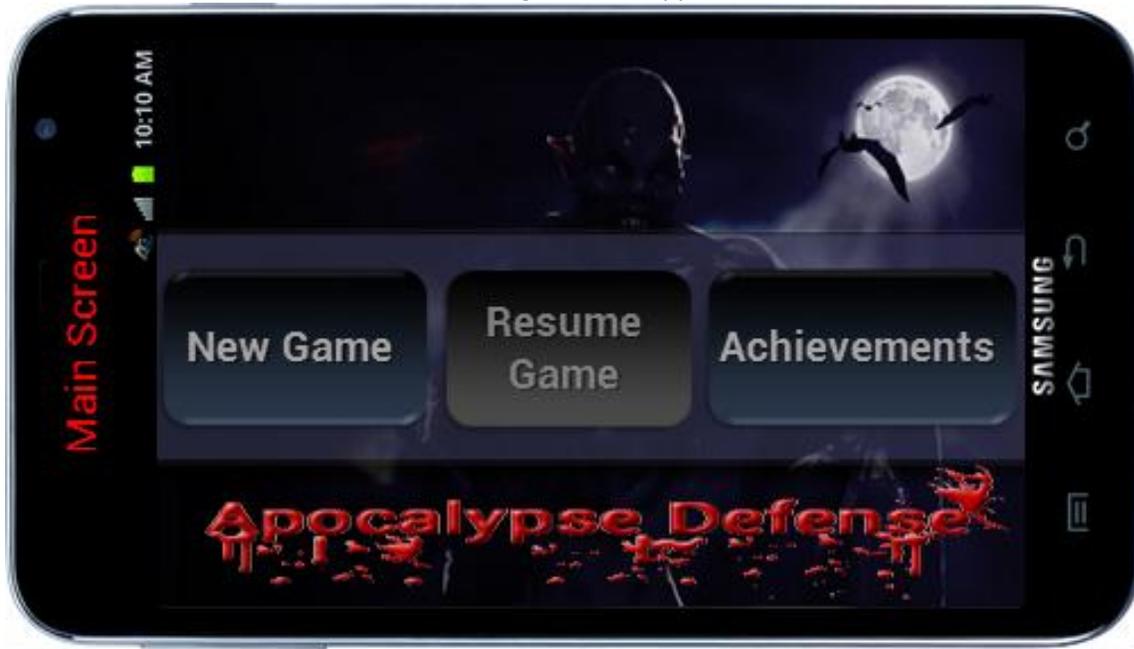
2. Tap "Easy" radiobutton if not already selected
   i. Effect: Yes, they want to set easy difficulty
   ii. Visible: Yes, but small and at the bottom
   iii. Recognition: **Maybe**; not labeled "Difficulty", but implied
      1. Maybe label the difficulty radiobuttons. Defer until usability testing confirms.
   iv. Feedback: Yes, radiobutton blue dot moves

3. Tap map

         i.   Effect: No, but label says what to do

        ii.   Visible: Yes, big image of a map

      iii.   Recognition: **Maybe**; relies on user reading instructions

             1.   Maybe add button overlaid on map saying "Begin" to afford tapping the map. Defer until usability testing confirms.

      iv.   Feedback: Yes, screen changes

4.   Drag tower to map and release when color halo is blue

         i.   Effect: No, but overlay instructions say so

        ii.   Visible: Yes, prominent icon

      iii.   Recognition: Yes, instructions say what to do and use arrows to identify

      iv.   Feedback: Yes, overlay will disappear

Resume a saved game

Precondition: A new game was in progress when application exited previously.

1. Tap "Resume Game" button
   i. Effect: Yes, they want to resume a game
   ii. Visible: Yes, obvious icon location
   iii. Recognition: Yes, obvious labeling
   iv. Feedback: Yes, screen will change and tower will be on map

View achievements

1. Tap "Achievements" button
   i. Effect: Yes, they want to view achievements
   ii. Visible: Yes, obvious button location
   iii. Recognition: Yes, label says "Achievements"
   iv. Feedback: Yes, screen will change
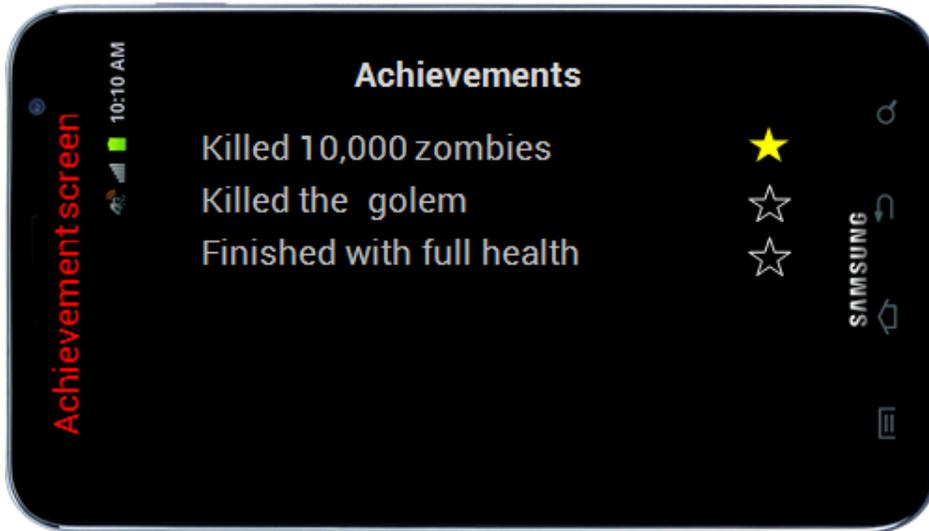
Mute sounds

