**BABY CRAWL**

**CIS 487 2D Game Pitch (Final)**

Alfred Kishek

Constantin-Antonio Costea

**Executive Summary**

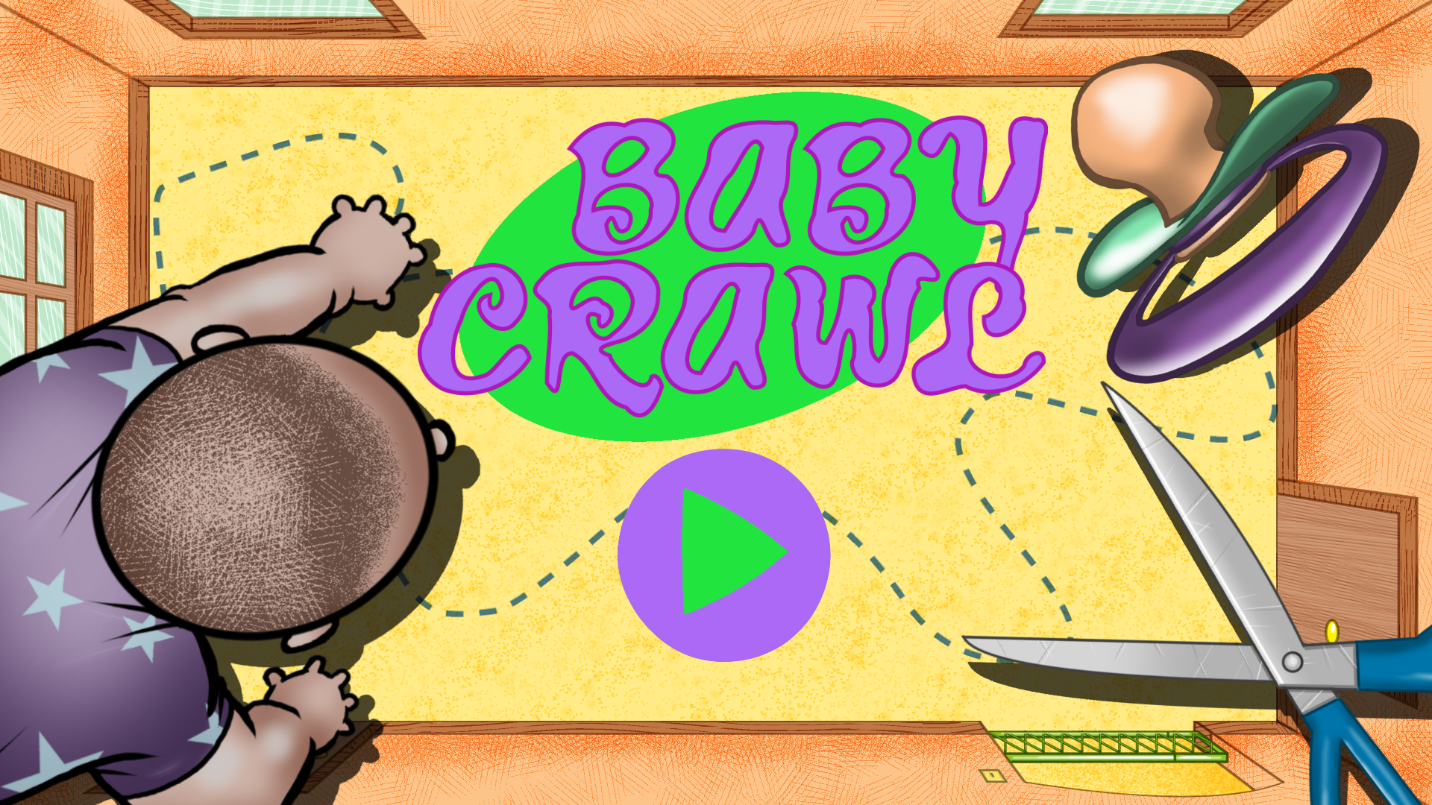
*Abstract of game story*

Modern smart phones allow homeowners to control practically any electronic devices found inside their houses. An absent-minded parent and a busy baby find themselves fortunate to live in just such a smart home. Parents often have chores to do in rooms away from their babies, sometimes even outside the house. Now that the busy baby has learned how to turn the locks on door handles, the parent (or other caretaker) has too often been found locked out of a room or even the house. Luckily, these caretakers always carry their smartphones wherever they go. With a webcam in every room and access to all electronics, a caretaker can keep a watchful eye on the baby all while activating various electronics to either distract, entertain, or soothe the baby and otherwise keep it away from dangerous items left lying around. The parent or babysitter only has to keep the baby safe long enough for the other parent to arrive and unlock the door.

