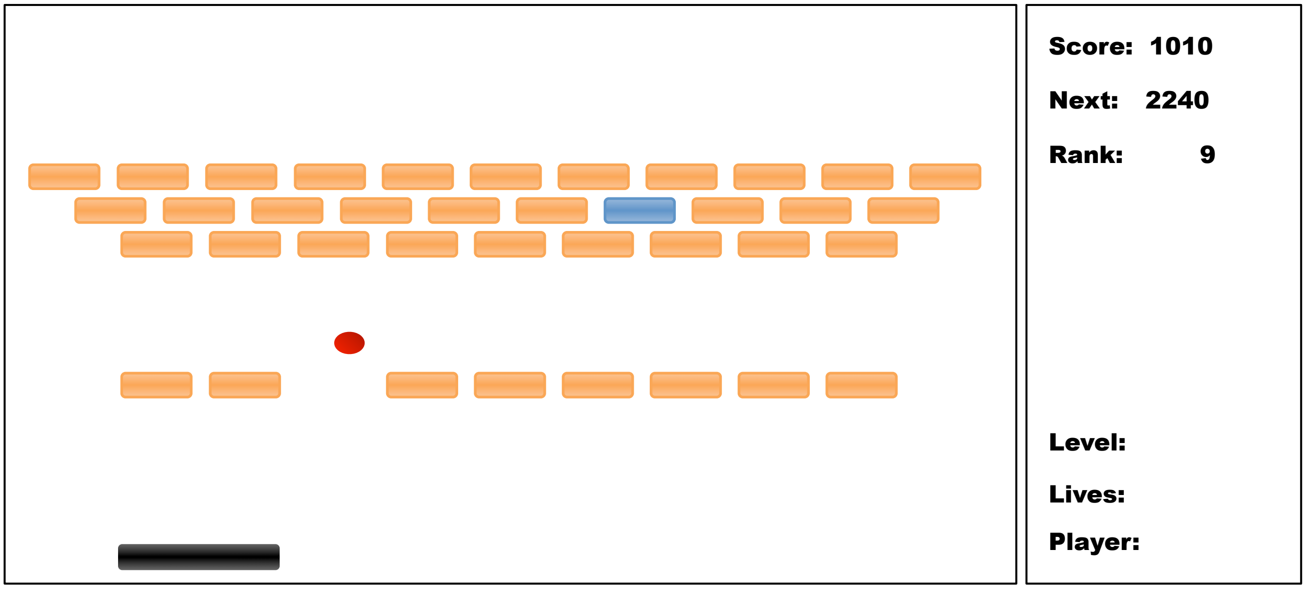
A Cooperative Breakout Game

In this project you will develop a breakout game for Games Unlimited. The goal of the game is to remove all bricks with a ball that is hit by the player. A brick is removed once it has been hit by the player's ball. The ball bounces off the removed brick. The player's ball is controlled by a paddle. The ball bounces from all edges but the lower edge. If the ball hits the lower edge of the playing area, a life is lost. A player has up to three lives. If all bricks are removed from the player's screen, the game advances to the next level. If the player has no more lives left, the game is over. If the high score is in the current top 10, the game asks the player, if an entry to the high score list is desired. After each played game, the player should see the welcome screen (detailed below). A typical game screen could look like this:



The game shows in the upper right the current score, the next score the player has to beat to move up one place in the high score list, and the current rank in the high score list (based on the current score). If the player is currently not in the top 10, the rank field would just show “-“. In the lower right, the game displays the current game level, the remaining lives, and the initials of the player.

The program has to be able to open game levels from a server and to store (and open) them locally on persistent storage. In addition, the program has to be able to read a high score file from a server and update it, if the player's

high score is in the top 10 and the player requests an entry in the high score list.

When the player starts a game a welcome screen lets the player select whether he or she wants to start a new game, download a new level from the server via a http connection, open a local level, access the game rules in form of a help screen, see the current (top 10) high scores, or end the game. If the user has to wait, for example, while downloading a level, the program needs to show a progress bar. Similarly, in case of exceptional circumstances, such as an unavailable connection to the server or a game level that cannot be properly initiated (due to a mistake in the level design), the program should provide appropriate feedback to the player.

Further tasks

Some bricks have extra bonuses: a brick could split the ball into two or three balls, reduce or enlarge the player's paddle, or provide an extra life.

Each group has to design at least two game levels. A game level provides a description of the layout for the bricks and requires a way to store bonus bricks (see above). The game level file can only use relative measures to facilitate portability. Each brick is specified in relative sizes, i.e., x % of the screen width and y % of the screen height. The game level file has to include the speed, direction, and initial position of the ball, and the size of the paddle. You can assume that the paddle is at rest and horizontally centered. Once several teams have implemented their version of the game file, we will take the best one as a reference and all other programs should be able read in this reference game file.

If the ball hits a paddle at rest it just reverses the direction of the ball in x- and y-direction. However, if the player moves the paddle, the horizontal speed of the paddle is added to the horizontal component of the ball's speed.

Extra challenge

Up to now the game can only be played by a single player. For extra marks you can try to design a collaborative version of this game where another player's paddle is located at the opposite of the screen. The game updates between the two players are performed via a Bluetooth connection. This task is difficult since it requires a permanent synchronization between the two

players. If you like to take the collaborative game a step further, even 4 players could be connected for one game, i.e., at each side of the game is a paddle of each participating player. The team loses the game until either one of the players has no live left or, if there is a common number of lives for all players, no live for the team is left.

Points to Check

- **All errors are treated correctly (network, file system)**
- **Successful download of game level files**
- **Successfully load, update, and store the high score from/on the server**
- **Locally load game level files**
- **Design at least two game level files**
- **Being able to read the reference format**
- **Well designed game screen**
- **Game mechanics OK (brick removal, bonus bricks, bouncing behavior)**
- **Collision detection OK**
- **All updates OK (current score, lives, advance to next level, ...)**
- **Nice welcome screen (for example a picture as background) including help screen**
- **Game works on different resolutions (also in portrait & landscape mode)**
- **Appropriate feedback (progress bar, successful download of files, ...)**
- **Game fluid and plays well**