1. Tap android options button
    i. Effect: Yes, users are familiar with android button opening options
    ii. Visible: Yes, button is standard and always visible
    iii. Recognition: Yes, users are familiar with options button
    iv. Feedback: Yes, screen changes
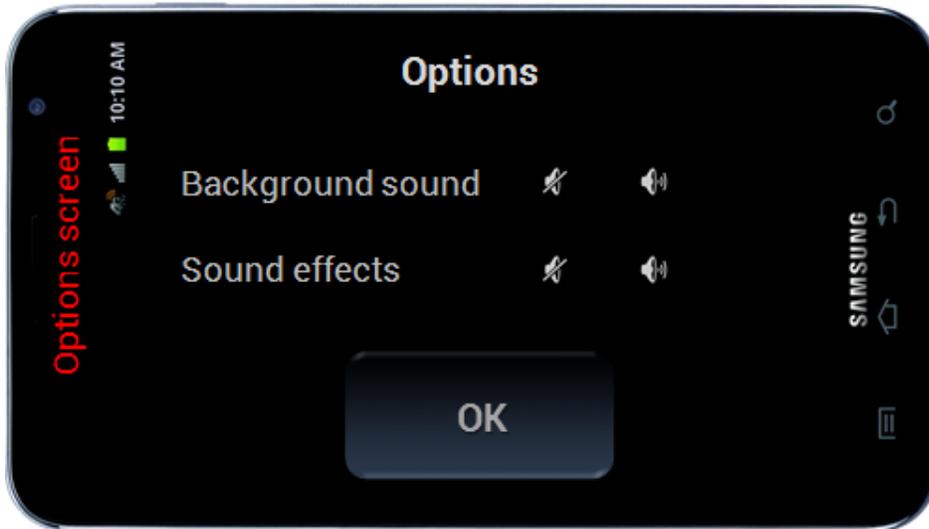
2. If "Background sound" button shows sound waves, tap it
    i. Effect: Yes, user wants to disable sound
    ii. Visible: Yes, one of few icons displayed
    iii. Recognition: Yes, conventional icon
    iv. Feedback: Yes, icon changes
3. If "Sound effects" button shows sound waves, tap it (same analysis as above)

Place tower on map

1. Drag survivor to map and release when color halo is blue
   i. Effect: Yes, the user wants to place a tower on the map
   ii. Visible: Yes, obvious icon location
   iii. Recognition: Yes, label says "Towers"
   iv. Feedback: Yes, tower will be on map and halo will disappear