**Game Play Look and Feel**

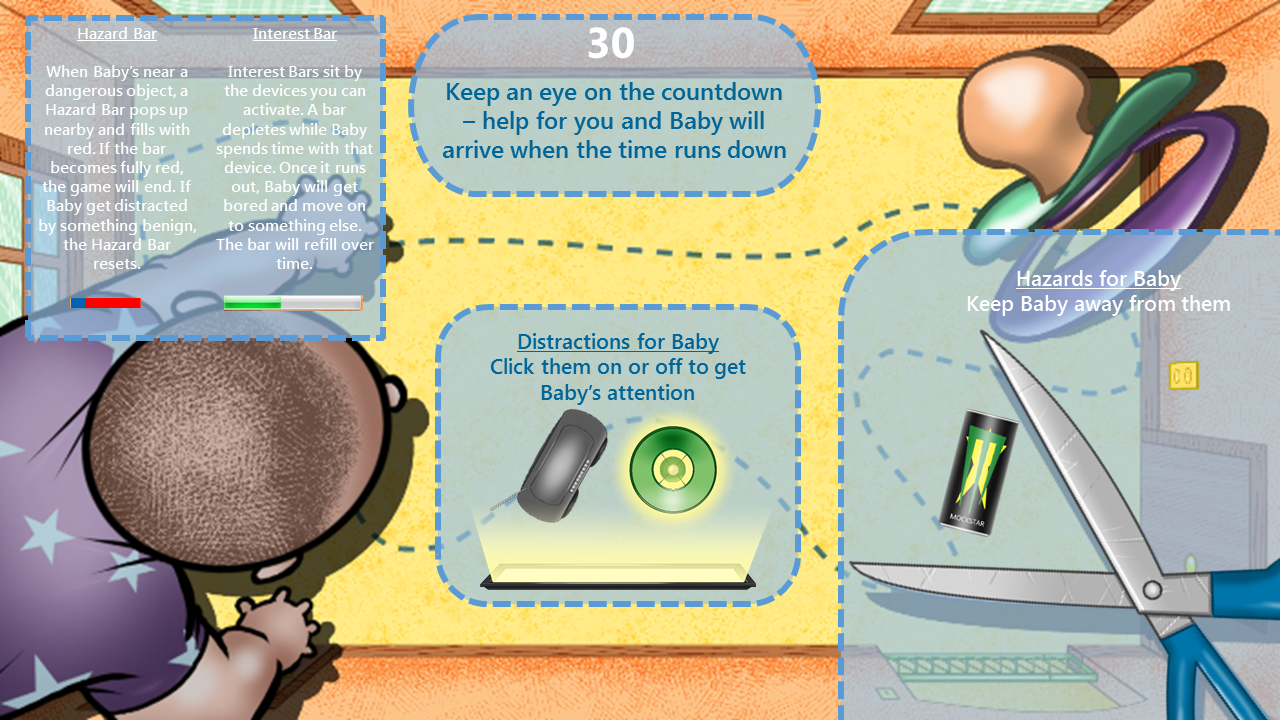
*Appearance*

The player’s perspective is a top-down view of the room the baby’s currently occupying, as if through a ceiling-mounted webcam streamed to the caretaker’s smartphone. The aesthetic is cartoony and simple. Various items and furniture one commonly finds in homes can provide differing interactions for the baby character. Relevant items are highlighted for the player in various ways. Items which are appealing to the baby and can be activated by the player are pointed out by floating hands when the level loads. Dangerous items glow red when the baby gets too close to them. The HUD displays the amount of time left until the other parent or caretaker arrives to unlock the door, relevant progress bars, as well as a reset button. Progress bars are of two types: Interest Bars that show the baby’s dwindling interest in an item and Hazard Bars that fill with red until the baby gets successfully distracted or the game ends.