Upgrade tower

1. Tap desired tower (which is already placed on map)
    i. Effect: Yes, user read the tutorial overlay and wants to upgrade
    ii. Visible: Yes, all towers on map are visible
    iii. Recognition: Yes, tutorial said to tap tower to show upgrade/sell menu. Also follows convention for Tower Defense games.
    iv. Feedback: Yes, tower graphic glows blue
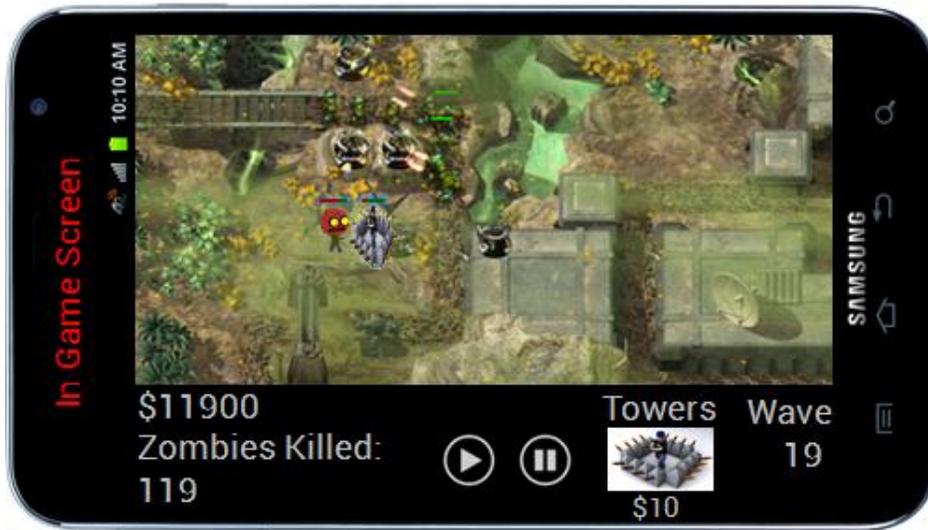
2. Tap button with vertical chevron image ("hat")
    i. Effect: Yes, user wants to upgrade
    ii. Visible: Yes, one of only two buttons. **But what if tower is at top of screen - where will menu be?**
        1. Implement so that menu shows up above tower unless it would go off-screen.
    iii. Recognition: **Maybe**. The other button is a $, so the user might think, "I need to buy an upgrade" and tap that one.
        1. Maybe add text labels saying "Sell" and "Upgrade." Defer until usability testing confirms.
    iv. Feedback: Yes, tower graphic changes; upgrade menu disappears; cash decrements
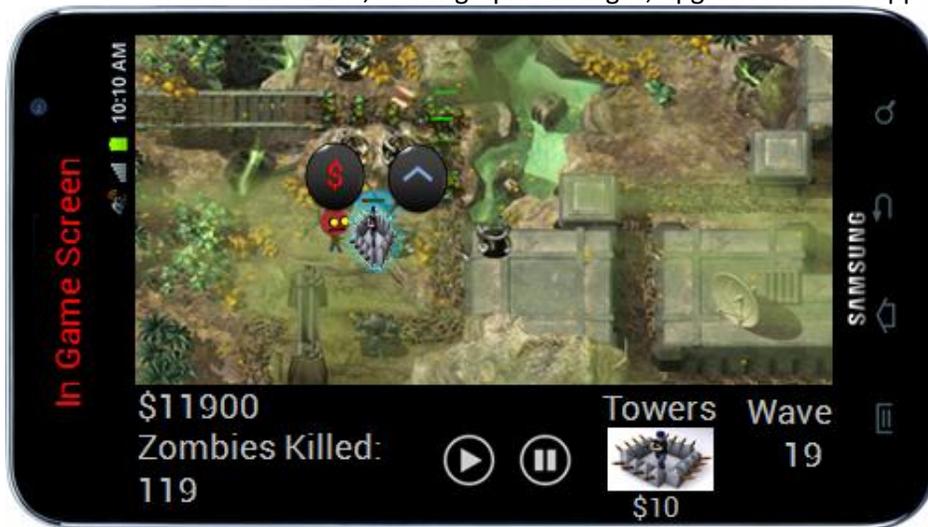
Exit game

1. Tap android home button
    i. Effect: Yes, user is familiar with android buttons
    ii. Visible: Yes, always displayed
    iii. Recognition: Yes, see Effect
    iv. Feedback: Yes, android home screen will appear
    OR
1. Tap android back button
    i. Effect: Yes, user wants to go "back" to android
    ii. Visible: Yes, hardware button is always displayed
    iii. Recognition: **Maybe**. The user might think that the options button would provide the ability to exit rather than the back button.
        1. Maybe add "Exit" button in options screen if usability testing confirms.
    iv. Feedback: Yes, confirmation screen will appear
2. Tap "Exit Game" button
    i. Effect: Yes, user wants to exit
    ii. Visible: Yes, one of few buttons
    iii. Recognition: Yes, labeled "Exit Game"
    iv. Feedback: Yes, android home screen will appear

Changes from cognitive walkthrough

No changes made from cognitive walkthrough – all decisions deferred until confirmation from usability testing with users.

Heuristic Evaluation

Below are my findings on analysis of the entire application, with team responses in the boxes prioritized (0-4) such that:

0. Don't agree
1. Cosmetic
2. Minor
3. Major
4. Must-fix

1. Simple and natural dialogue
   a. Well, when I click to start a new game and there is an existing game, it tells me that, "Starting a new game will replace the existing game" and gives me two options: "Keep existing game" or "Overwrite old game and start a new one." These choices make sense and feel natural.
2. Speak user's language
   a. Examples of sentences include "Starting a new game will replace the existing game" and "Swipe to change maps. Tap map to begin." These statements are simple enough for a child to understand and relate directly to the application.
3. Minimize user's memory load
   a. The only stored items seem to be Achievements. Maybe they could **include a leaderboard with the stats that I get after each win (on the scoreboard screen)**.
   b. When I sell a tower, do I need to remember how much it sells for? **Maybe the sell icon should include a price.**
   c. Also, do I need to **remember how upgraded my towers are**, or will it be obvious?
   d. The instruction overlay says that I need to survive 100 waves to win - **maybe that can be included in the wave count label (e.g. Wave 19/100)**

   > Team response:
   >
   > a. 2 – Good idea. Add to version 2.
   > **b. 3 – Yes, sell should include price**
   > c. 0 – Will be obvious by graphic
   > **d. 1 – Yes, wave label should have this**

4. Consistency
   a. Dialogs and buttons seem consistent. Usage of android buttons is consistent with other apps I've used. The style of the game is fairly different from other tower defense games I've played, but I guess it's not called "Apocalypse Tower Defense."
   b. **In-game, the label says "Towers," but in the scoreboard screen it refers to "Survivors defeated" - are they towers or survivors?**

   > **b. 1 - Should all be labeled towers**

5. Feedback
   a. Every button seems to change the screen in some way, so feedback seems sufficient.
6. Clearly marked exits
   a. Well, the android home button is always present and always brings me back to the home screen.
7. Shortcuts

      a. Again, the android home button returns me immediately to the home screen without having to say, "Yes, I want my game saved."

8. Good error messages
      a. As in *dialogue*, we see messages like, "Starting a new game will replace the existing game" and, "Are you sure you want to exit? I'll save your game so you can resume it later." These are good messages!

9. Prevent errors
      a. What happens if I drag a tower and it gets placed where I didn't want it? Can I undo, or do I have to sell at a loss to get the space back?

      b. What if I select the wrong tower? Do I need to unselect to re-select the correct one? Maybe this should be under *shortcuts*.

      a. 0 – You'll just have to sell back the tower.

      b. 2 – Good idea. Not UI-related, though.

Changes from heuristic evaluation

1. Sell and upgrade buttons should include price
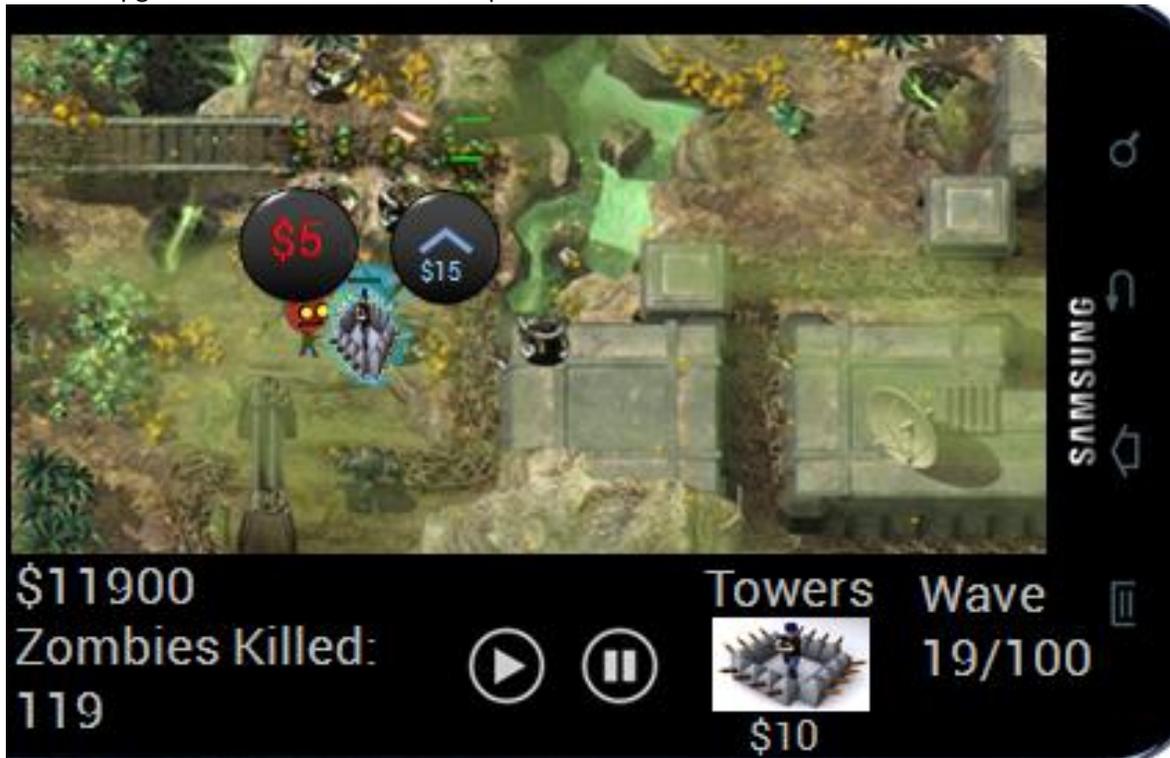
2. Wave label should indicate total number of waves (same screenshot)
3. Scoreboard screen should refer to towers