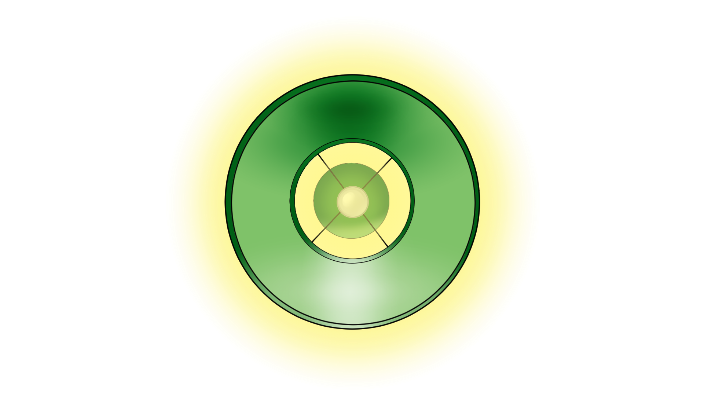
*Start screen*

An opening screen in the same cartoony style as the game takes the player through a couple of introductory pages that present the player with the premise and basic idea behind the gameplay. An additional help page can be accessed from these pages, as well. Game-ending situations such as level completion or failure to keep the baby safe currently only result in a bold message of congratulations or encouragement to try again, respectively.



*Help page*

Animation is limited to the baby sprite’s movement, particle effects, a countdown clock, and progress bars. Activation or deactivation of some elements changes their appearance or releases particle effects to alert the player to their status.



*Lamp object in off and on states, sans particle effects*

*Player roles and actions*

The player’s responsibility is to manipulate the baby’s environment to keep it away from dangers and keep it happy until the timer runs down. By clicking or touching (depending on the interface) actionable items, the player influences the baby’s behavior without actually being able to control the game’s protagonist.

*Strategies and motivations*

The player is motivated as any caretaker would be to keep the baby safe. How a player decides to go about this is open-ended. The baby’s behavior is random and it will wander wherever it pleases. As such, every play-through will be unique depending on what the player chooses to do.

In extreme cases, a player may choose to activate all devices at once. This may confuse the baby and lead to unexpected outcomes, though most likely an unsuccessful ending. The longer devices are on the less interested the baby will be in them. In this extreme case, the baby may have no benign items of interest left to go to and will instead only turn to game-ending hazardous objects.

A more successful strategy would be to activate a benign object to lure the baby in only to deactivate it as soon as the baby is out of the range of a dangerous item. As such, the Interest Bar depletes very little and the device may be turned back on as soon as the baby starts to pull away.

*Level summary/story progression*

Future levels would get more difficult as the player advances through each timed stage. At this time, the game features a single level to demo. To increase difficulty, the number of dangers would rise while the number of appealing items would decrease. Spheres of influence (which are not strictly visible to the player, but can be approximated by the particle effects released by active items) are also altered to increase difficulty. The amount of time a player has to keep the baby safe also impacts the increasing level of difficulty. The room layouts (and their inherent dangers) change from level to level. The baby can pick up speed through the levels leaving the player with less time to make tough choices. More babies may be added in a play date or day care scenario. The hardest levels may limit the player’s field of vision by shutting off the webcam, allowing only limited angles of view from surrounding windows.

There is no inherent story progression as the game’s levels are self-contained in nature.

**Development Specification**

*Hardware*

Most computers and mobile devices with internet connection access and HTML5 capability ought to be able to run the *Baby Crawl* game.

*Software*

*Baby Crawl* is a standalone game that is coded in HTML5 using Construct 2. As such, it may run on any HTML5-capable internet browser. It may also be exported to run on various mobile devices, including Windows, Android, and iOS smartphones and tablets.

The sprites were rendered using Manga Studio and Inkscape, as well as Construct 2’s built-in paint applet.

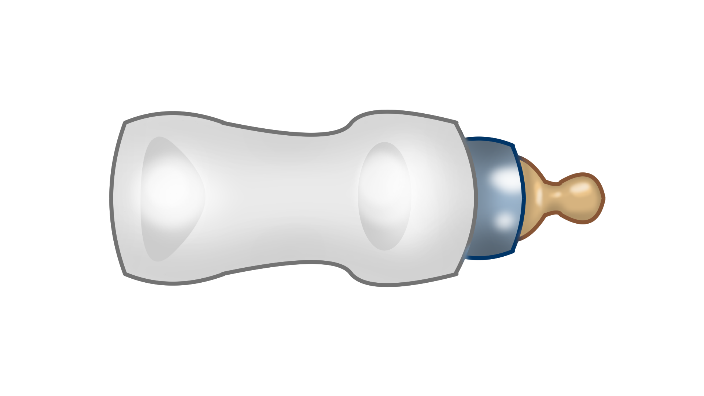
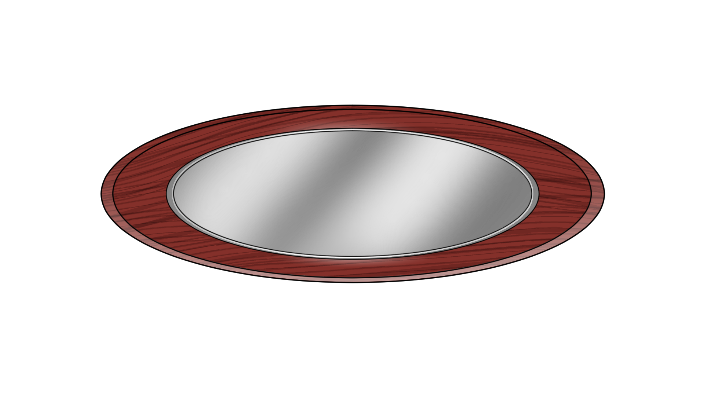
The only audio included at this time is a creative-commons free-to-use song, “Reverie (small theme) (ft. Pitx)” by \_ghost, obtained through ccMixter.org. The music loops throughout the game.

*Algorithm style*

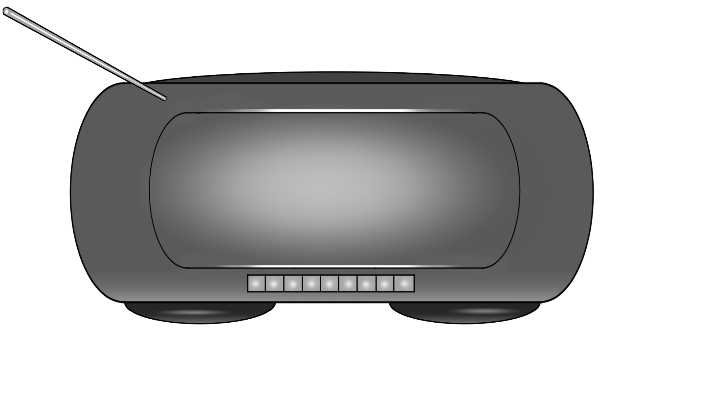
Much of the gameplay algorithm is handled by built-in mechanisms found in Construct 2, e.g., sprite movement angle and speed, collision, path-finding, etc. The character sprite is able to move in any direction and will do so for short distances before turning randomly (or purposefully if within an object’s sphere of influence). Object sprites found in the world will be of four types and will affect the main character accordingly:

1. Obstacles – upon collision with a baby, the latter will simply redirect
2. Desirables (uncontrollable) – items the character desires and will seek out and interact with which the player cannot control (excluded from the demo)
3. Desirables (controllable) – items the player can activate or deactivate which the character will seek out and interact with
4. Dangers – items lying around randomly which the player cannot manipulate and whose interaction with the character will end the game

Items which the player can manipulate are activated by touch (in the case of a touchscreen) or click (in the case of a mouse).



*1. Coffee table - obstacle 2. Baby bottle – uncontrollable desirable*



*3. Stereo – controllable desirable 4. Energy drink - danger*